



Dissertation

Kalman filter based integration of multiple sensor data for the estimation of vehicle trajectories

Ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Doktors der technischen Wissenschaft unter der Leitung von

Ao. Univ. Prof. Dipl.-Ing. Dr. techn. Robert Weber

am E 120-4

Department für Geodäsie und Geoinformation

eingereicht an der Technischen Universität Wien

Fakultät für Mathematik und Geoinformation

von

MSC. Qing Li

[Redacted]

[Redacted]

[Redacted]

Wien, September 11, 2023

(Unterschrift Verfasser/in)

(Unterschrift Betreuer/in)

Acknowledgement

Firstly I want to thank to my supervisor and Prof. Dr. Robert Weber, for providing advice regarding analysis during this doctoral thesis. Without his consistent support, guidance throughout the process, my research would have never been accomplished. Special thanks to the colleagues from TU Graz, at the Institute of Geodesy, Working Group Navigation and the iMar company for providing the sensor data.

I also want to thank to my wife Nora, for understanding me over these past four years and taking care of our lovely boys. Thanks to Reinhard, Silvia, Wolfgang and Christoph from Breitenfurt bei Wien for their support in my life. They give feeling of having a family in Austria.

Moreover, I would like to thank all other people for their support during the development of this doctoral thesis.

Abstract

Recently tremendous efforts aim at the development of satellite-based navigation techniques in combination with further sensors to ensure precise vehicle trajectory determination. The satellite-based navigation with basic GNSS stand-alone processing provides users for a vehicle positioning at several meters accuracy, almost all the time, anywhere on or near the earth's surface. In most geodetic applications a very precise and continuous knowledge of the navigation solution (e.g. position, velocity, attitude, etc.) of the moving body is required, which can not be guaranteed by GNSS stand-alone. For example, tracking the train's position is usually realized via an odometer from the on-board system, which measures wheel rotations. The odometer measurements are repeatedly corrected and initialized at fixed locations known as balises. A balise is an electronic beacon or transponder placed between the rails of a railway as part of an automatic train protection (ATP) system. These sensors constitute an integral part of the European Train Control System, where they serve as "beacons" giving the exact location of a train. This method is very popular to provide an accurate along track position of a moving train. However, balises are expensive sensors, which need to be placed over about 250 000 km of train tracks in Europe.

Therefore, recently many activities are around to monitor trains by EGNOS or an ensemble of on-board sensor devices like GNSS/IMU/odometer. Especially the latter might be a cost-effective but also accurate approach, especially suited for rail secondary lines. A passenger vehicle or a rail-mounted vehicle can simply be equipped with a GNSS unit, an IMU device and other navigation sensors where differential GNSS can provide the accurate position and velocity, while IMU delivers position increments, velocity and direct spatial orientation of the vehicle. When GNSS signal tracking fails for a short period in time this outage can be almost compensated by processing the observation of the strapdown mounted IMU platform in combination with odometer data.

Over the past years, a loosely coupled Kalman filter algorithm based on the fusion of GNSS, IMU and odometer data was developed by the doctoral candidate. This algorithm enables a reliable train positioning. Also an algorithm based on IMU/odometer integration is implemented, and can be applied in case of GNSS signal outages (e.g. tunnels and urban areas).

In parallel, also a software package employing a tightly coupled Kalman filter algorithm has been developed. The software allows for the fusion of GNSS and IMU data, which enables a reliable vehicle positioning performance. This approach has been utilized to estimate the trajectories of slow and fast-moving vehicles like cars or trains.

The first test case of this doctoral thesis concerns navigation along a railway track close to Vienna. The GNSS data is corrected by observations of the national reference

station network TEPOS. GNSS data gathered in combination with IMU observations and odometer data has been processed and the output has been compared to a commercial reference solution. This example shows the capability of the loosely coupled Kalman filter solution to provide centimeter-level accuracy in positioning. However, when the GNSS signal is blocked, the positioning accuracy decreases, and the IMU/odometer combination just provides metre-level accuracy in positioning over time.

In the second test case, an iMAR IMU device and a JAVAD GNSS receiver fixed on a private car provide the data input for the tightly coupled filter algorithm. The quality of the solution was assessed against trajectories calculated with the commercial Inertial Explorer software from NovAtel. All sensor data and reference trajectories of this second test case were generated and provided thankfully by colleagues from the research group Navigation at TU Graz.

Zusammenfassung

Seit geraumer Zeit werden große Anstrengungen zur Entwicklung ausgereifter Algorithmen zur Kombination der satellitengestützten Navigationsverfahren mit zusätzlichen Sensoren unternommen, um eine möglichst präzise und robuste Bestimmung der Fahrzeugtrajektorie zu gewährleisten. Die ungestützte GNSS Einzelpunktbestimmung mittels Codestrecken erlaubt heute global eine Fahrzeugpositionierung im $\pm 3\text{-}5\text{m}$ Bereich. Die meisten geodätischen Anwendungen erfordern allerdings eine höhere Genauigkeit für die statische Positionierung. Aber auch die kontinuierliche Verfolgung von bewegten Fahrzeugen zur Festlegung von Position, Geschwindigkeit und Orientierung stellt Anforderungen, die mit der GNSS Einzelpunktbestimmung nicht zu erfüllen sind. Zum Beispiel basiert die Positionierung von Zügen üblicherweise auf Odometermessungen, welche über die Radumdrehungen und den Radumfang die durchfahrene Strecke festlegen können. Die Messungen des Odometers werden zusätzlich an fixen Positionen mit Hilfe von Balisen initialisiert. Eine Balise ist ein elektronischer Transponder der zwischen den Schienen des Eisenbahngleises als Teil eines automatischen Zugbeeinflussungssystems (ATP) montiert ist. Diese Sensoren sind ein integraler Bestandteil des European Train Control Systems. Balise und Odometer bestimmen den durchfahrenen Streckenabschnitt und die Position des Zuges im Streckenabschnitt. Die Balisen sind jedoch teure Sensoren, die über etwa 250.000 km Eisenbahngleise in Europa platziert werden müssen.

Daher gibt es derzeit viele Entwicklungen zur raumgestützten Positionierung von Zügen durch EGNOS bzw. durch die Fusion der Messungen von am Triebfahrzeug angebrachten Navigationssensoren wie etwa GNSS/IMU/Odometer. Die Sensorfusion ist nicht nur kosteneffizient und präzise, sie ist auch besonders gut für die Anwendung bei Nebenstrecken der Eisenbahn geeignet. Auf den Bahnhauptstrecken wird aufgrund der höheren Geschwindigkeiten wohl auch in Zukunft die Positionierung auf Basis von Balisen erfolgen. GNSS Verfahren im differentiellen Modus liefern Information über die genaue Position und Geschwindigkeit, während die IMU die Position, Geschwindigkeit und direkte räumliche Orientierung des Fahrzeugs bestimmt. Bei kurzzeitiger Unterbrechung der GNSS-Signalverfolgung (z.B. in Tunneln oder in signal-abgeschatteten Gebieten) kann dieser Ausfall durch die Prozessierung der Messungen des IMU Sensors in Kombination mit dem Odometer überbrückt werden.

In den letzten Jahren wurde vom Autor dieser Dissertationsschrift ein lose-gekoppelter Kalman-Filter-Algorithmus entwickelt, der die oben genannten Anforderungen erfüllt. Gleichzeitig wurde auch Software zur eng-gekoppelten Kalmanfilterung entwickelt. Diese Software ermöglicht die Fusion von GNSS- und IMU-Daten und gewährleistet eine zuverlässige Positionierung von sich schnell und langsam bewegenden Fahrzeugen, wie Autos oder Zügen.

Der erste in dieser Doktorarbeit angeführte praktische Anwendungsfall befasst sich mit der Navigation entlang einer Eisenbahnstrecke in der Nähe von Wien. Die am Zug gewonnenen GNSS Beobachtungen wurden mit Beobachtungen des Referenzstationsnetzwerks TEPOS (T-Kom Services real-time positioning) korrigiert und im Anschluss mit IMU Beobachtungen und Odometermessungen verarbeitet. Die Ergebnisse der Berechnung wurden mit einer Referenzlösung einer bekannten kommerziellen Software verglichen. Das Beispiel zeigt, dass mit dem lose gekoppelten Kalman Filter eine Positionierungsgenauigkeit von wenigen Zentimetern erreicht werden kann. Ist das GNSS Signal blockiert, so verringert sich die Positionierungsgenauigkeit für die verbleibende IMU/Odometer Sensorkombination in den Sub-Meter Bereich.

Im zweiten Anwendungsfall wurden eine iMAR IMU und ein JAVAD GNSS-Empfänger verwendet, welche an einem privaten Auto befestigt waren. Die Sensordaten wurden im eng-gekoppelten Filter verarbeitet. Die Qualität der Lösung wurde wieder anhand von Trajektorien bewertet, die mit der kommerziellen Software Inertial Explorer von NovAtel berechnet wurden. Alle Sensordaten und Referenz-Trajektorien dieses zweiten Anwendungsfalles wurden dankenswerterweise von der Forschungsgruppe Navigation der TU Graz bereitgestellt.

Contents

List of Nomenclatures	10
List of Figures	13
List of Tables	17
1. Introduction	18
2. Coordinate frames and transformations	23
2.1. Earth-centered Inertial frame	23
2.2. Earth-centered Earth fixed frame	25
2.3. Local level frame	26
2.4. Body frame	27
2.5. Coordinate frames for the iMar's IMU products	28
2.6. Orientation and rotation	29
2.7. Quaternion	32
2.8. Kinematics	34
2.9. The transformation between the involved coordinate frames	35
3. Basics of GPS positioning	37
3.1. Determination of user position with GPS	37
3.2. Performance of GPS positioning	38
3.3. User equivalent range error	41
3.3.1. Tropospheric delay	42
3.3.2. Ionospheric delay	44
3.3.3. Other significant error sources (Multipath Signal Biases)	45
3.4. Pseudo-range rates and Doppler shift observations	46
3.5. Double difference approach	47
3.6. Phase smoothed code	50
4. Basics of Inertial Navigation and Inertial Sensor	54
4.1. IMU strap-down mechanization	55
4.2. Navigation equations	56
4.2.1. Navigation equations in the i-frame	57
4.2.2. Navigation equations in the e-frame	58
4.2.3. Navigation equations in the n-frame	59

4.3.	Error characteristics of the inertial sensor	63
5.	Kalman filter-based parameter estimation	65
5.1.	Least squares estimation	66
5.2.	Core element of Kalman filter	68
5.3.	Kalman filter algorithm	70
6.	GNSS/IMU integration algorithm	77
6.1.	Lever-arm vector	78
6.2.	Odometer	79
6.3.	IMU State propagation resolved in a Local Navigation Frame	80
6.4.	GNSS/IMU system model	86
6.5.	Loosely coupled GNSS/IMU/ODO integration algorithm	88
6.6.	Tightly coupled GNSS/IMU integration algorithm	92
6.6.1.	Combined GNSS DD observation	96
6.6.2.	Kalman filter initialization	97
6.6.3.	Re-initialization	100
6.6.4.	Cycle slip detection	102
6.6.5.	Combined GNSS DD observation/IMU integration	103
7.	Test drives and results	108
7.1.	The 1st test drive scenario	109
7.1.1.	The iMar inertial measurement system iRailLoc-C	109
7.1.2.	The GNSS device	112
7.1.3.	The iMar software	112
7.1.4.	TU Filter software development	114
7.1.5.	The IMU measurement data	117
7.1.6.	Track of the train	120
7.1.7.	The results	122
7.2.	The 2nd test drive scenario	131
7.2.1.	The IMU Device iNAV-RQH-003	131
7.2.2.	Coordinate system of the applied IMU device	134
7.2.3.	The GNSS device	135
7.2.4.	The Waypoint software	136
7.2.5.	The TU developed tightly coupled GPS/IMU algorithm	139
7.2.6.	The IMU measurement data	142
7.2.7.	Track of the car	144

7.2.8. The GPS measurement data	147
7.2.9. The results	149
7.2.10. Combined processing versus other GPS observation processing	155
7.2.11. The carrier smoothed code	159
8. Conclusions and outlooks	163
Appendices	165
A. Data sheet of iMAR iNAV-RQH-003	165
B. Data sheet of iMAR RailLoc-C	166
C. GNSS antenne GrAnt G3T	167
References	167

List of Nomenclatures

I) List of Abbreviations

<i>DD</i>	Double difference
<i>ECEF</i>	Earth-centered Earth fixed
<i>ECI</i>	Earth-centered Inertial system
<i>EPOSA</i>	Real-time positioning Austria
<i>GNSS</i>	Global Navigation Satellite System
<i>GPS</i>	Global Positioning System
<i>IMU</i>	Inertial measurement unit
<i>LSQ</i>	Least squares method
<i>PPP</i>	Precise point positioning
<i>SPP</i>	Single point positioning
<i>TEPOS</i>	T-Kom Services real-time positioning

II) List of Symbols

α	Roll angle
β	Pitch angle
$\delta\dot{\rho}_{ie,r}^s$	Range rate Sagnac correction
$\delta\rho_{ie,r}^s$	Range Sagnac correction
$\delta f_{ib}^b, b_a$	bias of accelerometers
$\delta w_{ib}^b, b_g$	bias of gyroscopes
δx	State error vector
δ^s	Satellite clock offset
δ_r	Receiver clock offset
δ_{equip}	Receiver electronic offset

δ_{f_D}	Carrier frequency
δ_{f_D}	Doppler shift
δ_{iono}	Ionospheric delay
$\delta_{k_{odo}}$	Odometer scale factor error
$\delta_{m.path}$	Multi-path delay
δ_{tropo}	Tropospheric delay
$\dot{\rho}_r^s$	Pseudo-range rate
\dot{d}_r^s	The true range-rate
γ	Yaw angle
λ	Longitude
$\Omega_{t_r}^s$	Skew-symmetric matrix of the angular rate
ψ	Azimuth
ρ	Pseudo-range
ρ_r^s	Pseudo-range
θ	Elevation
$\tilde{\rho}$	Corrected pseudo-range
φ	Latitude
Az	Satellite zenith angle
c	Speed of light (m/s)
C_t^s	Transformation matrix
e	Unit vector line of sight
F/Φ	Transition matrix
$f_{i_b}^b$	IMU measured acceleration
G	System noise distribution matrix
H	Measurement design matrix

j	Partial water vapor pressure
K	Kalman Gain matrix
k	Epoch number
l^b	Lever arm vector in b-frame
M	Earth's meridional radius of curvature
m_d	Dry-component mapping function
m_w	Wet-component mapping function
N	Earth's transverse radius of curvature
n	The smoothing weight factor
O	Atmospheric pressure
P	Process error covariance matrix
Q	System noise covariance matrix
q	Quaternion
R	Observation covariance matrix
r_s^e	Satellite position in the e-frame
r_r^s	Approximated user position in the e-frame
T	Temperature
v	Observation noise vector
v_r	Velocity of the body
v_s	Velocity of the observed satellite
w_{ib}^b	IMU measured angular rate
w_e	Rate of Earth rotation
w_k	Control vector
z	Measurement vector

List of Figures

1.	Relationship between the e-frame and the n-frame [Bla08]. The author used the symbol I to denote the navigation frame.	27
2.	b-frame of a helicopter [Bla08]	28
3.	Three figures above represent the NED navigation frame (left), the vehicle frame (centered), and the body frame (right). [iMa22]	29
4.	A SD approach between the rover and base station	48
5.	Double difference approach between the rover and base station	49
6.	Comparison of code and carrier-phase measurement noise [SZG13]	51
7.	Phase smoothed code [SZG13]	52
8.	Differences between the raw code and smoothed code measurements (e.g. PRN 24)	53
9.	Elements of Kalman filtering [Gro13]	69
10.	Kalman filter algorithm steps. [Gro13]	71
11.	Kalman filter loop [Jek01].	76
12.	Architecture of a loosely coupled GNSS/IMU/ODO integration. The block diagram at the bottom shows the architecture of the IMU/ODO integration in case of an outage of the GNSS signal	89
13.	Structure of a tightly Coupled Kalman filter based on DD/SPP combined code and phase observation	92
14.	IMU device type iRailLoc-C. [iMa21]	109
15.	ÖBB tests of the inertial measurement system iRailLoc-C [iMa21]	110
16.	Noise of the gyroscopes provided by the iMar company	111
17.	Noise of the accelerometer provided by the iMar compan	111
18.	Graphical interface of iXCOM-CMD (GUI)	113
19.	Lever-arm vector between the IMU device and the GNSS antenna	113

20.	Lever-arm vector between the IMU device and the Odometer	114
21.	Relationship between the IMU axes and the sensor directions placed on the train	114
22.	Flow chart of the TU developed loosely coupled GNSS/IMU/ODO integration algorithm	116
23.	IMU measurements of the first test drive performed by the train	117
24.	Relationship between sensor's regular orientation and its true orientation of the test scenario	118
25.	Figure on the top presents the measured accelerations, the bottom plot presents the measured angular rates	119
26.	Train's trajectory	121
27.	Standard deviation of GNSS processing. Upper one shows the position error, and the middle one shows velocity error, the bottom one shows the positioning quality before the GNSS outage	122
28.	3D positioning solutions processed by TU-Wien Software and reference software (Both solutions are estimated by using the loosely coupled approach)	124
29.	Positioning differences between solutions processed by iMar Software and TU-Wien software	125
30.	Positioning differences between both solutions before the GNSS outage	126
31.	3D velocity solutions processed by TU-Wien software and reference software	127
32.	Velocity differences between solutions processed by iMar software and TU-Wien software	128
33.	Roll/Pitch/Yaw solutions processed by TU-Wien Software and reference software	129
34.	Pitch/Roll/Yaw differences processed by iMar software and TU-Wien software	130
35.	Odometer scale factor error estimated by iMar software and TU-Wien software	131
36.	IMU device type iNAV-RQH-003 [Wie]	131
37.	GNSS and IMU sensors mounted on the car's roof for a test drive of TU Graz in the past [Lae10]	132
38.	Tailored calibrated platform applied for the sensor fusion [Wie]	133

39.	Relationship between the IMU axes and the vehicle axes [TU Graz,2019]	134
40.	Transformation from the ENU system to the NED system carries two rotations in Roll and Yaw directions (from the left side to the right side) [Gra07]	135
41.	Javad GrAnt-G3TH antenna(left) and Javad Sigma TRE-G3TH receiver(right) [Glo]	136
42.	Processing Settings [TU Graz, 2020]	137
43.	GNSS observation processing Settings [TU Graz, 2020]	137
44.	GNSS measurement noise Settings [TU Graz, 2020]	138
45.	Setting "Error Model Values" [TU Graz, 2020]	139
46.	Flow chart of the TU developed tightly coupled GPS/IMU integration algorithm	140
47.	An example of the applied IMU's measurement data saved in a txt file	143
48.	Measured acceleration time series (top), measured angular rate time series (bottom)	144
49.	Consistency between the estimated yaw angles and the car's trajectory	146
50.	Difference of C1 & C2 range measurements between before and after carrier smoothing in [m], shown for all visible satellites tracked by rover	147
51.	Difference of P1 & P2 range measurements between before and after carrier smoothing in [m], shown for all visible satellites tracked by rover	148
52.	Figure on the top represents the number of visible satellites, figure on the bottom represents the PDOP values along the test drive	149
53.	3D positioning solutions processed by TU-Wien software and reference software (Both solutions are estimated by using the tightly coupled approach based on the GPS DD combined code-and carrier-phase observation)	150
54.	Positioning differences between solution processed by Waypoint software and TU-Wien software	151
55.	3D velocity solutions processed by TU-Wien Software and reference software (both solutions are estimated by using the tightly coupled approach based on the GPS DD combined observation)	152

56.	Velocity differences between solutions processed by Waypoint software and TU-Wien software	153
57.	Roll/Pitch/Yaw solutions processed by TU-Wien Software and reference software	154
58.	Pitch/Roll/Yaw differences from Waypoint software and TU-Wien	155
59.	Position differences with respect to Waypoint reference solution (red: DD code+phase; blue: DD code only; green: code SPP)	156
60.	Velocity differences with respect to Waypoint reference solution (red: DD code+phase; blue: DD code only; green: code SPP)	157
61.	Differences in roll, pitch and yaw angles with respect to Waypoint reference solution (red: DD code+phase; blue: DD code only; green: code SPP)	158
62.	Position differences between the solutions processed by Waypoint software and the TU-Wien software. The blue line represents the solution differences calculated with the original GPS code measurements, and the red line shows the solution differences after using phase smoothed code.	160
63.	Velocity differences between the Waypoint reference solution and the DD code only solution based on raw code ranges (blue) and phase-smoothed-code-ranges (red)	161
64.	Differences in roll, pitch and yaw angles between the Waypoint reference solution and the DD code only solution based on raw code ranges (blue) and phase smoothed code ranges (red)	162

List of Tables

1.	Overview of the relevant error sources in the GPS observation equation including the corresponding symbols and magnitude. The parameters above are the relevant terms in equation 3.7 and 3.8.	41
2.	Example of bias errors, scale-factor errors and noises of an IMU device with an high quality gyroscope and an lower quality accelerometer	64
3.	Matrices and vectors in a Kalman filter	76
4.	Specifications of the device iRailLoc-C [iMa21]	110
5.	Specifications of the GNSS receiver OEM7700	112
6.	The determined bound for a bad GNSS solution (standard deviation)	123
7.	The specification of the device iNAV-RQH-003	132
8.	General processing settings used for the tightly coupled GPS/IMU integration	142
9.	Mean value and RMS of all three implemented GPS approaches	159
10.	Mean value and RMS after using PSC and the original code measurements . . .	163

1. Introduction

Geodesy is the science of the measurement, determination and mapping of coordinates of points on or near the Earth's surface [Jek01]. Positioning accuracy up to a few centimeters is required for most applications. Geodetic monitoring and surveying is often based on the traditional technical survey work with geodetic level instruments and total stations until now. However, surveying of hundred kilometers of railway tracks or positioning of a moving train along the entire track with the traditional surveying method is not practicable.

Precise navigation is an alternative method for determining the relative position and velocity of a moving body with respect to a known reference point and with high accuracy. In general, navigation, surveying, tracking and mapping, etc. belong to typical positioning applications [Gro13]. Today, one of the most frequently used positioning methods in practice to achieve the requested solution accuracy in geodetic applications is satellite-based navigation. Standard Global Navigation Satellite System (GNSS) stand alone positioning with a simple handheld receiver offers an about five-meter positioning accuracy [Cam14]. However, this single point positioning accuracy is by far not sufficient for nowadays geodetic and navigation applications, and therefore has to be replaced by GNSS differencing techniques or even by sensor fusion.

Vehicle positioning based on GNSS and IMU integration to determine a precise and continuous trajectory of the moving vehicle has been investigated over several years in detail by various researchers [HLW03], [Shi01], [Gro13], [Jek01] and [TW96]. This technique is a non-contact, fast and accurate positioning method that has considerable potential for rail track or on-road positioning. Different Multi-constellation GNSS Observation technique as processing such Double Difference and Precise Point Positioning integrated with an IMU are investigated [HS15], [VNB20] and [Che16]. Over past years, many current research works focus on the topic, how to overcome the GNSS signal outages. Adding other sensors, such as magnetometer and barometer [BAL11], omnidirectional vision sensor [Kim+11], odometer [Hem08], [Lee06] and [Hin16] as well as Lidar [El-04] in a multi-sensor integrated navigation system have been evaluated in various studies.

All navigation and positioning techniques are based on either position fixing or dead reckoning [EY20]. Dead reckoning measures the distance and direction traveled [Gro13]. An IMU device is a dead reckoning navigation sensor, and this sensor can provide position, velocity and orientation of the body. An initial position of the IMU device is required. The operation of the IMU device depends upon the Newton's laws. The working principle is self-contained,

non-jammable, and can work in all weather conditions. This sensor measures accelerations from three mutually orthogonal accelerometers and angular rates from three gyroscopes. An IMU stand alone processor can combine the accelerations and angular rates to provide the navigation solution. The navigation processor uses the angular rates to provide body's attitude with respect to the navigation frame. On the other hand, after mapping to the appropriate coordinate system the navigation processor uses acceleration measurements to deliver body's velocity and position.

The integration of GNSS and IMU provides a mixture of position fixing and dead reckoning, which has proven to be a successful combination for geodetic applications over years. For example, direct geo-referencing technology is the determination of the position and orientation of individual images, which is an important procedure in mobile mapping. The most commonly used technologies for this purpose are the fusion of the GNSS measurements and IMU measurements [El-04]. Usually, positioning can be provided entirely by GNSS without any additional sensors. The GNSS provides a long-term, accurate absolute positioning at a lower measurement update rate (usually < 10 Hz). This relatively low data rate causes problems in high kinematic applications. Furthermore, GNSS must face its weaknesses, like blockage of line-of-sight signals, multi-path and signal interference or jamming [Bar+22]. The IMU stand alone solutions are appropriate for short-term positioning and can provide a relative navigation solution at a high measurement update rate (up to 2000 Hz). Due to biases and sensor drifts in the gyros and accelerometer measurements, sole IMU based trajectories suffer from significant degradation over time. The accuracy depends on the quality of the IMU sensor.

In case of GNSS signal blockage, the integrated system still delivers a degraded navigation solution based on IMU stand alone processing. To reduce the IMU sensor error and to improve the positioning accuracy the output from another sensor such as an odometer can be added to the integrated system [Hin16]. Hence, a GNSS/IMU/Odometer integrated system for the vehicle positioning is implemented applying especially for the scenario described above. Odometer data are counts of impulses, which relate the wheel's circumference to the velocity and the distance traveled by the train. This odometer data provides non-holonomic constraints as one-dimensional velocity updates and complements the basic IMU/GNSS navigation system. These updates improve the velocity and position estimates of the train at high update rates while GNSS data is used to provide accurate determination in position with low rates. The exponentially weighted moving average method was implemented to characterize the odometer measurement noise in order to construct the measurement covariance matrices. The sensor fusion can limit IMU drift errors by GNSS update information. On the other hand, the IMU cannot be jammed and can help to detect GNSS carrier phase cycle slips and supports a precise

tracking of the Doppler shift. Therefore, the advantages of an integrated system are more than just an improvement of positioning accuracy. By sensor fusion, the advantages of each sensor can compensate for drawbacks of the other.

Kalman filter-based estimation was invented by Prof.R.E.Kalman in 1960 [Gro13]. This filter is a state estimation algorithm that plays a key role in almost all technical fields. This filter can determine the value of system parameters and is commonly used for processing various sensor data in real-time mode. In navigation, the filter states usually comprise the position, velocity and attitude solution as well as sensor errors. The IMU and GNSS integration is usually achieved in one of four different kinds of Kalman filter algorithms: uncoupled system, loosely coupled system, tightly coupled system and deeply coupled system to provide continuous navigation solutions. The most uncomplicated algorithm is the '**uncoupled system**'. The GNSS measurements are used to reset the inertial navigation solution for the IMU stand alone processing in this system periodically. In a '**loosely coupled system**', the GNSS update solution is integrated with the IMU solution explicitly. It means that a loosely coupled GNSS/IMU system uses the GNSS position and velocity solution directly as input in the measurement model to the integration algorithm [Shi05]. The dependence on the availability of GNSS data is the only drawback of this kind of system. A '**tightly coupled system**' bases on the GNSS raw measurements (e.g. pseudo ranges, carrier phase and doppler measurements) and is integrated directly with the IMU measurements. It means that a tightly coupled GNSS/IMU system uses the GNSS pseudo-range, pseudo-range rate and carrier phase as input in the measurement model to the integration algorithm [Gro13]. Suppose there are only two or three satellites visible. In that case, this system can carry out the GNSS measurement update to the integration algorithm, which is a significant advantage compared to the loosely coupled system. The '**deeply coupled system**', also known as an ultra-tightly coupled system, combines GNSS signal tracking and GNSS/IMU integration into an estimation algorithm [TW96]. In this thesis, the presented work is based on a preliminary available software which was used for determining the navigation solution based on a loosely coupled approach. This software has been seriously upgraded within this Phd work to allow for processing DD-GNSS, IMU as well as odometer observations, both in a loosely-coupled and a tightly-coupled Kalman filter algorithm. Currently, the developed software can only run in post-processing mode.

This thesis aims to develop the tightly coupled and loosely coupled integration algorithms to determine the trajectory of the vehicle in different environmental conditions. The first test case aims to develop software capable of performing the states of the train's trajectory by exploiting a loosely coupled Kalman filter scheme. The scenario of the test drive area is a urban region with a bridge, where GNSS signals are blocked. A filtering algorithm based on the

fusion of three sensors is presented. These sensors are an IMU device from the iMar Company, a GNSS receiver from the NovAtel Company and an Odometer fixed on a train provide the data input for the loosely coupled Kalman filter integration algorithm. The contribution of this work is to show how short time GNSS outages can be bridged with the fusion of the IMU and the odometer measurement data. The design of the developed algorithm has been in publication [LW21]. In the developed software, the IMU navigation equations are implemented in the local level frame with NED coordinate directions. This frame allows for an easy visualization of the navigation solution. The quality of this solution was tested against trajectories obtained with the software iXCOM-CMD (iMAR) as reference. All sensor data and reference trajectories were provided thankfully from the iMar Company. In the presented test case, the train traveled a distance of more than 7 km. The data sets are composed of about 30 minutes of recorded sensor measurements.

The second test case aims to develop software capable of performing the states of a land vehicle trajectory by using a tightly coupled Kalman filter. Different from first test case, only GPS observation data are combined with the IMU measurement data. Relative positioning is a positioning method (e.g. Double Difference) that allows to locate a rover with respect to a reference station installed at a permanently fixed position, which coordinates are known [Kap06]. The pseudo-range, the carrier phase and doppler measurements suffer from errors that affect the solution accuracy significantly. By receiving the observation data from one reference station, the rover can perform the DD positioning approach to correct for most of these errors (clock errors of satellites and receiver and atmospheric delay). Hence, one basic code-only and one combined code-phase integration strategy algorithm are developed in the integrated system. The performance of this integrated approach will be assessed by comparing with the reference solutions in terms of position, velocity and orientation. Besides, also single point positioning (SPP) or rather float Precise Point Positioning (float PPP) schemes were tested. The results were published in [LW22].

Overall, three data sets of test drives carried out with a passenger car on the same driving route with their respective reference trajectories are obtained for quality testing of the software. All the measured sensor data and reference trajectories are generated and provided thankfully by colleagues from the research group Navigation TU Graz. The reference trajectories were obtained from the commercial software Inertial Explorer of the Waypoint company, which calculates a tightly coupled integrated approach applying forward-backward filtering. A dual-frequency geodetic-grade receiver (JAVAD) and a strapdown IMU device from the company iMAR Navigation were rigidly equipped on a stable platform on the roof of the car. Each test drive covered more than 11 km distance. The data sets are composed of about 1 hour of

recorded sensor measurements, which contain static and kinematic parts. The GNSS reference station data was obtained from the EPOSA station located at Graz main train station. The reference station's antenna is a JAVAD DELTA antenna, producing reference observables at 1 Hz. A distance of not more than 10 km between the reference station and the rover provides sufficient accuracy for the tightly coupled integrated system [Weber, 2018]. Compared to basic code-only observations, the phase observation equation contains phase ambiguities, which were estimated in an initial step as float values instead of integer values. For kinematic applications, keeping the float ambiguity resolution is fast and reliable, and this fact would lead to dm-level accuracy instead of cm-level [GMV11c]. The ionosphere-free linear combination of the GPS observations (dual frequency) is used to remove the influence of the ionospheric delay.

After considering the theoretical background about coordinate system transformation in chapter 2 the GPS and IMU stand alone processing schemes are explained in chapter 3 and chapter 4. In chapter 5, the Kalman filter and its theoretical background is introduced. Then, the structures of the loosely coupled and the tightly coupled filter are described in chapter 6. The involved sensor devices are presented and their observations fed to the filter are discussed.

Finally, in chapter 7 the field test settings are described and the results from both test drive scenarios are discussed. Here the results obtained with algorithms and software developed in this thesis are compared to reference trajectories to assess their quality. At the end of this doctoral thesis, conclusions and an outlook on future sensor fusion are provided.

2. Coordinate frames and transformations

Navigation aims to determine the position, velocity and attitude of a body. When characterizing the position of this body on or near the Earth's surface, people mostly refer to the coordinates of the body in the underlying system and frame. This chapter introduces the mathematical and physical foundations of the diverse main coordinate frames used in navigation and the relationship between any two coordinate frames. In this research, a body is represented by the sensor.

A coordinate frame or a coordinate system is applied for defining the position of points by means of a set of coordinates along the axes with respect to system's origin. The coordinate system is well-known which includes the rules and physical theories to define the coordinate axes [Jek01]. A coordinate frame comprises from an origin and three axes of either a sensor or a reference [Gro13]. The origin of the sensor is defined as a specific point on the sensor to give the position and velocity. The three axes consist of the normal direction of sensor motion and the vertical direction of the sensor. It is necessary to define a set of three axes, so that the sensor's orientation can be characterized. Establishing the coordinate frame and the coordinate system are important in physics studies. A set of measurements of the sensor are used to define the coordinate frame. The coordinate frame can help us to describe a sensor relative to the frame by the use of the corresponding coordinate system [Gro13].

Any navigation application has to work with at least two coordinate frames, one for the sensor and one for the reference. A sensor with unknown navigation information can be defined as the body frame. The reference frame describes a known body, such as the Earth, relative to the body frame where the position and orientation of the sensor are desired. These two coordinate frames have any relative orientation, which is called spatial orientation or attitude [Gro13].

Chapter 2 begins with a look at frames. The Earth-centered inertial (ECI), Earth-centered Earth-fixed (ECEF), local level and body frames are the common used coordinate frames in navigation. Also the rotation and orientation of the moving sensor are introduced. The chapter ends with a look at transformations between these frames.

2.1. Earth-centered Inertial frame

The *Encyclopedia Britannica* defines Newton's first law as "If the sensor is at rest or moving at a constant speed in a straight line, it will remain at rest or keep moving in a straight line at

constant speed unless it is acted upon by force. This postulate is known as the law of inertia "<https://www.britannica.com/science/Newtons-laws-of-motion>". The Earth-centered inertial frame is defined as the frame in which Newton's first law of motion hold true. A non-rotating and non-accelerating coordinate system is regarded as the true inertial frame.

Generally, IMU measurements are related to the earth-centered inertial frame (ECI), which is fixed to the Earth. This frame enables to define the navigation equations in a simple way. The ECI frame is not a truly inertial system, which is accelerating, but not rotating, around the sun [Jek01]. For the purposes of navigation on the Earth, the Earth accelerates itself, when it moves in its orbit about the Sun. The ECI frame can be treated as a true inertial frame, because the acceleration rate of the Earth about the sun is slow [Nie08], and these effects are smaller than the measured noise obtained by navigation sensor [Gro13]. The symbol for the ECI frame is i ; its origin is fixed to the Earth's center, and this frame is oriented with respect to the Earth's spin axis and the measured positions of the stars (today ICRF's catalogue of quasars). The x-axis points towards the Vernal equinox, the z-axis is along the Earth's spin axis to the North pole, and the y-axis completes the right-handed coordinate system. The x- and y-axes don't rotate with the Earth. Furthermore, this frame is a freely falling in the gravitational field of the sun, moon and planets [Jek01].

The kinematic motion of the sensor in this frame can be described with Newton's second and third law of motion. The *Encyclopedia Britannica* states that "the force on an object is equal to its mass times its acceleration. In the third law, when two objects interact, they apply forces to each other of equal magnitude and opposite direction". Assuming the mass is constant, the form of Newton's second law can be defined as follows:

$$m_i \ddot{x} = F, \quad (2.1)$$

m_i is the inertial mass. \ddot{x} denotes the acceleration which is directly proportional to F (the applied force) in an inertial system. The ambient gravitational field of the sensor also causes an acceleration that is independent of the mass [Jek01]. It is necessary to modify Newton's second law in a rotating frame for the centrifugal and coriolis accelerations. Due to an ambient gravitational field the acceleration is added on the right side of the equation 2.1.

$$m_i \ddot{x} = F + m_g g, \quad (2.2)$$

The Earth's gravitational acceleration can be denoted as g . m_g is the gravitational mass. The principle of Equivalence states that inertial mass and gravitational mass are identical,

$m_i = m_g$, and it can be obtained:

$$\ddot{x} = a + g, \quad (2.3)$$

where $a = F/m$ is an acceleration due to an applied force. The variable a is well known as specific force (force per unit mass). The variable g comprises from the Earth's gravitational acceleration, as well as the differences in tidal accelerations relative to the Earth's center. The tidal accelerations are the solar, lunar and planetary gravitational accelerations [Jek01].

The inertial coordinate system is called the celestial coordinate system. The International Earth Rotation and reference systems Services (IERS) establishes the inertial system, by means of the International Celestial Reference System (ICRS). The system is realized by a catalogue of 4536 quasars (ICRF3) whose directions are defined by using the Very Long Baseline Interferometry (VLBI) [Has20].

2.2. Earth-centered Earth fixed frame

As the name implies, the "Earth-centered Earth fixed" frame (ECEF) is a frame that is fixed to the Earth [Jek01], and the symbol \mathbf{e} denotes the ECEF. In many ways, this frame is similar to the Earth-centered inertial coordinate frame. Its origin is fixed to the Earth's center, and the coordinate axes define a Cartesian coordinate system, which is oriented with respect to the Earth [Gro13]. Coordinates in the ECEF reference frame can also be expressed in geodetic coordinates, which comprises the geodetic latitude, geodetic longitude and ellipsoidal height. The z-coordinate axis almost coincides with the z-axis in the ECI frame. It points along the earth rotation's axis from the center to the true north pole. The x-axis points from the center to the intersection of equator with the reference meridian (IRM), which defines 0° longitude. The y-axis completes the right-handed coordinate system, and points from the origin to the intersection of the equator with 90° east meridian.

The ECEF is a realization of the International Terrestrial Reference system (ITRS). ITRS is realized through modern space applications, such as GNSS, SLR, VLBI and Doris. The ECEF is geocentric [Jek01]. The GNSS services use the ECEF system as the primary coordinate system. The ECEF is relevant for the navigation applications, as mostly people want to fix their position with respect to the Earth.

2.3. Local level frame

The local level frame of the sensor is defined as a topocentric coordinate frame, either with the North-East-Down (NED) or East-North-Up (ENU) orientation of the coordinate axes. This local level frame is well suited to act as the navigation frame, which is formed by a tangent plane to the geodetic reference ellipse. This topocentric frame is denoted by the symbol \mathbf{n} (some author uses the symbol \mathbf{l}) [Gro13]. The origin of this frame is the body, or rather in this research, the center of mass of the navigation sensor. This tangent plane is used to define a navigation frame that is useful for determining the navigation solution of the sensor relative to this reference frame. The n-frame is an important frame used for local navigation purposes, and is mostly fixed with a platform on a vehicle and moves with the body frame.

The navigation frame can be defined using a set of Cartesian coordinate axes. The z-axis is aligned with the ellipsoidal normal at a point in the down direction. The x-axis always points to the north direction, and the y-axis indicates the east direction. The second option is an ENU coordinate system. The z-axis is also aligned with the ellipsoidal normal of point but in the up direction. The x-axis points to the east, and the y-axis indicates to the north. The n-frame is primarily used to determine the sensor's velocity in a three-dimensional local direction. The reason is that the sensor axes, such as of an IMU, are easily aligned with this local direction.

In addition, the n-frame is important for the navigation applications, because the users always want to know their attitude relative to north, east and down directions [Gro13]. However, in this frame, the axes don't pass through the Earth's center of mass [Jek01]. Hence, when the sensor's position is required, it is better and easier to estimate it using GNSS observation. The transformation of parameters between e-frame and n-frame have to be determined permanently to relate IMU observations within the ECEF. The purpose of the n-frame primarily is to deliver velocities of the sensor in the NED or in the ENU directions.

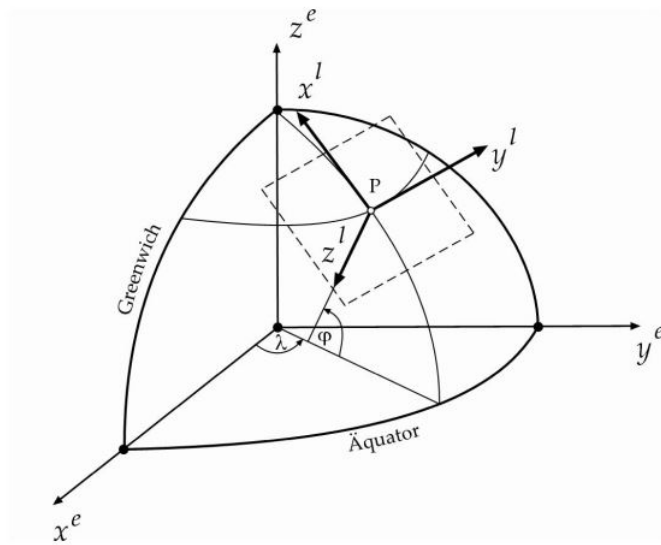


Figure 1: Relationship between the e-frame and the n-frame [Bla08]. The author used the symbol **I** to denote the navigation frame.

In this thesis, the doctoral candidate has chosen the NED coordinate axes orientation due to its popularity and ease of use in inertial navigation. Particularly, this frame allows for an easy visualization of the navigation solution.

2.4. Body frame

The body frame is the coordinate frame of the IMU. This frame is denoted with symbol **b**. The coordinate axes of the body frame usually coincide with the major axis of the vehicle. In case of a strapdown mounted IMU system, the inertial platform is orientated according to the axes of the body frame. The origin of the b-frame coincides with the origin of the n-frame, such as the center of mass of the sensor. The combination of the forward, right and down (direction of gravity) directions is the most common order of the axes in the b-frame. The attitude of the b-frame with respect to the n-frame can be described by the Euler angles. The axes in the b-frame are pointing towards roll, pitch and yaw axes of the platform. Roll motion is about the x-axis, pitch motion is about the y-axis and yaw motion is about the z-axis [Gro13]. Furthermore, an IMU device measures the specific force and angular rate of the sensor, which are relative to the i-frame along the body-frame axes using an accelerometer and a gyroscope.

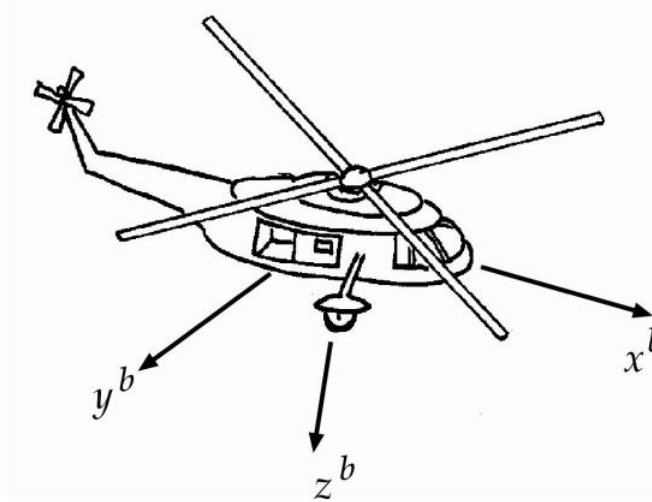


Figure 2: b-frame of a helicopter [Bla08]

2.5. Coordinate frames for the iMar's IMU products

The iMar company is well-known in the industry for their IMU products. iMar's IMUs are designed for use in different environments, such as underwater, aircraft and land vehicles. As mentioned before, for defining the n-frame, the ENU and the NED coordinate frames are commonly in use. Using iMar's IMU for the aircraft and watercraft, the ENU frame is usually defined as the n-frame, and for the land vehicles (e.g., a car), the NED frame is usually used. In this doctoral thesis, the measurement data of two iMar's IMU devices specifically for the land vehicle are generated and analysed.

The vehicle's frame represents the coordinate system tied to the vehicle. This frame has the same origin as the b-frame. The chapter 2.4 describes the b-frame, which consists of the x-axis, y-axis, and z-axis. In general, it is helpful to set a same orientation regarding to z-axis for the vehicle's frame. Therefore, the x-axis of the vehicle's frame is longitudinal in vehicle's forward direction, the y-axis points lateral to the right-hand side, and z-axis represents the downwards direction of the vehicle.

In summary, the IMU's output measurements in the b-frame can be preconfigured that respect to the orientations of the coordinate axes of the n-frame and the vehicle's frame [iMa22]. The relationships between the NED-frame, the vehicle frame and the b-frame can be shown in figure 3.

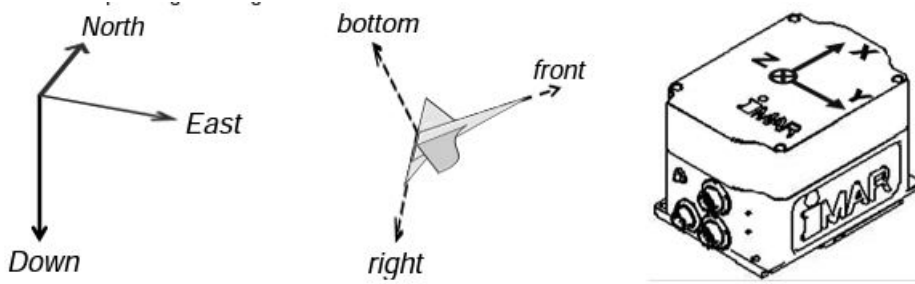


Figure 3: Three figures above represent the NED navigation frame (left), the vehicle frame (centered), and the body frame (right). [iMa22]

2.6. Orientation and rotation

Knowledge of the spatial orientation of the sensor is required for defining the coordinate transformation from one frame to another one. Considering two reference frames with a common origin, the transformation between the frames can be achieved with the direction cosines, Euler angles or quaternions. The coordinate transformation matrix is used to present the spatial orientation. This is a 3-by-3 matrix, which is denoted by symbol C_t^s . The coordinates of the sensor may be transformed from the t-frame to the frame **s** by the appropriate coordinate transformation matrix C . The subscripts **t** and **s** signify the original frame and the transformed frame.

$$x^s = C_t^s x^t, \quad (2.4)$$

The components of the matrix C comprise of the product of the unit vectors which describe the axes in s- and t-frames. They are equal to the cosines of angles between the axes:

$$C_t^s = \begin{pmatrix} C_{s1,t1} & C_{s1,t2} & C_{s1,t3} \\ C_{s2,t1} & C_{s2,t2} & C_{s2,t3} \\ C_{s3,t1} & C_{s3,t2} & C_{s3,t3} \end{pmatrix} = \begin{pmatrix} \cos C_{s1,t1} & \cos C_{s1,t2} & \cos C_{s1,t3} \\ \cos C_{s2,t1} & \cos C_{s2,t2} & \cos C_{s2,t3} \\ \cos C_{s3,t1} & \cos C_{s3,t2} & \cos C_{s3,t3} \end{pmatrix}, \quad (2.5)$$

The matrix C has nine components, this matrix is usually called the direction cosine matrix. C_t^s denotes a rotation matrix and is therefore an orthogonal matrix:

$$C_t^s = (C_s^t)^T. \quad (2.6)$$

Also an arbitrary 3-by-3 matrix A can be transformed from the t-frame to the s-frame. The transformation between both frames can be derived as follows. Assuming that

$$y^t = A^t x^t, \quad (2.7)$$

The equations 2.6 and 2.7 are used now to derive the transformation.

$$C_s^t y^s = A^t C_s^t x^s \quad \text{and} \quad y^s = C_t^s A^t C_s^t x^s. \quad (2.8)$$

Finally, from all equations above, the following equation can be derived.

$$A^s = C_t^s A^t C_s^t. \quad (2.9)$$

Moreover, the multiplication of two coordinate transformation matrices can be defined as follows:

$$C_t^s = C_y^s C_t^y. \quad (2.10)$$

Euler angles are an alternative to direction cosine matrix transformation, which are used to describe the relative orientation of the sensor between the t- and the s- frames by the three rotations about each coordinate axis. In the navigation application, the t- frame is usually the body frame with respect to the navigation frame. The rotation can be described using a rotation matrix about the specific axes.

$$R_1(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix}; R_2(\theta) = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix}; R_3(\theta) = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (2.11)$$

$R_i(\theta)$ ($i= 1,2,3$) represents the rotation about the one axis by the angle θ .

Three rotation matrices about three axes can multiply together to perform the coordinate transformation between both frames. Each $R_i(\theta)$ defines a specialized direction cosine matrix and this matrix is orthogonal, $R_i(\theta)^{-1} = R_i(\theta)^T$ is obtained.

The s-frame can be the result of rotating the t-frame by three rotation angles. First

rotation about the axis by α , second rotation about the second axis by β , finally third rotation by γ [Jek01]. The transformation matrix C from the t-frame to the s-frame is then defined by a sequence of rotation $R_3(\gamma)R_2(\beta)R_1(\alpha)$.

$$x^s = R_3(\gamma)R_2(\beta)R_1(\alpha)x^t. \quad (2.12)$$

γ is the yaw angle. The rotation based on this angle is called yaw rotation, representing a rotation about the sensor's vertical axis in both frames. Which means that the x- and y-axis are rotated. α represents the roll angle, and a rotation about the x-axis is performed by the roll angle, which means that this angle rotates the y- and z-axis. β represents the pitch angle, and the pitch rotation is performed about the y-axis, so that the pitch angle rotates the x-and z-axis. Euler angles can also be used to express the sensed data in terms of rates of angles in the n-frame.

The Euler angles and the direction cosine matrix both can be used for the description of the orientation of the sensor. The relation between the Euler angles and the direction cosines can be shown with the formula [Jek01]:

$$R_3(\gamma)R_2(\beta)R_1(\alpha) = \begin{pmatrix} \cos\gamma\cos\beta & \cos\gamma\sin\beta\sin\alpha + \sin\gamma\cos\alpha & -\cos\gamma\sin\beta\cos\alpha + \sin\gamma\sin\alpha \\ -\sin\gamma\cos\beta & -\sin\gamma\sin\beta\sin\alpha + \cos\gamma\cos\alpha & \sin\gamma\sin\beta\cos\alpha + \cos\gamma\sin\alpha \\ \sin\beta & -\cos\beta\sin\alpha & \cos\beta\cos\alpha \end{pmatrix}. \quad (2.13)$$

If the direction cosines in the C matrix are known, the Euler angles can be derived by:

$$\alpha = \tan^{-1}\left(\frac{-C_{3,2}}{C_{3,3}}\right); \beta = \sin^{-1}(-C_{3,1}); \gamma = \tan^{-1}\left(\frac{-C_{2,1}}{C_{1,1}}\right); \quad (2.14)$$

When the rotation angles are small, and assuming that $\cos\alpha \approx 1$, $\sin\alpha \approx \alpha$, etc.; the relationship between the Euler angles and the direction cosine matrix (see Equation 2.13) becomes [Jek01]:

$$R_3(\gamma)R_2(\beta)R_1(\alpha) \approx \begin{pmatrix} 1 & \gamma & -\beta \\ -\gamma & 1 & \alpha \\ \beta & -\alpha & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & -\gamma & \beta \\ \gamma & 0 & -\alpha \\ -\beta & \alpha & 0 \end{pmatrix} = I - \Psi, \quad (2.15)$$

Ψ denotes the skew-symmetric matrix of the small rotation angles. If rotation angles (α, β, γ) are small, from the equation 2.12 it can be obtained:

$$C_t^s = I - \Psi = \begin{pmatrix} 1 & \gamma & -\beta \\ -\gamma & 1 & \alpha \\ \beta & -\alpha & 1 \end{pmatrix}. \quad (2.16)$$

Considering two frames that rotate with respect to each other, and the space orientation also changes over time. To describe these changes, the derivative of the transformation matrix C_t^s with respect to time can be determined. It can be called as \dot{C}_t^s .

The time derivative of the transformation matrix is formulated by

$$\dot{C}_t^s = C_t^s \Omega_{st}^t. \quad (2.17)$$

Ω_{st}^t is a skew-symmetric matrix that consists of three components of angular rate measurement $(w^1, w^2, w^3)^T$.

2.7. Quaternion

Quaternions can also be used to describe a rotation between two frames instead of three rotation angles. Quaternions are a convenient mathematical notation to describe the spatial orientation parameter of a sensor in 3D space. A quaternion is a hyper-complex number that represents a vector with four basis components. This method avoids trigonometric functions appearing in the equation 2.14. Therefore, using quaternions makes implementing an algorithm easier [Web21].

$$q = (q_1, q_2, q_3, q_4). \quad (2.18)$$

q_1 is a function of the rotation angle and q_2, q_3, q_4 can be defined as vector pointing to the three rotational axes. In this case, the coordinate transformation from the t- frame to the s-frame can be defined by:

$$q = (q_1, q_2, q_3, q_4) = \left(\cos \frac{\theta}{2}, e_t^s \sin \frac{\theta}{2}, e_t^s \sin \frac{\theta}{2}, e_t^s \sin \frac{\theta}{2} \right) \quad (2.19)$$

where e_t^s is the normalized rotation axis and θ is not the rotational angle but the

transformation angle.

$$\theta = 2(\arccos q_1); e_t^s = \frac{q_{2:4}}{\|q_{2:4}\|} \quad (2.20)$$

For quaternions, „imaginary units” i, j and k represent the units along three mutually orthogonal axes. „Imaginary units” are defined by:

$$i^2 = -1; k^2 = -1; j^2 = -1. \quad (2.21)$$

A quaternion is given by the sum:

$$q = q_1 + iq_2 + jq_3 + kq_4 \quad (2.22)$$

where $q_{1:4}$ are real numbers. The conjugate of q is obtained by reversing the sign of all „imaginary unit”:

$$q^* = q_1 - iq_2 - jq_3 - kq_4 = (q_1, -q_2, -q_3, -q_4) \quad (2.23)$$

The transformation matrix C_t^s can be noted in a form of the quaternion to perform the transformation. The relationship between the transformation matrix and the quaternion can be described as:

$$C_t^s = \begin{pmatrix} q_1^2 + q_2^2 - q_3^2 - q_4^2 & 2(q_2q_3 + q_4q_1) & 2(q_2q_4 - q_3q_1) \\ 2(q_2q_3 - q_4q_1) & q_1^2 - q_2^2 + q_3^2 - q_4^2 & 2(q_3q_4 + q_2q_1) \\ 2(q_2q_4 + q_3q_1) & 2(q_3q_4 - q_2q_1) & q_1^2 - q_2^2 - q_3^2 + q_4^2 \end{pmatrix} \quad (2.24)$$

and

$$\begin{aligned} q_1 &= \frac{1}{2} \sqrt{1 + C_{1,1} + C_{2,2} + C_{3,3}} \\ q_2 &= \frac{C_{3,2} - C_{2,3}}{4q_1} \\ q_3 &= \frac{C_{1,3} - C_{3,1}}{4q_1} \\ q_4 &= \frac{C_{2,1} - C_{1,2}}{4q_1}. \end{aligned} \quad (2.25)$$

Where $C_{j,j}$ is a direction cosine (see chapter 2.6) [Jek01]. The representation of the spatial orientation of a moving sensor by using Euler angles or the direction cosine matrix is easy to understand, but less computationally efficient. The quaternion is more computationally efficient for some processes compared to Euler angles or a direction cosine matrix. Also a quaternion avoids singularities, which is a big problem that occurs when describing the orientation of the moving sensor with the Euler angles.[Gro13] Overall, the quaternions are used to determine a sensor's spatial orientation relative to a coordinate reference frame. It can also perform the coordinate transformation, such as transforming a state from the i-frame into the b- frame.

In this research, the quaternion is applied for obtaining the corresponding direction cosine matrix C_t^s , which can describe the attitude of the moving vehicle.

2.8. Kinematics

Kinematics is the science of the motion of the body without consideration of the cause of that motion. In navigation, an investigation on the kinematic performance is mainly a study of the kinematic states, such as navigation solutions, acceleration and angular rate, etc. A notation x_{sr}^t can be used for representing the axial state vectors all above. It can be described as a vector x of the t-frame with respect to s-frame, represented in the r-frame. If there the sensor is at rest, the t-frame is equal to the r-frame. The r-frame presents a resolved frame which could also be defined by the s-frame, the t-frame or totally a different frame [Gro13].

For instance, the angular rate vector can be represented using the notation w_{sr}^t . This is the rate of the rotation angle of the t-frame with respect to the s-frame axes, resolved about three axes in r-frame [Gro13]. In general, w is an axial vector with three components $w = (w_1, w_2, w_3)^T$. These rates are infinitesimal angles in the ratio to infinitesimal increments of time. When w is associated with the Earth's rotation, the axes are aligned with Earth's spin axis, then $w = (0, 0, w_e)^T$, where w_e is the angular rate of the Earth's rotation [Jek01]. The relationship of the angular rate vector in a different frame can be described by using the corresponding transformation matrix:

$$w_{sr}^t = C_s^t w_{st}^s = -C_s^t w_{rs}^s. \quad (2.26)$$

From this, it follows that:

$$w_{st}^s = -w_{rs}^s, \quad (2.27)$$

The angular rate vector, which is resolved about the same axes in the same frame, can be added,

$$w_{sr}^t = w_{so}^t + w_{or}^t. \quad (2.28)$$

The 3x3 skew-symmetric matrix of the angular rate, Ω_{sr}^t , is described by [Gro13]:

$$\Omega_{sr}^t = [w_{sr}^t \times] \quad \text{and} \quad \Omega_{sr}^t = \begin{pmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{pmatrix}. \quad (2.29)$$

2.9. The transformation between the involved coordinate frames

The state vectors of the sensor in motion can be transformed between the i-, e-, n- and b-coordinate frames. In general, the transformation matrix from the i-frame to the e-frame is a rotation about three axes. Only the Earth rotation is considered, which means that nutation, precession and polar motion can be neglected, because these effects are small. The e-frame and the i-frame differ by a rotation about z-axes. Hence, the frames rotate only about z-axis with argument $w_e t$:

$$C_i^e = \begin{pmatrix} \cos(w_e t) & \sin(w_e t) & 0 \\ -\sin(w_e t) & \cos(w_e t) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (2.30)$$

The vector w_e is the rate of Earth rotation, and t denotes the time. Moreover, $w_e t$ is the angle that rotates the i-frame into the e-frame. Here, w_e has the value $7.2921158 \cdot 10^{-5} \text{rad/s}$ [Shi01];

$$w_{ie}^e = \begin{pmatrix} 0 & 0 & w_e \end{pmatrix}^T. \quad (2.31)$$

and

$$w_{ie}^e = w_{ie}^i. \quad (2.32)$$

The relative orientation of the sensor with respect to an e-frame in the n-frame whose axes points to the NED direction can be determined by the geodetic latitude φ and geodetic

longitude λ of the sensor in the n-frame.

$$C_n^e = \begin{pmatrix} -\sin\varphi\cos\lambda & -\sin\lambda & -\cos\varphi\cos\lambda \\ -\sin\varphi\sin\lambda & \cos\lambda & -\cos\varphi\sin\lambda \\ \cos\varphi & 0 & -\sin\varphi \end{pmatrix}, \quad (2.33)$$

w_e is translated to n-frame by using the matrix C_n^e [Shi01], which is given by

$$w_{ie}^n = C_e^n w_{ie}^e = \begin{pmatrix} w_e \cos\varphi & 0 & -w_e \sin\varphi \end{pmatrix}^T. \quad (2.34)$$

A relevant transformation for navigation applications is how to perform the transformation between the b-frame and the n-frame. The Euler angles (roll α , pitch β and yaw γ) with respect to the NED axes define this transformation. The b-frame is defined by the sensor whose axes point to forward, right and down directions. The transformation from b-frame to n-frame bases on three rotations about the corresponding axes by the negative corresponding Euler angles, and the rotation matrix consists of three individual rotations. (see equation 2.12)

$$C_b^n = R_3(-\gamma)R_2(-\beta)R_1(-\alpha), \quad (2.35)$$

From this, it follows that

$$x^n = C_b^n x^b = R_3(-\gamma)R_2(-\beta)R_1(-\alpha)x^b. \quad (2.36)$$

3. Basics of GPS positioning

GNSS can provide the user globally with a three-dimensional positioning solution by passive ranging using radio signals transmitted by the GNSS satellites. The most well-known GNSS navigation system is the US GPS. As this doctoral thesis focuses just on processing GPS data, the following sub-chapters just deal with GPS positioning, while not considering other global GNSS navigation systems.

GPS was established in 1993 by the U.S. Secretary of Defense (OSD). It can provide accurate, continuous, worldwide covered, in all weather conditions three-dimensional position and velocity information to users with the appropriate receiving equipment [Kap06]. An indispensable prerequisite of operating the GPS measurements to fix the user's location is an unobstructed line of sight to four or more GPS satellites.

3.1. Determination of user position with GPS

GPS positioning is based on passive ranging of satellite signals. Passive ranging determines the distance between a single satellite and a sensor. The user's position can be anywhere on or close to the Earth's surface.

The definition of the pseudo-ranges $P(k)_r^s$ from the satellite s to the receiver r is as follows:

$$P(k)_r^s = c(t_r(k) - t^s(k)). \quad (3.1)$$

where $t_r(k)$, is the reception time (sec) of the signal at epoch k at the receiver r . $t^s(k)$ is the transmit time of the same signal based on the clock of the satellite s (sec) [Kap06]. Here, c is the speed of light and is 299792458 (m/s).

Unfortunately, GPS satellite clocks and GPS receiver clocks are not synchronized. In addition, satellite signal delays exist in the ionosphere and troposphere as well as multi-path and electronic receiver delays, which have to be corrected. Therefore, the GPS measurements between the receiver and the satellites are denoted as pseudo-ranges, accounting for the satellite clock error and receiver clock error, the pseudo-range code observation equation is obtained:

$$P(k)_r^s = \rho(k)_r^s + c\delta_r + c\delta^s, \quad (3.2)$$

where δ_r and δ^s are the receiver clock error and the satellite clock error including the

relativistic satellite clock correction. Here, ρ describes the geometric distance from the satellite to the receiver without any atmospheric influences. From "Pythagoras Theorem", the range is obtained by [GMV11c]:

$$\rho = \sqrt{(x_s - x_r)^2 + (y_s - y_r)^2 + (z_s - z_r)^2} + c\delta_r. \quad (3.3)$$

where x_s , y_s and z_s denote the position of the satellites which can be calculated by means of a set of parameters broadcasted in the navigation message and the corrected measurements of the transmission time of the signals. The parameters in the navigation message describe the satellite orbit, known as the ephemeris. The measurement noise of highend receivers for the C/A-code ranging is within 0.3m, and for the P- and L-code ranging is 0.1 - 0.2m. The antenna's position (x_r , y_r and z_r) and receiver clock error δ_r remain unknown. In order to determine the antenna's position and receiver clock error (4 unknown parameters), pseudo-range measurements to at least 4 satellites are required. An GPS measurement equation-system with at least 4 nonlinear equations shows up:

$$\begin{aligned} P_1 &= \sqrt{(x_{s_1} - x_r)^2 + (y_{s_1} - y_r)^2 + (z_{s_1} - z_r)^2} + c\delta_r \\ P_2 &= \sqrt{(x_{s_2} - x_r)^2 + (y_{s_2} - y_r)^2 + (z_{s_2} - z_r)^2} + c\delta_r \\ P_3 &= \sqrt{(x_{s_3} - x_r)^2 + (y_{s_3} - y_r)^2 + (z_{s_3} - z_r)^2} + c\delta_r \\ P_4 &= \sqrt{(x_{s_4} - x_r)^2 + (y_{s_4} - y_r)^2 + (z_{s_4} - z_r)^2} + c\delta_r \end{aligned} \quad (3.4)$$

These nonlinear equations of a stationary process can be solved for the unknowns by using the least square method or the Kalman filter. The least-square estimation is an iterative estimation algorithm to solve linear systems. The Kalman filter estimates some unknown state variables based on the measurements observed over time. In general, the positioning accuracy by using GPS standalone processing depends on the quality of the pseudo-range measurements, the number and geometry available satellites, and the satellite ephemeris data.

3.2. Performance of GPS positioning

The GPS satellites are equipped with the atomic clocks with high stability which control all satellite timing operations. However, the satellite clock error in the navigation data message is sized such as the deviation between satellite time and GPS time may be as large as 1 ms. An offset of 1 ms translates to 300km pseudo-range error [Kap06]. The satellite clock error can be compensated by the receiver with the correction parameter obtained from the navigation

message by using the second- order polynomial:

$$\delta^s = a_{f0} + a_{f1}(t - t_{oc}) + a_{f2}(t - t_{oc})^2 + \Delta t_r, \quad (3.5)$$

where:

a_{f0} = the clock correction parameter bias

a_{f1} = the clock correction parameter drift

a_{f2} = the clock correction parameter drift rate

t_{oc} = the clock data reference time

t = the current time epoch

Δt_r = the correction due to relativistic effects.

Some sources of error corrupt the signal travel time between the satellites and the receiver. When the signals from the GPS satellites travel through the ionosphere and troposphere, these cause signal delays. The code-signals pass those mediums with reduced speed compared to a vacuum.

Further, the total signal transmission time can be affected by the reflections caused by the reception of signals arrived not only directly from satellites, but also reflected or diffracted from nearby objects (e.g. multi-path effect). The signals also travel through the GPS receiver hardware, and there are slight signal travel time delays between the receiver code correlation point, satellite electronics and the antenna phase center. The antenna phase center is a defined point where the GPS satellite signal is received. However, as the phase center is not a physical center, the current position of the point varies due to the different directions of incoming satellite signals. This antenna phase center variations can reach several centimeters [Daw10]. Therefore, the offset and variation of a GPS receiver's antenna should be determined. The actual transit time may be represented by:

$$\delta t_{DP} = \delta t_{tropo} + \delta t_{iono} + \delta t_{m.path}, \quad (3.6)$$

where:

δt_{iono} = the delay due to ionospheric refraction

δt_{tropo} = the delay due to tropospheric refraction

$\delta t_{m.path}$ = the delay due to multi-path effect on pseudo-ranges

The extended pseudo-range observation equation between a satellite s and a receiver r

can be derived as follows:

$$P(k)_r^s = \rho(k)_r^s + c\delta_r - c\delta^s + \delta t_{D_P} + P_r - P^s + \epsilon_{r,P}^s, \quad (3.7)$$

ρ_r^s is the true geometric range between the satellite at time of transmission and receiver Antenna Phase Centers (APCs) at time of reception, respectively, including the relativistic path range effect (less than 2 cm [SZG11]) and the Sagnac effect. P_r and P^s denote the code hardware error sources of the receiver and the satellite. $\epsilon_{r,P}^s$ is the code measurement noise, in meter. A pseudo-range observation equation of this kind can be established for each of the transmitted codes.

A similar equation for the carrier-phase observation equation can be derived as follows:

$$\phi(k)_r^s = \rho(k)_r^s + c\delta_r - c\delta^s + \delta t_{D_\phi} + \lambda(N_r^s + \phi_r - \phi^s) + \epsilon_{r,\phi}^s, \quad (3.8)$$

and

$$\delta t_{D_\phi} = \delta t_{tropo} - \delta t_{iono} + \delta t_{m.path}. \quad (3.9)$$

where:

λ = the wavelength at the carrier frequency and $\lambda = c/f$

f = the nominal frequency

N_r^s = the phase ambiguity for the respective carrier frequency

ϵ_r^s = the remaining random errors and white noise

The phase measurement is not only influenced by the delays due to atmospheric refraction, multi-path effect and receiver electronic effect, but also by the phase wind up effect and geometric offsets of the phase center from the geometric center of the receiver antenna. The raw carrier-phase measurements are usually delivered in the units of cycles, sometimes the measurement can be translated to meters by multiplying the corresponding equation with the carrier wavelength λ . Compared to the pseudo-range measurement, the phase measurement includes a so-called ambiguity N . Here, N is an integer or float number multiple of a wavelength λ . ϕ^s and ϕ_r are phase calibration biases of the receiver and satellite, which are frequency dependent. Together with the ambiguity these parameters describe the number of full wavelengths between satellite and receiver [Gla22].

The precise orbits correction and the precise satellite clock correction can be obtained from the International GNSS service (IGS). The GPS standalone processing based on

the ambiguity-fixed precise point positioning (PPP) approach can provide the post-processed positioning solution in static and kinematic modes with centimeter level accuracy [BAE17]. Post-processed PPP solutions usually provide better results than Real-time solutions [GMV11b].

Alternatively, difference observation schemes (e.g. RTK) can achieve a position solution accuracy at the centimeter level in real-time positioning after the integer values of the ambiguity have been correctly fixed, if the baseline between rover and master station are less than 10 km [BG09] and [Zha+16], or active GNSS reference networks are utilized. The clock errors of the satellites and receivers are eliminated and the atmospheric errors are also minimized.

3.3. User equivalent range error

The accuracy of the GPS processing depends on the quality of the pseudo-range and carrier-phase measurements as well as broadcast navigation data [Kap06]. The user-equivalent range error (UERE) is the uncertainty in the pseudo-range values between a satellite and the user's receiver which is caused by control, space and user segment. The UERE can be divided into the user equipment error (UEE) and the signal-in-space range error (SISRE) [Liu+22]. The UEE mainly consists of the errors related to user segments, including receiver noise, multipath delay, ionospheric delay and tropospheric delay, etc. The SISRE consists of the error of the broadcast ephemeris and the satellite clock errors. The total sum of the UERE obtained due to all the error sources is a few metres. The table 1 is an overview of typical UERE of GPS caused by the system and environmental errors in the pseudo-range measurement equations:

Symbol	Error sources	Typical Magnitude [Gla22]
ρ	Theoretical range	2×10^7 [m]
δ^s	Satellite clock error	ns
δ_r	Receiver clock error	ns - ms
δ_{orb}	Satellite orbit error (broadcast)	1m
δ_{tropo}	Tropospheric delay	1m - 15m
δ_{tropo}	Wet tropospheric delay	mm - m
δ_{iono}	Ionospheric delay	1m -100 m
$\delta_{m.path}$	Multi-path error	0.1m - 1.5m [Gro13]

Table 1: Overview of the relevant error sources in the GPS observation equation including the corresponding symbols and magnitude. The parameters above are the relevant terms in equation 3.7 and 3.8.

To achieve precise coordinates, the error sources should be removed or reduced. The most significant error is the delay due to the receiver clock error, which must be determined as the fourth unknown in the GPS positioning equation 3.4. As an alternative, the receiver clock error is eliminated by using the DD approach.

Secondly, the satellite clock bias has to be handled. The GPS control segment comprises of a global network of monitor stations. The monitor stations are equipped with atomic clocks and GPS receivers for collecting measurements. The monitor stations provide these measurements to the master control station. The master station can predict the behavior of each satellite's orbit and clock by means of these measurements, and broadcast it via the navigation message [Adm] and [GMV11a]. A more precise choice are the IGS products, which can usually provide the final orbit (2.5 cm accuracy) and final satellite clock correction (75 ps accuracy) about 1 day later, or in advance with slightly reduced accuracy. Moreover, the DD approach is also an alternative to eliminate the satellite clock error.

Next, the electromagnetic signals of GPS Satellites have to travel through the Earth's atmosphere (the ionosphere and the troposphere). In these Earth's atmospheric layers, the signals are delayed and refracted. The signal of a satellites with low elevation angle is stronger affected by the atmospheric layers, and the signal path of such satellite is longer. The signal path of a satellite with low elevation angles is about 10 times longer than for the signal from a satellite close to the zenith direction [Gla22].

3.3.1. Tropospheric delay

The troposphere is the lowest layer of Earth's atmosphere, which mainly contains electrically neutral particles. Hence, the GPS signal has to travel through the neutral atmosphere. In this layer, the delay varies from zenith towards horizon between 2 and 25 meters. The delay size depends on the elevation angle of the satellite and the height of the station on the ground. This layer contains about 80% part of the atmosphere, and is a non-dispersive medium for microwaves. Therefore, the delay in this layer is frequency-independent for GPS signals, and has the same size for code and phase measurements [Gla22]. Thus, the delay can not be reduced by using a multi-frequency linear combination. Next, the index of refraction in the neutral atmosphere is a little over 1 in the microwaves spectrum. Therefore, the GPS signals of satellites have to face with a delay, but the signals do not bend in this layer [Nil+13]. There are various models which have been developed to minimize the troposphere delay. Generally, the hydrostatic (or dry) part contributes about 90% of the total delay to the tropospheric delay, and the non-hydrostatic (or wet) part contributes about 10% of the total delay. Atmospheric gases cause the dry part,

and this part can be modeled with the help of the surface air pressure and temperature. A hydrostatic model can be used to estimate it accurately, such as the Saastamoinen model. The wet part depends on water vapor content and is difficult to model. The reason is that the water vapor is highly variable with the location and time, the typical values for the wet zenith delay varies between a few mm (poles) and about 40 cm (equator region).

$$\delta_{tropo} = m_d \cdot d_{dry} + m_w \cdot d_{wet}, \quad (3.10)$$

where:

m_d = the dry-component mapping function

m_w = the wet-component mapping function

d_{dry} = the dry zenith delay

d_{wet} = the wet Zenith delay

The Saastamoinen model is used to estimate the path delay of the electro-magnetic signal of a satellite, which passes through the troposphere. This model was derived based on ideal gas law, and assuming the gravity acceleration as a function of height with hydrostatic equilibrium [Qia16]. The dry zenith delay can be obtained by the following formula:

$$d_{dry}[m] = \frac{0.002277}{\cos Az} \left(O + \left(\frac{1255}{T} + 0.05 \right) e - \tan^2 Az \right). \quad (3.11)$$

where:

O = the atmospheric pressure (millibar)

e = the partial water vapor pressure (millibar)

T = the temperature [C°]

Az = the satellite zenith angle [°]

The parameters at the station with the height h are obtained by using standard atmosphere model:

$$O = O_0 \times (1 - 0.0000226 \times (h - h_0))^{5.225}, \quad (3.12)$$

$$T = T_0 - 0.0064 \times (h - h_0), \quad (3.13)$$

$$Rh = Rh_0 \times \exp(-0.0006396 \times (h - h_0)), \quad (3.14)$$

$$w = Rh \times \exp(-37.2465 + 0.213166T - 0.000256908T^2). \quad (3.15)$$

where h_0 is the reference height (usually 0 meter). The other reference parameter are assigned as follows:

$$O_0 = 1013.25 \text{ [mb]}, \quad T_0 = 18 \text{ [}^\circ\text{C]}, \quad Rh = 50 \text{ [%]}. \quad (3.16)$$

Furthermore, the elevation angles of satellites and the azimuth play a role in the mapping function.

$$m_d(\theta, \psi) \text{ or } m_w(\theta, \psi) = m_{f_0}(\theta) + \delta m_f(\theta, \psi). \quad (3.17)$$

The first term of the mapping function m_d and m_w can convert the delay from the zenith direction into the direction of the signals. Next, the tropospheric delay depends on the horizontal direction due to specific weather phenomena [Gla22].

3.3.2. Ionospheric delay

The ionospheric layer is part of Earth's upper atmosphere ranging from 50 km to about 2000 km altitude, which contains a large number of ions and free electrons. This number is large enough to affect the propagation of the electromagnetic signals from the GPS satellites. The number of free electrons periodically changes over time (e.g. daily, seasonally, sun cycle) and depends on the position, the solar radiation produces the ionization and the number of free electrons [Gla22]. For example, the ionosphere changes from day to night on the Earth. The ionosphere can cause a signal path delay from several meters up to 100 meters in special cases.

The ionospheric layer is a dispersive medium for microwaves, thus the refraction in the ionosphere depends on the frequency of the signals [See03]. The propagation speed of GPS signal can be distinguished between the phase and group velocity. On the one hand, the phase velocity is important for the phase measurements, on the other side, the group velocity is important for the code measurements.

Most of ionospheric delay can be eliminated by building the ionospheric free linear combination of code or carrier measurements using dual-frequency observations. The code

ionospheric free linear combination reads [Gla22]:

$$P(k)_{r,IF}^s = \frac{f_1^2 P_1 - f_2^2 P_2}{f_1^2 - f_2^2}. \quad (3.18)$$

Similarly, the ionosphere-free linear combination of the GPS carrier-phase measurements can be formed as follows:

$$\phi(k)_{r,IF}^s = \frac{f_2^2 L_2 - f_1^2 L_1}{f_2^2 - f_1^2}. \quad (3.19)$$

The combination can remove the first order up to 99.9% of the effects. The remaining 0.1% of the ionospheric refraction are still present which causes a range error in the GPS measurements of only a few centimeters or even less [MMA11]. For baselines longer than 50km, the remaining term has to be accounted in the data processing. For a baseline, less than 50km, the DD approach is used to remove the remaining terms [Kap06]. Usually, the PPP approach uses code and carrier-phase measurements in the ionosphere-free combination to remove the ionospheric error, because this effect is challenging to model accurately [SZ11].

3.3.3. Other significant error sources (Multipath Signal Biases)

Other significant errors sources in the pseudo-range and carrier-phase measurements are the errors due to multi-path and receiver noise. The multi-path is an uncorrelated error in the GPS observation equation [Sic20b]. This phenomena occurs, when part of the signals from the satellites reach the receiver after more reflections by nearby objects or the ground near the antenna. These reflected signals can interfere with the signal which is received at the GPS antenna of the satellite. Usually, reflected signals have weak signal strength and the range measurements are always longer [Gla22]. The magnitude of this error depends mainly on the reflection of the reflected surface near the antenna, which means that the error is dependent on the location environment near the GPS antenna. Hence, it is difficult to determine this error. Code multipath can reach up to several tens of meters, while phase multipath only a quarter of a wavelength.

The satellite elevation angle, the receiver signal processing and the antenna gain pattern are the other factors. The receiver noise depends on the signal-to-noise ratio (SNR) of the GPS signal. The SNR is defined as the ratio between the signal power to the power of the ambient noise [Jek01]. When the noise caused by the receiver is low, then the SNR value of the

signal is better.

3.4. Pseudo-range rates and Doppler shift observations

The Doppler shift measurements can be utilized to calculate the velocity of the sensor. This effect describes the change in signal frequency due to the relative motion between the GPS receiver and the GPS satellite. Hence, the Doppler shift of the signal is generated based on this relationship between transmitted and observed frequencies:

$$\delta_{f_D} = f_r - f_t = -f_t \frac{(v_s - v_r) \cdot e}{c}, \quad (3.20)$$

where:

δ_{f_D} = the Doppler shift measurement

f_r = the received signal frequency

f_t = the satellite signal frequency

v_s = the velocity of the observed satellite

v_r = the velocity of the sensor

e = the unit vector along the line of sight from receiver to satellite

The dot product $(v_s - v_r) \cdot e$ represents the radial component of the relative velocity vector along the line of sight to the satellite. Sometimes, the Doppler shift measurements can also be represented as pseudo-range rate measurements. The relativistic effects and atmospheric propagation delay can be neglected. The pseudo-range rates are obtained from the observed Doppler shift measurements. Typically, the Rinex files provide the Doppler shift measurements, which are usually denoted as D. D1 and D2 represent the Doppler frequencies on L1 and L2 (Hertz). The equation for calculating the pseudo-range rates reads as follows [Gro13]:

$$\dot{\rho}_r^s = -\left(\frac{\delta_{f_D}}{f_{ca}} + \delta \dot{t}_r - \delta \dot{t}_s\right)c. \quad (3.21)$$

where:

$\dot{\rho}$ = the pseudo-range rate measurement

f_{ca} = the carrier frequency

$\delta \dot{t}_r$ = the receiver clock drift

$\delta \dot{t}_s$ = the satellite clock drift

Additionally, the receiver clock and satellite clock are affected by a frequency drift.

The receiver generates the frequency of the signal received from a satellite, which provides a bias due to drift rate, $\delta\dot{t}_r$, of the receiver clock relative to the GPS time. $\delta\dot{t}_s$ is the clock drift of the satellite.

An alternative method to obtain the pseudo-range rates is based on the position of the sensor and the visible satellites. The pseudo-range rate are the rates of change of the pseudo-ranges, after differentiating the equation 3.3,

$$\dot{\rho}_r^s = [(V_{x_s} - V_{x_r}) + (V_{y_s} - V_{y_r}) + (V_{z_s} - V_{z_r})] \cdot e. \quad (3.22)$$

Thus, the velocity difference between each visible satellite and the GPS receiver along the line of sight vector is used to calculate the pseudo-range rates. Furthermore, the pseudo-ranges and pseudo-range rates between the receiver and each visible satellite are basic components of the tightly coupled sensor integration.

3.5. Double difference approach

The GPS receiver has to face different error sources (described in subsection 3.3). To remove these error terms, the simultaneous measurements of two or more different receivers to one or more different GPS satellites can be combined, well-known as differential GPS positioning. Differential GPS positioning belongs to relative positioning, which can be used either in static or kinematic situations. There are various types of differential GPS positioning. Single difference (SD) is performed, if the simultaneous measurements are combined from at least two receivers to one GPS satellite. A rover receiver and a base receiver are separated with a fixed distance in static processing or some flexible distance in kinematic processing called a baseline. When the baseline between both receivers is short, both receivers reveal similar atmospheric or orbital errors. If the length of the baseline increases, the difference in the error terms at both receivers simultaneously increases. The combination of simultaneous measurements from both receivers is formed to the same satellite, which removes the error terms that are common in the measurements from this satellite to both receivers. This approach allows carrier phase positioning with cm-level accuracy with a baseline up to 10 km and code pseudo-range positioning with m-level accuracy with a baseline of 100 km [Sic20a].

The SD approach also eliminates the satellite clock error. The simultaneous measure-

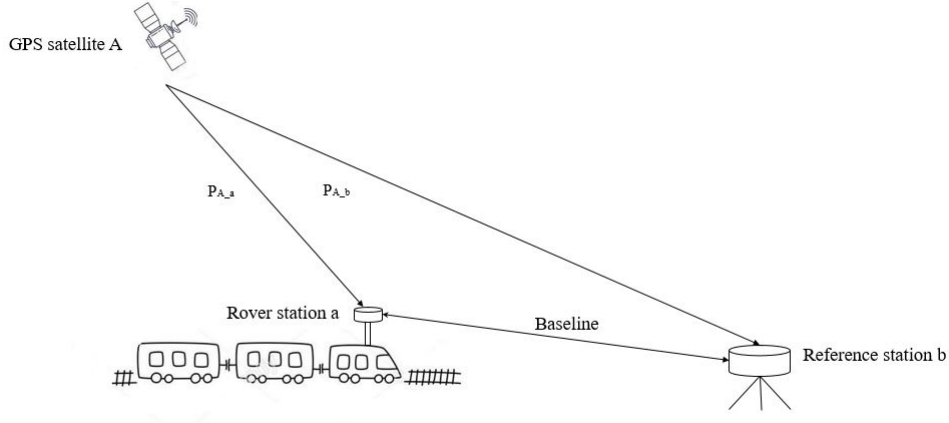


Figure 4: A SD approach between the rover and base station

ments from two receivers a,b to the common GPS satellite A are:

$$\begin{aligned} P(k)_a^A &= \rho(k)_a^A + c\delta_a - c\delta^A + \delta t_{D_{P_a}}^A + \epsilon_{r_a P}^A \\ P(k)_b^A &= \rho(k)_b^A + c\delta_b - c\delta^A + \delta t_{D_{P_b}}^A + \epsilon_{r_b P}^A, \end{aligned} \quad (3.23)$$

The GPS receivers generate the pseudo-range and carrier phase measurements at each epoch and for all visible satellites being actively tracked for each epoch. The SD pseudo-range measurements can be formed as the difference between the simultaneous measurements of two receivers a and b:

$$P(k)_{a,b}^A = P_b^A - P_a^A = \rho(k)_{a,b}^A + c\delta_{a,b} + \delta t_{D_{P_{a,b}}}^A + \epsilon_{a,b P}^A, \quad (3.24)$$

and

$$\rho(k)_{a,b}^A = \rho(k)_b^A - \rho(k)_a^A \quad (3.25)$$

The carrier-phase measurements with the SD approach can be formed as follows:

$$\begin{aligned} \lambda\phi(k)_{a,b}^A &= \lambda\phi_b^A - \lambda\phi_a^A \\ &= \rho(k)_{a,b}^A + c\delta_{a,b} + \delta t_{D_{\phi_{a,b}}}^A + \lambda(N_{a,b}^A + \phi_{a,b} - \phi_{a,b}^A) + \epsilon_{a,b\phi}^A \end{aligned} \quad (3.26)$$

The double difference approach (DD) is based on the difference of two SD measure-

ments:

$$\begin{aligned} P(k)_{a,b}^A &= \rho(k)_{a,b}^A + c\delta_{a,b} + \delta t_{DP_{a,b}}^A + \epsilon_{a,bP}^A \\ P(k)_{a,b}^B &= \rho(k)_{a,b}^B + c\delta_{a,b} + \delta t_{DP_{a,b}}^B + \epsilon_{a,bP}^B, \end{aligned} \quad (3.27)$$

Double differencing is used to remove the receiver clock bias. The tropospheric and ionospheric delays are still present, but these effects strongly correlate with the baseline length. If the rover station's receiver and base station's receiver are relatively close to each other, the differencing approach can remove most of the error sources. The ionospheric delay is usually eliminated or mitigated using the ionosphere-free linear combination, if the GPS data comes from a dual-frequency GPS receivers. It has to be noted that the linear combination does usually not allow for integer ambiguity resolution. The multi-path error can only be mitigated, but not totally removed. Furthermore, the receiver noise is present.

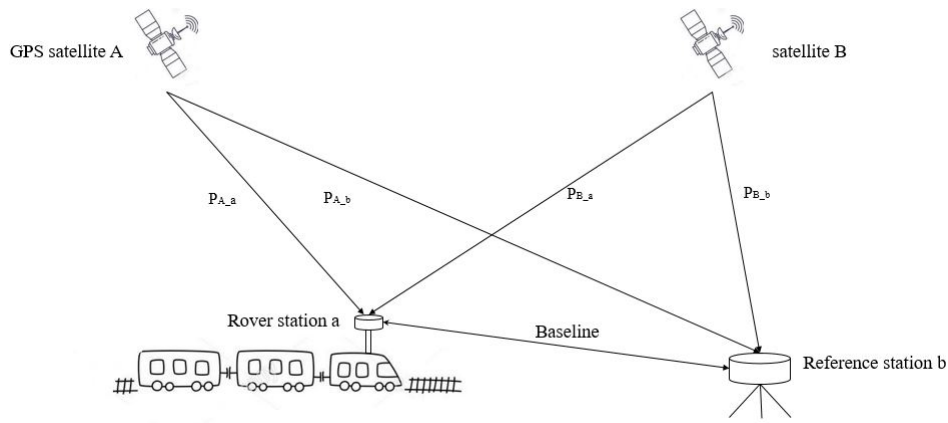


Figure 5: Double difference approach between the rover and base station

The DD code pseudo-range observation equation can be formed as follows:

$$\begin{aligned} P(k)_{a,b}^{A,B} &= (P_a^A - P_a^B) - (P_b^A - P_b^B) \\ &= \rho(k)_{a,b}^{A,B} + \delta t_{a,bDP}^{A,B} + \epsilon_{a,bP}^{A,B}, \end{aligned} \quad (3.28)$$

The DD carrier-phase observation equation can be formed as follows:

$$\begin{aligned}\lambda\phi(k)_{a,b}^{A,B} &= (\lambda\phi_a^A - \lambda\phi_a^B) - (\lambda\phi_b^A - \lambda\phi_b^B) \\ &= \rho(k)_{a,b}^{A,B} + \delta t_{a,bD_\phi}^{A,B} + \lambda(N_{a,b}^{A,B} + \phi_{a,b} - \phi_{a,b}^{A,B}) + \epsilon_{a,b\phi}^{A,B}\end{aligned}\quad (3.29)$$

where

$\phi_{a,b}^{A,B}$ = the combined carrier phase measurements made at receivers a and b to satellites A and B

B

$N_{a,b}^{A,B}$ = the combined unknown ambiguities

$m_{a,b}^{A,B}$ = the combined multi-path error

By means of the estimated baseline components and the known reference station coordinates, the rover site can be calculated.

3.6. Phase smoothed code

The GPS code and carrier-phase measurements suffer from similar errors, but the carrier-phase measurement provides a more accurate positioning solution. The ambiguous carrier phase measurements provide a relatively low observation noise (less than 1 cm). The code measurements does not contain any carrier-cycle ambiguities. However, the code measurements can not be measured as precise as than the carrier-phase measurement due to its noise (a few dm). Figure 6 describes the difference between the noises of code and carrier-phase measurement noise. Hence, the geodetic applications require high-accuracy positioning based on the processing with the carrier-phase or combined measurements. The GPS signals between the satellites and a moving receiver sometimes fail. There are various disturbing factors such as signal blockage by buildings, a tunnel and by vehicle dynamics. Strictly speaking, the receiver phase tracking loops face a temporary loss of lock due to the signal blockage. Such signal loss causes a discontinuity of the integer number of cycles in the measured carrier phase, which is known as a cycle slip. Simply speaking, a cycle slip occurs and causes a phase jump in carrier-phase measurements. When a cycle slip occurs, the Doppler counter in the receiver restarts. The cycle slip suddenly causes a jump in the accumulated phase measurements by an integer number of cycles [Lan14]. In contrast, the GPS code measurements are not affected by cycle slips, and there is no need to solve for ambiguities when processing code range data.

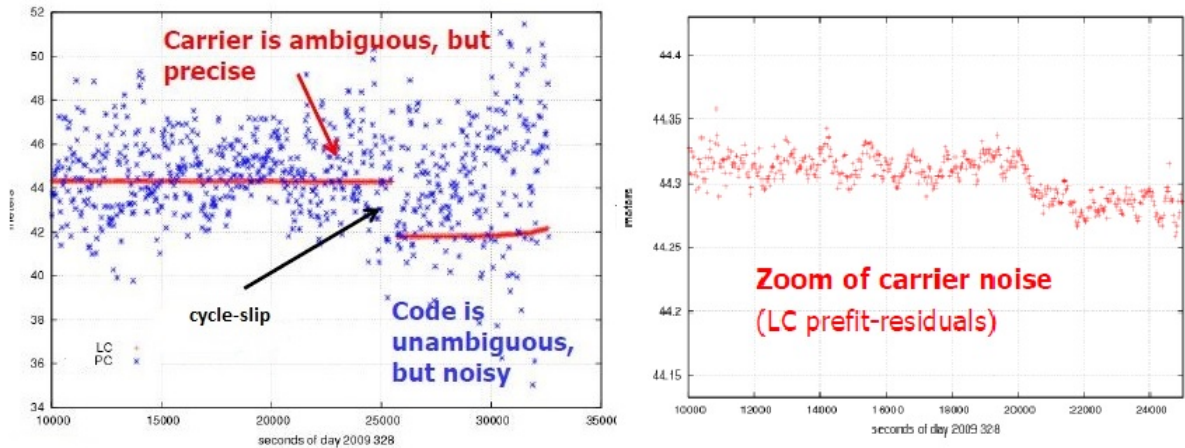


Figure 6: Comparison of code and carrier-phase measurement noise [SZG13]

The phase smoothed code (PSC) approach is applied for reducing the noise of the code measurement to improve the accuracy and the stability of the code-only processing. The PSC algorithm for dual frequency measurements is defined as follows [SZP11]:

$$\hat{P}(k)_r^s = \frac{1}{n}P(k)_r^s + \frac{n-1}{n}(\hat{P}(k-1)_r^s + \lambda(L(k)_r^s - L(k-1)_r^s)). \quad (3.30)$$

This approach is based on the combination of the code and carrier-phase measurements. $P(k)_r^s$ and $L(k)_r^s$ are the code and carrier-phase measurements made by receiver r to the satellite s at epoch k . λ is the wavelength. $\hat{P}(k)_r^s$ denotes the smoothed code measurements in meter. Here, n is the smoothing weight factor. The smoothed code measurements at epoch (k) are based on using the estimated code measurements from the previous epoch ($k-1$) and the difference of the carrier-phase measurements between the last epoch and the current epoch. The differences between both carrier-phase measurements across an epoch removes the integer ambiguity, so that the smoothed code measurements remain unambiguous. At the first epoch the weight factor is set to one. Here the code measurements has the full weight. For the next consecutive epochs, the carrier-phase measurements obtain higher weights, on the other hand, the weight of code measurements is decreased [Ber+10]. The disadvantage of this approach is that the algorithm has to be initialised each time when a carrier phase cycle-slip occurs. It means that the initialised epoch for the smoothed code measurements contain noise. The initialization of the approach works with:

$$\hat{P}(1)_r^s = P(1)_r^s. \quad (3.31)$$

An example for a PSC time series is displayed in figure 7. The original raw code measurements P are quite noisy, which are represented by the green dots. The blue line shows the carrier-phase measurements L , which are more stable and less noisy. The differences between P and L originate from the phase biases. The red line represents the smoothed code measurements \hat{P} time series taking the advantages from both measurement types.

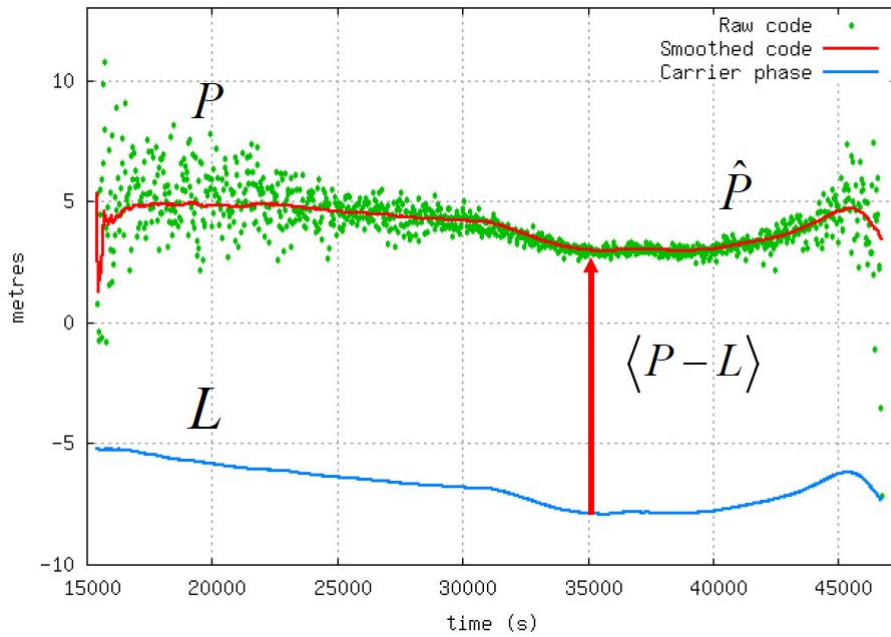


Figure 7: Phase smoothed code [SZG13]

Figure 8 represents the differences between the raw code measurements and the smoothed code measurements. The observed code data come from the 2nd test drive scenario, which are described in chapter 7.2. The subfigures display the differences between both measurements at C1, C2, P1 and P2 of a visible satellite number 24 obtained by a rover (a moving car). The L1 carrier-phase measurements are used to produce the smoothed C1 and P1 ranges, and L2 for the C2 and P2 ranges.

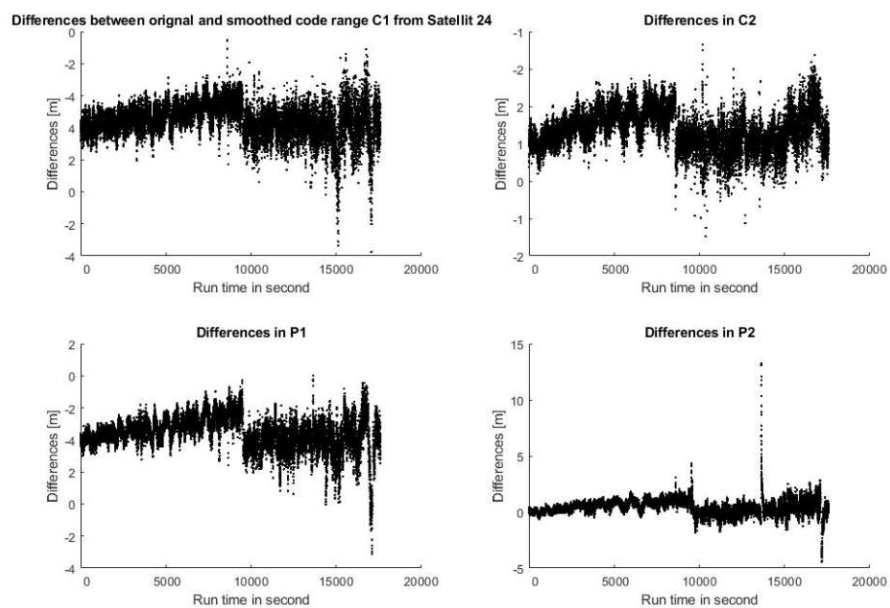


Figure 8: Differences between the raw code and smoothed code measurements (e.g. PRN 24)

4. Basics of Inertial Navigation and Inertial Sensor

Inertial navigation enables the user to determine the navigation solution of a moving sensor by using sensors that react on the Newton's law of motion [Jek01]. This kind of navigation approach does not belong to absolute positioning, but rather to a relative positioning such as a dead reckoning navigation system. This positioning approach determines the position of some points relative to the absolute coordinate of a starting point. The start point provides a known initial position, the relative distance to the next new point are added continuously. The determination of the subsequent points always relies on the previous determined point position [HLW03]. An inertial measurement unit (IMU) is an electronic instrument with the self-contained technique of the inertial navigation system, which measures about six degrees of freedom, including three rotational motions and three translational motions. Typically, an IMU is composed of a 3D motion sensor accelerometer and a 3D rotation sensor gyroscope that senses the specific force and the angular rate of the sensor in space. The position solution is derived by integrating velocity, in turn, the velocity is derived by integrating the measured acceleration. The coordinate axes are controlled by monitoring their angular motion. The attitude can be obtained by integrating IMU's angular rates [Gro13].

IMU measurements, representing 3D vectors of accelerations (translational motion) and angular rates (rotational movement), normally refer to the inertial frame. Even when an IMU is stationary, the sensors still measure forces. The Earth's gravity is sensed on the vertical axis of the accelerometer (about 9.8 m/s^2). In other words, an accelerometer in free fall provides zero values [Web21]. If the IMU is moving forward, the accelerometers measure this additional accelerations acting on the sensor [CS14]. The angular rate measurements are associated with the Earth rotation, and the sensed 3-axis angular rates are aligned with the Earth's spin axis. When the IMU is at rest, the rotation vector can be expressed as $w = (0, 0, w_e)^t$ [Jek01]. w_e is well-known as the angular rate of the Earth rotation. To determine the IMU's motion, gravitational and rotational forces of the Earth have to be removed from the measurements, so that only inertial forces stemming from the sensors motion remain.

The Inertial navigation system has been applied for the military since 1960s and 1970s. In the mean time, low-cost IMU devices also became increasingly popular for civil applications. The IMU data processing can typically be performed as standalone processing or as part of an multi-sensor integrated navigation system [Gro13] (see chapter 7).

This chapter firstly introduces the IMU strap-down mechanization. Secondly, the navigation equations in three different frames are described. At the end, the systematical errors

of an IMU device are presented.

4.1. IMU strap-down mechanization

There are mainly two design types of IMU: the stabilized and the strap-down systems. In the stabilized mechanization, the accelerometers and the gyroscopes are mounted on the stable element of the platform. They are isolated from the rotational dynamics of the vehicle. The stable element is connected to the vehicle through a set of gimbals. Each gimbal allows rotation about a particular axis. Angular motion of the vehicle can be isolated by the gimbals themselves and the inertia of the stable element [Jek01]. The gyroscopes mounted on the stable element provides the amount of rotation of the platform, which is needed to isolate the platform from the vehicle motion. The gyroscopes drive servomotors of gimbals to keep the platform at a fixed orientation relative to inertial space, and the gimbal angles are zero. Therefore, the dynamic range of the gyroscopes can usually be kept low since the output of gyroscopes is always close to zero.

A strap-down mechanization uses the combination of accelerometers and gyroscopes, which are rigidly mounted at a platform on the body, to determine the navigation solution of the body, or more explicitly, the IMU sensor. Nowadays, strap-down mechanization is widely used for determining the navigation solution. In this mechanization, the accelerations are measured with respect to the origin of the i-frame, while they are given in the b-frame. The sensed accelerations, f_{ib}^b , are analytically transformed from the b-frame to the navigation frame (e.g. n-frame) by using:

$$f_{ib}^n = C_b^n f_{ib}^b, \quad (4.1)$$

where

C_b^n = the transformation matrix from the b-frame to the n-frame

f_{ib}^n = the acceleration vector in n-frame

The transformation matrix comprises the Euler angles, which are the Roll, Pitch and Yaw angles (α, β, γ) . C_b^n can be obtained according to the specific order of angle rotation (see equation 2.35) [Jek01].

The significant differences between the stabilized and the strap-down mechanizations concern the transformation from the accelerometer frame to the n-frame. The stabilized mechanization has a fixed orientation in the i-frame, and the accelerometers sense the accelerations

directly in the i-frame. This means that stabilized mechanization defines the i-frame as the navigation frame for the accelerometers mechanically. In the strap-down mechanization, the transformation matrix C_b^n has to be defined by using sensed accelerations and the calculated angular rates in the n-frame to convert the sensed accelerations to the n-frame.

A significant advantage of the strap-down mechanization is that the measurements from gyroscopes bring additional information about the spatial orientation of the sensor. Moreover, a strap-down mechanization benefits from the smaller size and weight, the lower power consumption, the lower price, and the simplified maintenance.

4.2. Navigation equations

The IMU accelerometer and gyroscope measurements are processed based on the navigation equations. These can estimate the trajectory of the vehicle by integration with initial information. The navigation equations can be described in different reference frames. Four important reference frames were already described in chapter 2. They are commonly used in inertial navigation comprising of the inertial frame (i-frame), the Earth-fixed frame (e-frame), the local level frame (n-frame) and the body frame (b-frame).

Updating the navigation solution of the sensor by using an inertial navigation system is based on the combination of the double integration of the sensed accelerations and the single integration of sensed angular rates over a time interval. In this subsection, the sensed acceleration vector of the IMU, f_{ib}^b , is denoted as a , to understand the navigation equations better and easier. The accelerations can be mathematically defined as the second-order time-derivatives of position [Jek01]. In the inertial frame this equation can be formulated as follows (see equation 2.3).

$$\ddot{x}^i = g^i(x^i) + a^i, \quad (4.2)$$

where

x^i = the position vector in the i-frame

g^i = the accelerations due to the gravitational field in the i-frame

a^i = the measured specific forces

\ddot{x}^i = the acceleration vector in the i-frame

The gravitational accelerations depend on the sensor's position, and can be derived from well-known models. Now, the equation can be easily solved. The navigation equations used for updating the navigation solution of the sensor can be formulated in the different frames.

The formulation and how to determine the solution of navigation equations is different for each coordinate frame, which is desired to represent the navigation solution.

In the strap-down mechanization, the navigation equations may be solved in one of the reference frames. Generally, the equations in the arbitrary a-frame can be formulated after first and second integration [Jek01]:

$$\frac{d}{dt}\dot{x}^a = -2\Omega_{ia}^a\dot{x}^a - (\dot{\Omega}_{ia}^a + \Omega_{ia}^a\Omega_{ia}^a)x^a + a^a + g^a, \quad (4.3a)$$

$$\frac{d}{dt}x^a = \dot{x}^a, \quad (4.3b)$$

where Ω denotes a skew-symmetric matrix with components from the sensed data of gyroscopes. The specific force vector a describes the difference between the true accelerations in space and the accelerations due to gravity [TW96], and g is the gravitational accelerations. The remaining terms in equation 4.3a refer to the Coriolis, rotational and centrifugal apparent forces.

4.2.1. Navigation equations in the i-frame

Firstly, let us look at the body frame that rotates with respect to the i-frame. A position vector in the i-frame can be converted to the coordinates in the i-frame as follows [Jek01]:

$$x^i = C_b^i x^b, \quad (4.4)$$

where

C_b^i = the transformation matrix between the b-frame and i-frame

Thus, the accelerations measured by a strap-down mechanization can be transformed from the b-frame to the i-frame

$$a^i = C_b^i a^b. \quad (4.5)$$

In the i-frame, ω_{ii}^i is equal to zero. Therefore, the navigation equations (see equations 4.3a and 4.3b) represented in the i-frame can be formulated after first and second integration

[Jek01]:

$$\frac{d}{dt}\dot{x}^i = a^i + g^i, \quad (4.6a)$$

$$\frac{d}{dt}x^i = \dot{x}^i. \quad (4.6b)$$

In the strap-down mechanization, the transformation matrix C_b^i from b-frame to i-frame is obtained from the IMU sensed angular rate measurements w_{ib}^b . Now, the transformation matrix C_b^i is determined by integrating the time derivative of the transformation matrix \dot{C}_b^i (see Equation 2.17). This is the change of the relative orientation with time. This matrix can be derived with respect to time as follows:

$$\dot{C}_b^i = C_b^i \Omega_{ib}^b. \quad (4.7)$$

w_{ib}^b and Ω_{ib}^b describe the rotation of the b-frame relative to the i-frame.

$$\Omega_{ib}^b = [w_{ib}^b \times]. \quad (4.8)$$

4.2.2. Navigation equations in the e-frame

For most navigation applications, the final position shall be established in an Earth fixed frame. However, in an GNSS/IMU integrated system, it becomes complicated to use the e-frame as the reference frame for the navigation solution. If the angular rate and acceleration measurements of the IMU in the i-frame are used to update the navigation solution in an Earth reference frame, the transformation matrix of the reference frame with respect to the inertial frame brings more complexity.

In the e-frame, $\dot{\Omega}_{ie}^e$ is equal to zero. Therefore, the navigation equations (see equations 4.3a and 4.3b) in the e-frame can be formulated as follows [Jek01]:

$$\frac{d}{dt}\dot{x}^e = -2\Omega_{ie}^e \dot{x}^e - \Omega_{ie}^a \Omega_{ie}^e x^e + a^e + g^e, \quad (4.9a)$$

$$\frac{d}{dt}x^e = \dot{x}^e. \quad (4.9b)$$

where Ω_{ie}^e is the symmetric matrix of the earth rotation vector w_{ie}^e , which can be assumed as constant. Here, w_{ie}^e can be described by $(0, 0, w_e)^T$.

$$\Omega_{ie}^e = \begin{pmatrix} 0 & -w_{ie}^e & 0 \\ w_{ie}^e & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \quad (4.10)$$

In the IMU strap-down mechanization, the transformation matrix C_b^e can be obtained by integrating the time derivative of the transformation matrix \dot{C}_b^e .

$$\dot{C}_b^e = C_b^e \Omega_{eb}^b. \quad (4.11)$$

4.2.3. Navigation equations in the n-frame

As mentioned before, the navigation equations of a strap-down mechanization may be solved in an arbitrary reference frame. Practically, the processing bases always on the solution in the n-frame. A convenient way to derive the navigation equations in the n-frame is firstly to develop the specific-force equation in the e-frame, and transform this solution to the n-frame later on [HLW03]. To realize the navigation equations in the n-frame, an Earth fixed frame is usually defined as the reference frame, while the resolving axes are comprised by the n-frame (North, East, and Down directions) [Gro13]. The sensor's motion can be conveniently represented in the e-frame, e.g. in latitude, longitude and height. The velocity vector is given in a frame parallel to the n-frame [Jek01]. Attitude is expressed as a coordinate transformation from the b-frame to the n-frame.

The desired form of the n-frame navigation equations is defined as follows [Jek01]:

$$\frac{d}{dt} v^n = a^n - (\Omega_{in}^n + \Omega_{ie}^n) v^n + g^n, \quad (4.12a)$$

$$\frac{d}{dt} x^e = C_n^e v^n. \quad (4.12b)$$

Firstly, v^n and \dot{x}^n do not have the same definition. On the one hand, v^n is the vector covering the velocity components. While on the other hand, \dot{x}^n is the time-derivative of the vector x^n . Here, \ddot{x}^n is the second time-derivative of the x^n . Moreover, g^n is the gravity vector, which comprises from the gravitational vector and the centrifugal accelerations.

The differential equation for the position vector x^e is defined in a geodetic coordinate system. The transformation between the geodetic coordinates system and the Cartesian coordinate system can be shown as follows:

$$x^e = \begin{pmatrix} (N + h)\cos\varphi\cos\lambda \\ (N + h)\cos\varphi\sin\lambda \\ (N(1 - e^2) + h)\sin\varphi \end{pmatrix}, \quad (4.13)$$

where

e^2 = the square of the eccentricity of the ellipsoid associated with geodetic coordinates

$$(e^2 = 2f - f^2)$$

N = the Earth's transverse radius of curvature

M = the Earth's meridional radius of curvature

h = the ellipsoidal height

$$N = \frac{a}{\sqrt{1 - e^2\sin^2\varphi}}, \quad (4.14)$$

$$M = \frac{a(1 - e^2)}{(1 - e^2\sin^2\varphi)^{3/2}}, \quad (4.15)$$

where

a = the Earth's equatorial radius

$\dot{\lambda}$ and $\dot{\varphi}$ are the changes in longitude and latitude of the sensor moving parallel to the ellipsoidal surface [Jek01]. The rate of φ for representing the velocity during the sensor's motion along the meridian is $1/M$. The rate of λ during the sensor's motion along the surface normal to the meridian is $1/N$. The equation 2.33 defines the transformation matrix C_n^e , v^n can be obtained as follows:

$$v^n = \begin{pmatrix} \dot{\varphi}(M + h) \\ \dot{\lambda}(N + h)\cos\varphi \\ -\dot{h} \end{pmatrix}. \quad (4.16)$$

The velocity components of v^n refer to the Earth-fixed vector \dot{x}^e , but represented in the NED directions in the n-frame. For example, the north velocity is the local rate of change in position along the tangent to the meridian to north direction.

The term $\Omega_{in}^n + \Omega_{ie}^n$ in equation 4.12a is given by:

$$\Omega_{in}^n + \Omega_{ie}^n = \begin{pmatrix} 0 & (\dot{\lambda} + 2w_e)\sin\varphi & -\dot{\varphi} \\ -(\dot{\lambda} + 2w_e)\sin\varphi & 0 & (\dot{\lambda} + 2w_e)\cos\varphi \\ \dot{\varphi} & (\dot{\lambda} + 2w_e)\cos\varphi & 0 \end{pmatrix}, \quad (4.17)$$

Now, the formula of time-derivative of the velocity vector v^n with $(v_N, v_E, v_D)^T$ can be derived as follows:

$$\frac{d}{dt}v^n = \begin{pmatrix} a_N + g_N - 2w_e\sin\varphi v_E + \dot{\varphi}v_D - \dot{\lambda}\sin\varphi v_E \\ a_E + g_E + 2w_e\sin\varphi v_N + 2w_e\cos\varphi v_D + \dot{\lambda}\sin\varphi v_N + \dot{\lambda}\cos\varphi v_D \\ a_D + g_D - 2w_e\cos\varphi v_E - \dot{\lambda}\cos\varphi v_E - \varphi v_n \end{pmatrix}. \quad (4.18)$$

The time derivative of the sensor's position $(\dot{\varphi}, \dot{\lambda}, \dot{h})$ can be formulated as follows [Gro13]:

$$\begin{pmatrix} \dot{\varphi} \\ \dot{\lambda} \\ \dot{h} \end{pmatrix} = \begin{pmatrix} \frac{v_N}{(N+h)} \\ \frac{v_E}{(N+h)\cos\varphi} \\ -v_D \end{pmatrix}. \quad (4.19)$$

Now, the six differential equations from 4.18 and 4.19 (obtained from the equation 4.16) can be used to determine the Earth-referenced position and velocity of the sensor in the n-frame [Jek01]. In the strap-down mechanization, C_b^n is usually expressed as attitude, which is obtained by integrating the time derivative of the transformation matrix \dot{C}_b^n :

$$\dot{C}_b^n = C_b^n \Omega_{nb}^b = C_b^n (\Omega_{ib}^b - \Omega_{in}^n), \quad (4.20)$$

and

$$\Omega_{in}^n = (\Omega_{ie}^n + \Omega_{en}^n). \quad (4.21)$$

Ω_{ib}^b are the angular rates measured by the gyroscopes. Ω_{ie}^n is Earth's rotation with

respected to the i-frame, which can be formulated as follows:

$$\Omega_{ie}^n = w_{ie}^n \times = \begin{pmatrix} w_e \cos \varphi \\ 0 \\ -w_e \sin \varphi \end{pmatrix} \times = w_e \begin{pmatrix} 0 & \sin \varphi & 0 \\ -\sin \varphi & 0 & -\cos \varphi \\ 0 & \cos \varphi & 0 \end{pmatrix}. \quad (4.22)$$

Ω_{en}^n is known as the transport rate, which is caused by rotation of the n-frame relative to Earth, while the sensor moves. When the attitude of the b-frame with respect to the n-frame remains constant, the gyroscopes sense the Earth rotation and the transport rate, which have to be corrected, so that the attitude can not be changed [Gro13]. The elements of Ω_{in}^n can be given by using calculated parameters $(\varphi, \lambda, h, v_N, v_E, v_D)$ [Shi01]. The turn rates of the rotation vector in the n-frame with respect to the e-frame are usually expressed as change of latitude and longitude [TW96]:

$$w_{en}^n = \begin{pmatrix} \dot{\lambda} \cos \varphi \\ -\dot{\varphi} \\ -\dot{\lambda} \sin \varphi \end{pmatrix}. \quad (4.23)$$

Collecting the first two components of the equation 4.19, the above equation can be expressed as:

$$w_{en}^n = \begin{pmatrix} \frac{v_E}{(N+h)} \\ \frac{-v_N}{(N+h)} \\ \frac{-v_e \tan \varphi}{N+h} \end{pmatrix}. \quad (4.24)$$

and

$$\Omega_{en}^n = w_{en}^n \times = \begin{pmatrix} 0 & -w_{e,3} & w_{e,2} \\ w_{e,3} & 0 & -w_{e,1} \\ -w_{e,2} & w_{e,1} & 0 \end{pmatrix}. \quad (4.25)$$

Moreover, the angular rates of the n-frame relative to i-frame can be formulated as follows [Shi05]:

$$w_{in}^n = w_{ie}^n + w_{en}^n = \begin{pmatrix} \frac{v_E}{(N+h)} + w_e \cos \varphi \\ \frac{-v_N}{(N+h)} \\ (-w_e) \sin \varphi - \frac{-v_e \tan \varphi}{N+h} \end{pmatrix}. \quad (4.26)$$

In the n-frame, the position is expressed using geodetic coordinates of the point, and the velocity is defined by using local directions but in an Earth fixed reference frame. The orientation is described as the transformation matrix from the b-frame to the n-frame. Compared to other

reference frames, expressing the navigation equations in n-frame has a significant advantage that provides the navigation solution with a user-friendly and straightforward output.

4.3. Error characteristics of the inertial sensor

The output data from an accelerometer and a gyroscope contain unknown biases, scale factor and cross-coupling errors, and random noises. Some error sources can be defined as a function of time.

The bias is the dominant term in the IMU device's overall error. The bias is a constant offset of the output value from the inertial sensors defined as the sum of all perturbations resulting from zero inputs. The bias depends on acceleration, temperature changes and magnetic field. Hence, many different bias parameters can be measured, such as in-run bias stability, turn-on bias stability or repeatability, and bias over temperature. Thus, the sensor bias can be different each time, when the sensors are turned on. Normally, the mean and variance are known. The manufacturer of the IMU device gives such information due to the laboratory and field test experiences. The accelerometer and gyroscope biases of an IMU are always represented in the b-frame and these biases are denoted as b_a with (b_{ax}, b_{ay}, b_{az}) and b_g with (b_{gx}, b_{gy}, b_{gz}) . The accelerometer biases are usually provided in milli-gal (mg) or micro-gal (μg). The gyroscope biases are provided in degree per hour ($^\circ/hr^{-1}$ or $^\circ/h$), where $1^\circ/hr^{-1}$ is equal to 4.848×10^{-6} [rad s^{-1}] [Gro13]. When the sensed angular rate data are integrated to determine the transformation matrix, the constant biases of the gyroscope cause an angular rate error which grows linearly with time. The constant biases from the accelerometer cause an error in position and velocity, which grows quadratically with time.

The scale factor error is a ratio of the input to the output of the IMU device for the measurement range. This error uses either a "parts per million error (ppm)", or an error percentage (%) as the unit of measurement. This error depends on temperature, and it can not be perfectly calibrated. There are some remaining errors in the estimated ratio. The accelerometer and gyroscope scale factor errors of the IMU are defined as s_a with (s_{ax}, s_{ay}, s_{az}) and s_g with (s_{gx}, s_{gy}, s_{gz}) . All types of IMU exhibit cross-coupling errors, which includes an orthogonality offset caused by the misalignment of the sensitive axes of the inertial sensor relative to the orthogonal axes of the body frame.

All inertial sensors produce random noise. The sensor noise of the electronics influences the output of the IMU. If an angular rate output signal is integrated to determine an angle, the integration will cause an offset over time due to the noise. The gyroscope exhibits high-frequency

white noise due to thermoelectrical reactions. An accelerometer produces white noise due to mechanical instability. This random noise is denoted as vector w_a with (w_{ax}, w_{ay}, w_{az}) , and for the gyroscope w_g with (w_{gx}, w_{gy}, w_{gz}) . Unfortunately, the random noise can not be calibrated and removed. Sometimes, the white noise on the output of an accelerometer creates a random velocity walk, which is specified in units $\mu g/\sqrt{Hz}$. Thus, the integrated random noise of the specific force measurements produces a random walk error in the inertial velocity solution. The white noise on the output of an gyroscope creates an orientation random walk error, which is specified in unit $^\circ/\sqrt{hr}$ or $^\circ/hr/\sqrt{Hz}$, where $1^\circ/\sqrt{hr}$ is equal to $2.909 \times 10^{-4}rads^{-0.5}$ and $1^\circ/hr/\sqrt{Hz}$ is equal to $4.848 \times 10^{-6}rads^{-0.5}$. This means that integrated random noise on the angular rate measurements produces a random walk error on the orientation solution.

An IMU manufacturer may provide a specification of the sensor errors based on the calibration. Table 2 is an example of specification of an IMU device:

Accelerometer:	Bias repeatability	220 mgal
	scale factor error	300 ppm
	white noise	$40 \text{ mg}/\sqrt{Hz}$
Gyroscope:	Bias repeatability	$^\circ/hr$
	scale factor error	10 ppm
	white noise	$40^\circ/\sqrt{hr}$

Table 2: Example of bias errors, scale-factor errors and noises of an IMU device with an high quality gyroscope and an lower quality accelerometer

The second column in the table represents the type of error, while the third column represents the specific numerical values with their unit of measurement [Jek01].

5. Kalman filter-based parameter estimation

Kalman filtering allows to determine a set of unknown parameters of a system (e.g. the position, velocity and attitude etc.) by means of the given uncertainty information of that system and sensor measurements [Gro13]. Kalman filtering predicts the system state for the next epoch based on past estimation, and can be used in any applications where uncertain information in a dynamic system is available. The sensor uncertainty information includes statistical noise and other inaccuracies, which are observed over time. These information are typically represented as the variance of the normal distribution. They have to be determined once for the sensors by the sensor's manufacture. A perfect sensor has a variance of zero, which means that this sensor provides the measurements extremely accurately. However, such a perfect sensor does not exist. The algorithm uses the variance information to produce a prediction about what this system is going to do next. Hence, the Kalman filter estimation is ideal for implementing in a navigation system that is continuously changing. For instance, the true trajectories of the moving sensor in the dynamic system are usually hidden from the user. The trajectories of the sensor can be estimated from different sensor measurements, such as GNSS or IMU. Kalman filtering can combine the sensor measurements observed simultaneously, and use the given uncertain information to provide the state errors in the measurements made by the system. The measurement errors are the deviations, which represent the differences of the estimated navigation solutions from their true values, such as IMU processing position, velocity, and attitude errors. A true state is the measurement without any deviations obtained by the process of the sensor directly. The goal is to estimate many state variables of the system for many epochs in time with external knowledge about this state. An estimation here means evaluating the hidden error states of the navigation solution, which is key to obtain the optimal navigation solution from the available measurement information of the sensors.

A set of state parameters is estimated by a Kalman filter algorithm, known as the state vector, x_k , where k denotes the epoch [Gro13]. The state vector contains the sensor's uncertain properties. The major parameters in the navigation solution can build a nine component vector, which contains the position, the velocity and the attitude of the object. The remaining random parameters, such as the biases of the IMU are also estimated. State variable estimations are made based on the dynamics of these variables and the measurements of the observable variables obtained at each time point. The Kalman filter estimation is a popular method in navigation applications for estimating the optimal navigation solution of a vehicle by integration of GNSS with the other navigation sensors. In this research, a Kalman filter estimation algorithm is constructed, in which all error states mentioned above would be estimated at once.

5.1. Least squares estimation

A least-squares adjustment (LSQ) is also a well known and often used estimation algorithm in geodesy, which can be used for estimating the unknown parameters of a system based on a set of measurements of that system. Generally, in navigation applications, the estimated parameters play the role of the variables in the state vector like the navigation solution.

In this research, a LSQ algorithm is implemented in the software to estimate the initial parameters (such as e.g. position, velocity and initial ambiguities) for the tightly coupled Integration (see chapters 6.6.2 and 6.6.3). LSQ is an iterative estimation method, and the best-estimated solution is defined as the navigation solution by minimizing the sum of the squares of the measurement errors [Gro13]. The measurement errors are often called measurement residuals, representing the difference between the measured and the estimated measurements. The standard LSQ deals with true and unknown observable parameters, measurements and true and unknown observational errors and often some other unknown parameters. The relationship between unknown parameters and measurements defines the functional model, which is normally represented by a linearized equation [Fot18]. The term 'linear' means that the measurements are linear to the unknown parameters. Hence, l measurements are used to determine the n unknown parameters, and the basic observation equation can be formulated as follows:

$$z = Hx + v, \quad (5.1)$$

where

x = the n -dimensional state vector

z = the l -dimensional measurement vector

v = the measurement and observation noise vector

The number of unknown parameters n has to be smaller than the number of the known measurements l , which is the prerequisite for the functional model [Jek01]. The H matrix is the design matrix, which describes the linearized relation between unknown parameters and measurements. It is generally considered for the vector v :

$$v \sim N(0, R), \quad (5.2)$$

where

R = the covariance matrix of the measurement error with $l \times l$ dimension

The noise vector v bases on a Gaussian normal distribution with zero mean and random

process with covariance matrix R . The best estimate \hat{x} of the unknown state vector could be simply determined by minimizing the sum of squared residuals, and this minimization problem can be expressed by [He15]:

$$v^T R^{-1} v = (z - H\hat{x})^T R^{-1} (z - H\hat{x}) = \min \quad (5.3)$$

where R^{-1} is used as the suitable weight matrix for the observation. The equation above can be simplified, which leads to the normal equation as the LSQ method:

$$\hat{x} = (H^T R^{-1} H)^{-1} H^T R^{-1} z. \quad (5.4)$$

In case of GNSS positioning, the observation vector can be linearized as follows:

$$P_{observed} = P_{x,y,z,\delta_r} + v. \quad (5.5)$$

The given navigation message allows us to compute the three-dimensional satellite position and the satellite clock bias δ^s , and the unknown parameters contain the three-dimensional receiver position and the receiver clock bias δ_r . The residual observation ΔP is defined to be the difference between the actual-epoch measured observation and the calculated observation by using the initial receiver clock error and the approximate position, $P(x_0, y_0, z_0, \delta_{r0})$ [Ble97]:

$$\begin{aligned} \Delta P &= P_{observed} - P_{computed}, \\ &= \frac{\delta P}{\delta x} \Delta x + \frac{\delta P}{\delta y} \Delta y + \frac{\delta P}{\delta z} \Delta z + \frac{\delta P}{\delta \delta_r} \Delta \delta_r + v. \end{aligned} \quad (5.6)$$

For m satellites in the obtained observation, the equation above can be written in a matrix form:

$$\begin{pmatrix} \Delta P^1 \\ \Delta P^2 \\ \vdots \\ \Delta P^m \end{pmatrix} = \begin{pmatrix} \frac{\delta P^1}{\delta x} & \frac{\delta P^1}{\delta y} & \frac{\delta P^1}{\delta z} & \frac{\delta P^1}{\delta \delta_r} \\ \frac{\delta P^2}{\delta x} & \frac{\delta P^2}{\delta y} & \frac{\delta P^2}{\delta z} & \frac{\delta P^2}{\delta \delta_r} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\delta P^m}{\delta x} & \frac{\delta P^m}{\delta y} & \frac{\delta P^m}{\delta z} & \frac{\delta P^m}{\delta \delta_r} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta \delta_r \end{pmatrix} + \begin{pmatrix} v^1 \\ v^2 \\ \vdots \\ v^m \end{pmatrix}. \quad (5.7)$$

The H matrix are the partial derivatives of each satellite observation relative to other state variables. The components of H matrix can be derived by the partial differentiation of the

observation equation.

$$H = \begin{pmatrix} \frac{x_0-x^1}{\rho} & \frac{y_0-y^1}{\rho} & \frac{z_0-z^1}{\rho} & C \\ \frac{x_0-x^2}{\rho} & \frac{y_0-y^2}{\rho} & \frac{z_0-z^2}{\rho} & C \\ \frac{x_0-x^3}{\rho} & \frac{y_0-y^3}{\rho} & \frac{z_0-z^3}{\rho} & C \\ \vdots & \vdots & \vdots & \vdots \\ \frac{x_0-x^m}{\rho} & \frac{y_0-y^m}{\rho} & \frac{z_0-z^m}{\rho} & C \end{pmatrix} \quad (5.8)$$

where

c = the speed of light

In general, for m visible satellites, the H matrix contains the line of sight vector between the sensor and the visible satellites. ρ represents the approximate distance from the satellite at transmit time to the receiver at the receive time, which can be calculated by using equation 3.4.

Nowadays, the GNSS receivers fix their positions using Kalman filtering or LSQ estimation algorithm, whereby the Kalman filter is more commonly used in geodesy. Primarily, Kalman filtering works for navigation applications of GNSS and IMU. The Kalman filter estimation bases on a time update (prediction step) from the previous epoch, which predicts the state vector x and its covariance matrix for the next epoch. It provides the state information which changes with time. Kalman filtering minimizes the mean-square error of the solution. On the other hand, LSQ bases on the minimization of the measurement residuals [Pet15].

5.2. Core element of Kalman filter

Kalman filtering can be described by its five core elements: the state vector and its covariance, the system model, the measurement vector and its covariance, the measurement model, and the algorithm [Gro13].

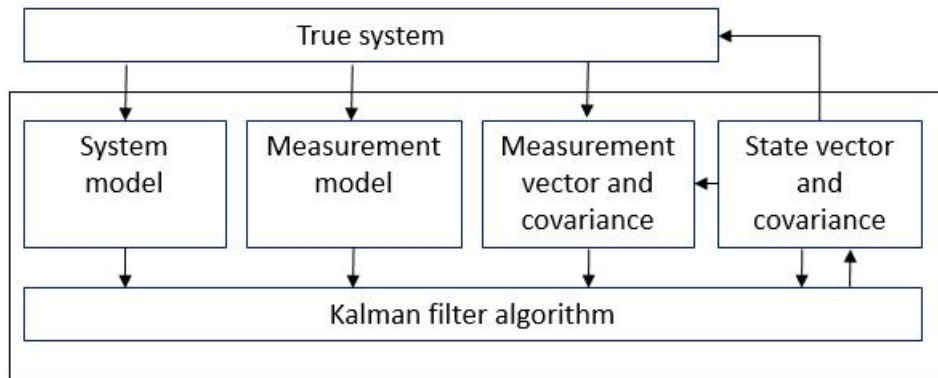


Figure 9: Elements of Kalman filtering [Gro13]

The state vector is the set of parameters to be estimated. An error covariance matrix is associated with the state vector. The covariance matrix describes the uncertainties of the state estimation during the processing and the degree of correlation between the errors during estimation processing. A Kalman filter estimation bases on recursive processing, so the initial values of the state vector and covariance matrix must be defined or determined at first.

The system model comprises of the known properties of the dynamic system. It describes the behavior of states and the error covariance matrix over time. The change of the state in position over time can be characterized by the integral of velocity. The uncertainty will increase with time as the integral of the velocity uncertainty. Hence, the position and velocity estimates are definitely correlated.

The measurement model describes the behavior of the measurement vector as a function of the true state vector without any measurement noise. The measurement vector is a set of simultaneous measurements of properties of the system, and this can be defined as functions of the state vector. The measurement vector, for example, can be the range measurements from GNSS. In the integrated system, the vector can be understood as the difference of position and velocity vector between the IMU and GNSS solution or the difference between the IMU derived range measurements and the GNSS range measurements. There is a measurement noise variance matrix that is associated with the measurement vector. This matrix describes the noise of the measurements.

The Kalman filter algorithm based on these five core elements can provide the user with the most optimal state vector estimation.

5.3. Kalman filter algorithm

The Kalman filtering algorithm comprises three main steps: filtering, prediction and update [Jek01]. Filtering is an essential step of the whole processing. In this step the state parameters are determined in a stochastic optimum way. Observations with different statistical characterizations from various sources are combined. Prediction is the propagation of the states and error covariance matrix from last epoch to the current epoch based on the system dynamic model and the a priori probabilities [Gro13]. Update is a generalization of the step filtering. This step is used to estimate the states at any time based on all of the available observations before or after that time [Jek01]. The algorithm of filtering bases on the recursive processing. The estimated states from the previous epoch and all information about system states at the current epoch are required to compute the estimated states, which are used to predict the new states of the system. Hence, the recursive estimation solution at any time bases on all observations available up to that time.

The system state vector x_k contains the terms of interest for the integrated system within the Kalman filter algorithm up to any time t_k . In case of GNSS/IMU combined data processing, the state vector of the system constitutes the set of parameters in position, velocity, attitude of the body and various parameters describing systematic errors of the IMU device. As mentioned before, the state errors represent the deviation the differences of the estimated measurements from their true values, which are the measured data by process of the sensors directly. A GNSS observation adds information to the system from a source external to the IMU device. The time epochs can be defined as follows:

$$t_k - t_{k-1} = \delta t, \quad k = 0, 1, \dots \quad (5.9)$$

The estimated state vector at the time, t_k , is defined as \hat{x}_k . The state vector residual, δx , can be defined as the difference between the true state vector and the state vector estimated by Kalman filtering [Gro13].

$$\delta x = x - \hat{x}_k, \quad (5.10)$$

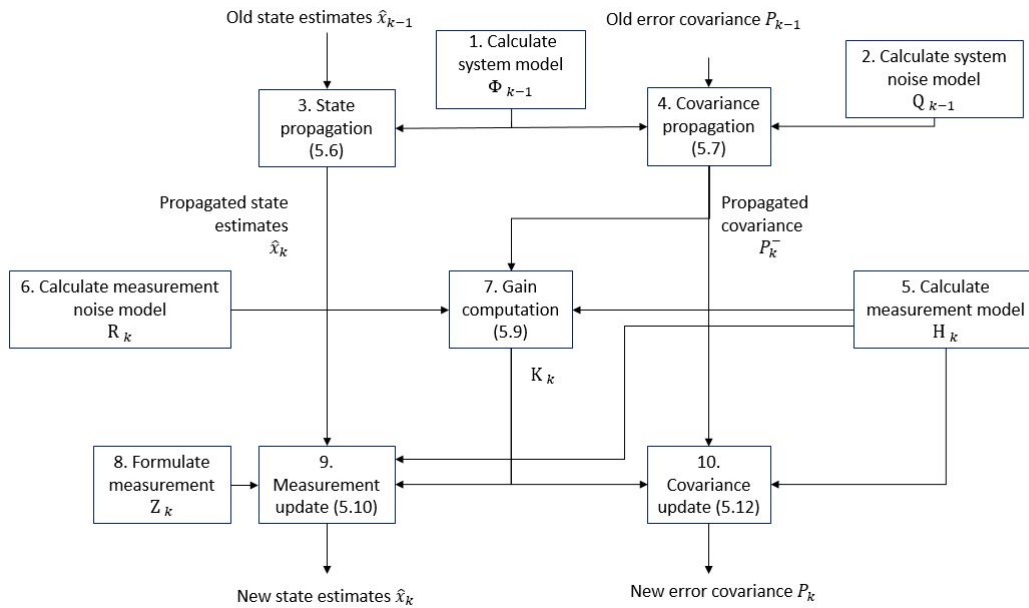


Figure 10: Kalman filter algorithm steps. [Gro13]

The Kalman filter algorithm can be described by the sequence of below noted steps :

1. Calculate the transition matrix, Φ_{k-1} .
2. Calculate the system noise covariance matrix, Q_{k-1} .
3. Propagate the state vector estimate from \hat{x}_{k-1} to \hat{x}_k^- .
4. Propagate the error covariance matrix from P_{k-1} to P_k^- .
5. Calculate the measurement design matrix, H_k
6. Calculate the measurement noise covariance matrix, R_k
7. Calculate the Kalman Gain matrix, K_k
8. Create the measurement vector, z
9. Update the state vector estimate from \hat{x}_k^- to \hat{x}_k .
10. Update the error covariance matrix from P_k^- to P_k .

The first four steps are used for filtering and prediction. The prediction can be defined as the estimation of the state vector based on the system dynamics from t_{k-1} to t_k . Step 1 represents how the state vector changes with time as a function of the system's dynamics. For example, the variability of the state variable in position can be described by the integral of a velocity state [Gro13]. The state transition matrix Φ is always derived from the dynamic system model for the time interval (δt) epoch by epoch. The Kalman filter is an iterative estimation processing, which means that the transition matrix has to be recalculated on every iteration.

$F(t_k, t_{k-1})$ is the system dynamics matrix. This matrix represents the Kalman filter

system model and describes how the states and their uncertainties are propagated with time [Gro13]. The system matrix is derived from the known properties. This research focuses on a GNSS/IMU integrated system, and $F(t_k, t_{k-1})$ can be defined by the equation 6.43. A prediction equation of the state vector x is used to describe the dynamic system model as:

$$\dot{x}(k) = F(t_k, t_{k-1})x_{k-1}, \quad (5.11)$$

The estimation of a state vector bases on the modelling as a linear function of its value from last epoch, which is coupled by the transition matrix Φ_{k-1} . The hat operator " \hat{x} " stands for an estimate of the state vector.

$$\hat{x}_k = \hat{x}_{k-1} e^{F_{k,k-1} \delta t}, \quad (5.12)$$

where $\dot{x}(k)$ is the linear function of x_k . The equations 5.11 and 5.13 create the state representation of the dynamic system, which has been solved, in order to define the transition matrix.

$$\hat{x}_k = \Phi_{k-1} \hat{x}_{k-1}. \quad (5.13)$$

The relationship between F_{k-1} matrix and Φ_{k-1} can be formulated as follows:

$$\Phi_{k-1} = \exp(F_{k-1} \delta t). \quad (5.14)$$

Step 2 represents how the uncertainty of the estimated state vector increases over time. There are unknown changes in the true values of state variables, such as instrument noise and navigation solution deviations. These changes increase uncertainties of state estimates with time. The calculation of the system noise covariance matrix defines how the uncertainties increase with time. The system noise is always a function of the time interval (δt) between iterations. Q_{k-1} is the symmetric system noise covariance matrix at time t_{k-1} . The system noise covariance matrix Q is usually approximated [Gro13], and can be obtained by:

$$Q_{k-1} = G_{k-1} Q_{t_{k-1}} G_{k-1}^T \delta t. \quad (5.15)$$

where

$Q_{t_{k-1}}$ = the corresponding noise coefficient matrix

G = the system noise distribution matrix

The G matrix is constant over the time interval (δt). This matrix is derived from the known properties of the system (see equation 6.45). In general, the noise coefficient matrix, Q_t , is a diagonal matrix comprising the components of the process noise vector, w_k . The expectation value of the noise vector is zero. Here, w_k provides a measurable input to the system, this vector bases on a set of random noise sources, each of them is assumed to be of zero-mean [Gro13].

$$w_k \approx N(0, Q_k), \quad (5.16)$$

In the GNSS/IMU integration, the noise on the navigation solution calculated by the IMU data are random walk due to noise on the acceleration measurements and the angular-rate measurements (see chapter 6.5).

The Q_{k-1} can also be calculated using the first order approximation of the transition matrix Φ_{k-1} [Shi01],

$$Q_{k-1} = \Phi_{k-1}(G_{k-1}Q_{k-1}G_{k-1}^T)\Phi_{k-1}^T \delta t. \quad (5.17)$$

Step 3 defines the propagation of the state vector with respect to time, which is carried out by equation 5.13. Step 4 is used to describe the error covariance propagation, which is formulated by:

$$P_k = \Phi_{k-1}P_{k-1}\Phi_{k-1}^T + Q_{k-1}. \quad (5.18)$$

The error covariance matrix is denoted as P matrix, which can be regarded as the expectation of the square of the deviation of the estimated state vector from the true value of the state vector. P is a symmetric matrix. The diagonal elements of the matrix are the variance of each estimated state,

$$P_{ii} = \sigma_i^2, \quad (5.19)$$

where

σ_i^2 = the uncertainty of the *i*th estimate state

In statistics, the variance σ_i^2 is defined as the measurements of variability between numbers in a data set from the mean values.

$$\sigma_i^2 = \frac{\sum_{i=1}^n (x_i - x_m)^2}{n - 1} \quad (5.20)$$

where

x_i = *i*th data value

x_m = the mean value

n = the number of data values

The standard deviation σ can be calculated from the square root of the variance.

The off-diagonal elements of the P matrix describe the variance of each state estimate while their square roots are the uncertainties [Gro13].

$$P_{ij} = P_{ji} = \sigma_i \sigma_j \rho_{i,j}. \quad (5.21)$$

where

$\rho_{i,j}$ = the correlation coefficient

The remaining steps in the Kalman filter algorithm describe the update process. The estimation of the state vector using the measurements is defined as update. Step 5 defines the H matrix, also called as the design matrix, representing how the measurement vector depends on the state vector parameters. Each row of the H matrix describes the measurements, and each column describes a state variable. H matrix varies with the values of the state vector, and the matrix has to be recalculated on each iteration step. If the state vector comprises the position errors of the IMU measurements, and the measurement vector is built as the difference between the IMU solution and the GNSS solution, the H matrix is simply the identity matrix.

Step 6 represents the calculation of the R matrix. R_k matrix is the measurement noise covariance matrix which is a symmetric matrix. It describes the expectation of the square of the measurement noise. The diagonal elements of the R matrix contain the variance of each measurement, and the off-diagonal elements describe the correlation between the different measurement combinations. The matrix components of R are not constant over time. In the integrated system, the system can be modeled as a function of dynamics. The R matrix can also be modeled as a function of signal-to-noise measurement of GNSS satellites.

Step 7 is used to determine the weight matrix, which represents the weighting between the measurements and the estimated state vector. Each row describes a state variable and each column describes a measurement. The Kalman gain matrix can be formulated as follows:

$$K_k = P_k H_k^T (H_k P_k H_k^T + R_k)^{-1}. \quad (5.22)$$

where

R_k = the error covariance of the true measurement vector

$H_k P_k H_k^T$ = the error covariance of the measurement vector predicted from state vector

The Kalman gain depends on the error covariance matrix of the true measurement vector, and the error covariance matrix of a predicted measurement, which is calculated from the state vector, $H_k \hat{x}_k$.

Step 8 relates to the measurement vector. Here, z is a set of measurements of the properties of the integrated system, which is described by the state vector. The choice of vector components depends on the approach of the Kalman filter algorithm. The components of the measurement vector may comprise GNSS range measurements, or it may comprise the difference between the IMU solution and the GNSS solution. For example, this can be difference between the raw range measurements of the GNSS and the range derived from the IMU standalone processing or the difference of a positioning and velocity solution between the GNSS and the IMU.

Step 9 describes the update of the state vector by means of the measurement vector. The update process starts from the following observation equation:

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H_k \hat{x}_k^-). \quad (5.23)$$

The measurement innovation δz_k^- describes the difference between the true measurement vector and its expected value before the measurement update [Jek01]. This can be formulated as follows:

$$\delta z_k^- = (z_k - H_k \hat{x}_k^-). \quad (5.24)$$

In this research, the measurement vector comprises from observations of the different sensors, and H and z together make up the measurement model based on the observation differences between the sensors.

Step 10 describes the update of the error covariance matrix with

$$P_k = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T. \quad (5.25)$$

The Kalman gain represents how much innovation is appropriate to be added to the estimates of the last epoch. If the observation is not accurate enough, R_k is large compared to the P matrix. The matrix makes the Kalman gain smaller, and only little innovation is added. The observation with lower accuracy will have little influence on the estimation. On the other hand, if the observation is accurate, R_k has a small value and the observation will have more

influence on the estimation for the next epoch.

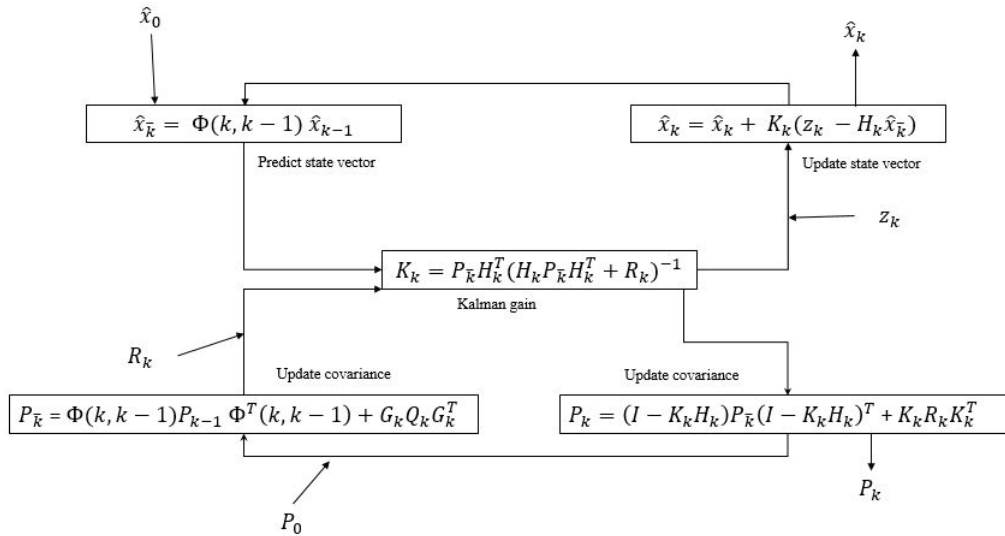


Figure 11: Kalman filter loop [Jek01].

Figure 11 represents a simple description of the prediction and update steps within a Kalman filter process. Table 3 gives an overview about the important matrices and vectors in the Kalman filter.

Parameter	Name	Dimension
x	State vector	$n_x \times 1$
F / Φ	Transitionsmatrix	$n_x \times n_x$
G	System noise distribution matrix	$n_x \times n_{w_k}$
w_k	Control vector	$n_{w_k} \times 1$
z	Measurement vector	$n_z \times 1$
H	Measurement matrix	$n_z \times n_x$
v	Measurement noise vector	$n_z \times 1$
K	Kalman gain	$n_z \times n_x$
P	Process error covariance matrix	$n_x \times n_x$
Q	System noise covariance matrix	$n_x \times n_x$
R	Measurement noise covariance matrix	$n_z \times n_z$

Table 3: Matrices and vectors in a Kalman filter

6. GNSS/IMU integration algorithm

It is well known that IMU-data standalone processing does not offer a long-period stability. Therefore, IMU data has to be combined with another external sensor data, such as GNSS or odometer, to obtain high-quality position-related information. This section is a general description how to set up an appropriate Kalman filter algorithm to determine the navigation solution by an integrated system.

As mentioned in chapter 5, the goal of the GNSS/IMU integration is firstly to estimate the errors of the currently available navigation solution. The term errors relate in this thesis to the estimated parameter updates to the predicted solution. The estimated position and velocity errors may be expressed in a n -frame as the geodetic coordinates δr^n and δv^n . In general, the position vector r^n covers the φ , λ , h components, and the velocity vector v^n the v_N , v_E , v_D components. Using sensor integration is also a significant benefit in estimating the attitude errors of a moving body. The attitude of the body can be expressed as small angles resolved about the IMU sensor's b -frame axes, denoted as $\delta\psi_{nb}^n$. In this research, the algorithm is developed that the sensor's attitude errors are converted to a quaternion and can be represented as a rotation vector based on the Euler angles. In addition, the algorithm also estimates the accelerometer biases; denoted as δf_{ib}^b . Certainly, the gyroscope biases, δw_{ib}^b , are also components of the state vector. These biases become significant, as there is a quadratic growth of error in velocity with time. In general, the state vector is comprised of the classical 15 state components concept. Sometimes, the Kalman filter algorithm may include other further six-state components, which are the acceleration and gyroscope scale factors. To take these parameters into account may improve the accuracy of estimated acceleration and gyroscopes biases. Here, the state vector consists of the following 15 components:

$$\delta x^n = \left(\delta r^n \quad \delta v^n \quad \delta\psi^n \quad \delta w_{ib}^b \quad \delta f_{ib}^b \right). \quad (6.1)$$

Two different Kalman filter concepts will be applied. On the one hand, within the loosely coupled integration, the positioning solutions derived from the GNSS signal processing and the velocity derived from the Odometer measurements are used together with the position and velocity derived from the IMU standalone measurements. However, when there is an outage of the GNSS signals, the algorithm calculates the parameters based just on the IMU/odometer combined observations. On the other hand, within the tightly coupled integration architecture, raw GNSS pseudo-range, carrier-phase and doppler-shift observations are introduced. Tracking at least two visible satellites is required for the processing. The GNSS observation data from

a master station and a rover station with dual-frequency must be available to apply the DD approach. The DD approach allows to eliminate the satellite and receiver clock errors. The ionosphere delay is also eliminated by using the ionosphere-free linear combination.

6.1. Lever-arm vector

Traditionally, the IMU device is installed separately from the other sensors (e.g. GNSS rover and odometer) at the vehicle. They work independently of each other and simultaneously. A navigation solution is often transposed from the IMU's position to another sensors or vice versa. For instance, the distance from the GNSS rover phase center relative to the IMU device center can be defined as the "lever-arms" effect [Shi05].

In real life, the lever arm is difficult to be measured in the coordinate frame of the IMU. The lever arm can be represented in form of a Cartesian a three-parameter position vector l^b in the body-fixed frame [Gro13]. This vector can manually be measured by surveying with a total station with high accuracy of 1 mm level [Fu+18]. Inaccurate measurements of the lever-arm components lead to a deterioration of the determined navigation solution.

The Cartesian position of the GNSS antenna may be described as:

$$r_{GNSS}^n = r_{IMU}^n + D^{-1}C_b^n l^b \quad (6.2)$$

where

r_{GNSS}^n = the position of the GNSS antenna centre

r_{IMU}^n = the position of the IMU centre in the n-frame

l^b = the lever arms vector of the GNSS antenna relative to IMU in the b-frame

C_b^n = the coordinate transformation matrix from the b-frame to the n-frame

D = The conversion from the Cartesian to the geographic coordinates

r_{IMU}^n usually represents the position of the IMU reference center. In case, both the position of the IMU reference center and the GNSS antenna shall be expressed in geographic coordinates. The Cartesian lever arm vector has to be converted by means of matrix D .

$$D^{-1} = \begin{pmatrix} \frac{1}{M+h} & 0 & 0 \\ 0 & \frac{1}{(N+h)\cos\varphi} & 0 \\ 0 & 0 & -1 \end{pmatrix}, \quad (6.3)$$

N represents the radius of curvature of the ellipsoid in the vertical direction, and M is

the meridian radius of curvature. They are defined in equations 4.14 and 4.15. The influence of the lever arm vector on the velocity measurements is formulated as follows [Gro13], where w_{ib}^b denotes the angular rate measurements of the IMU device.

$$v_{GNSS}^n = v_{IMU}^n + C_b^n (w_{ib}^b \times l^b) \quad (6.4)$$

6.2. Odometer

The odometer is another suitable sensor, which monitors the wheel rotations of the vehicle. This sensor can be extremely supportive in addition to the GNSS and the IMU devices. While the train moves inside the tunnel or under a bridge. The odometer usually obtains the train wheel velocity and the distance information. The sensor counts the number of pulses, which is generated by a rotating wheel on a vehicle. The obtained pulses number can be converted to the corresponding distance [m] and corresponding velocity [m/s]. The absolute velocity can be obtained by using the following equation [Che16]:

$$v_{ODO}^b = \frac{N_{ODO}}{Q_{ODO}} * \pi * d, \quad (6.5)$$

where

v_{ODO}^b = the velocity of the odometer in b-frame

N_{ODO} = the number of pulses per second counted by the odometer

Q_{ODO} = the number of pulses after a full wheel rotation

d = the diameter of the train wheel

The velocity measured by the odometer should be transformed into the n-frame, in order to simplify the fusion of all sensor data. The transformation of the velocity from b-frame to n-frame can be carried out by using the following equation:

$$v_{ODO}^n = C_b^n * \begin{pmatrix} v_{ODO}^b \\ 0 \\ 0 \end{pmatrix}. \quad (6.6)$$

Physically, the odometer can not be located close to the GNSS and the IMU sensors. The difference between the velocity measured by the odometer and the one derived by the IMU is defined as a part of the measurement vector. However, the odometer is usually fixed at the wheel of a vehicle, thus the lever arm vector between both sensors should be considered. At the

same epoch, the velocity tracked by the IMU and the one of an odometer show a little difference due to the lever-arm vector [Lee06]. Similar to equation 6.4, the lever-arms vector between the IMU device and an odometer can be formulated as follows:

$$v_{ODO}^n = v_{IMU}^n + C_b^m (w_{ib}^b \times l^b) \quad (6.7)$$

6.3. IMU State propagation resolved in a Local Navigation Frame

This subsection provides guidance on the IMU error propagation of the estimated states within the Kalman filter. Kalman filtering with a system model (see Equations 5.11 and 5.13) and a measurement update (see Equations 5.23 and 5.25) are described. To obtain the system model, the time derivative of each error state parameter (see Equation 6.1) has to be calculated by using the error propagation equations [Gro13]. The system error propagation of the IMU can be described by differential equations, which mainly represent the error state propagation equations of position, velocity and attitude. In this research, the final estimated position solution is computed in latitude, longitude and height. The velocity vector is Earth referenced but resolved in the NED axes. The estimated attitude refers to the Earth, and is represented by the pitch, roll and yaw angles.

In general, the true solution of the position, velocity and attitude of the sensor are unknown, thus the best estimates are usually the corrected IMU solution. The IMU error state (symbol „ δ ”) shows the difference between the solution derived by using the IMU measurements with symbol „ $\tilde{\cdot}$ ” and the corrected solution with symbol „ $\hat{\cdot}$ ”. The relationship can be described as follows:

$$\begin{aligned} \hat{r}_{eb}^n &= \tilde{r}_{eb}^n - \delta r_{eb}^n \\ \hat{v}_{eb}^n &= \tilde{v}_{eb}^n - \delta v_{eb}^n, \\ \hat{\psi}_{eb}^n &= \tilde{\psi}_{eb}^n - \delta \psi_{eb}^n \end{aligned} \quad (6.8)$$

In terms of the attitude the IMU state error can be expressed in form of the coordinate transformation matrix from the b-frame to the n-frame, C_b^m :

$$\hat{C}_b^m = (I - E^n) \tilde{C}_b^m, \quad (6.9)$$

Here, the skew symmetric matrix, E^n , with the three small angle errors resolved in the NED axes, is used to describe the attitude error:

$$E^n = \epsilon^n \times = \begin{pmatrix} 0 & -\epsilon_D & \epsilon_E \\ \epsilon_D & 0 & -\epsilon_N \\ -\epsilon_E & \epsilon_N & 0 \end{pmatrix}, \quad (6.10)$$

The position error propagation depends on the position and velocity of the sensor. The partial derivatives are used to compose the equation into a position and a velocity error term [Chi04].

$$\delta \dot{r}_{eb}^n \begin{pmatrix} \delta \dot{\varphi} \\ \delta \dot{\lambda} \\ \delta \dot{h} \end{pmatrix} = F_{rr} \begin{pmatrix} \delta \varphi \\ \delta \lambda \\ \delta h \end{pmatrix} + F_{rv} \begin{pmatrix} \delta v_N \\ \delta v_E \\ \delta v_D \end{pmatrix}, \quad (6.11)$$

The position error equation in geodetic coordinates in the n-frame can be obtained by deriving the equation 4.12a. The time derivative of the position error reads [Gro13]:

$$\delta \dot{r}_{eb}^n \begin{pmatrix} \delta \dot{\varphi} \\ \delta \dot{\lambda} \\ \delta \dot{h} \end{pmatrix} = \begin{pmatrix} \frac{\delta v_N}{M + \hat{h}} - \frac{\hat{v}_N \delta h}{(M + \hat{h})^2} \\ \frac{\delta v_E}{(N + \hat{h}) \cos \hat{\varphi}} + \frac{\hat{v}_E \sin \hat{\varphi} \delta \varphi}{(N + \hat{h}) \cos^2 \hat{\varphi}} - \frac{\hat{v}_E \delta h}{(N + \hat{h})^2 \cos \hat{\varphi}} \\ -\hat{v}_D \end{pmatrix}. \quad (6.12)$$

and

$$F_{rr} = \begin{pmatrix} \frac{\delta \dot{\varphi}}{\delta \varphi} & \frac{\delta \dot{\varphi}}{\delta \lambda} & \frac{\delta \dot{\varphi}}{\delta h} \\ \frac{\delta \dot{\lambda}}{\delta \varphi} & \frac{\delta \dot{\lambda}}{\delta \lambda} & \frac{\delta \dot{\lambda}}{\delta h} \\ \frac{\delta \dot{h}}{\delta \varphi} & \frac{\delta \dot{h}}{\delta \lambda} & \frac{\delta \dot{h}}{\delta h} \end{pmatrix} = \begin{pmatrix} 0 & 0 & \frac{-\hat{v}_N}{(M + \hat{h})^2} \\ \frac{\hat{v}_E \sin \hat{\varphi}}{(N + \hat{h}) \cos^2 \hat{\varphi}} & 0 & \frac{-\hat{v}_E}{(N + \hat{h})^2 \cos \hat{\varphi}} \\ 0 & 0 & 0 \end{pmatrix}, \quad (6.13)$$

$$F_{rv} = \begin{pmatrix} \frac{\delta\dot{\varphi}}{\delta v_N} & \frac{\delta\dot{\varphi}}{\delta v_E} & \frac{\delta\dot{\varphi}}{\delta v_D} \\ \frac{\delta\dot{\lambda}}{\delta v_N} & \frac{\delta\dot{\lambda}}{\delta v_E} & \frac{\delta\dot{\lambda}}{\delta v_D} \\ \frac{\delta\dot{h}}{\delta v_N} & \frac{\delta\dot{h}}{\delta v_E} & \frac{\delta\dot{h}}{\delta v_D} \end{pmatrix} = \begin{pmatrix} \frac{1}{(M + \hat{h})} & 0 & 0 \\ 0 & \frac{1}{(N + \hat{h})\cos\hat{\varphi}} & 0 \\ 0 & 0 & -1 \end{pmatrix}, \quad (6.14)$$

The time derivative of the velocity error is derived from the rate of change of the velocity calculated from the equation 4.12b [Shi01]:

$$\dot{v}_{eb}^n = \hat{C}_b^n \tilde{f}_{ib}^b - (\hat{\omega}_{en}^n + 2\hat{\omega}_{ie}^n) \times \hat{v}_{eb}^n + g_b^n. \quad (6.15)$$

and

$$\delta\dot{v}_{eb}^n + \dot{v}_{eb}^n = (I - E^n) \hat{C}_b^n (\tilde{f}_{ib}^b + \delta f_{ib}^b) - (\hat{\omega}_{en}^n + 2\hat{\omega}_{ie}^n + \delta\omega_{en}^n + 2\delta\omega_{ie}^n) \times (\hat{v}_{eb}^n + \delta v_{eb}^n) + \tilde{g}_b^n + \delta g_b^n, \quad (6.16)$$

Thus the velocity error can be noted as follows:

$$\begin{aligned} \delta\dot{v}_{eb}^n &= -(\delta\omega_{en}^n + 2\delta\omega_{ie}^n) \times \hat{v}_{eb}^n + \delta g^n - (\hat{\omega}_{en}^n + 2\hat{\omega}_{ie}^n) \times \delta v_{eb}^n - \epsilon^n \times f_{ib}^n + \delta f_{ib}^n + \delta g_b^n, \\ &= (\delta\omega_{en}^n + 2\delta\omega_{ie}^n) \times \hat{v}_{eb}^n - (\hat{\omega}_{en}^n + 2\hat{\omega}_{ie}^n) \times \delta v_{eb}^n + f_{ib}^n \times \epsilon^n + \delta f_{ib}^n + \delta g_b^n \end{aligned}, \quad (6.17)$$

According to the equation 4.26 the $\omega_{en}^n + 2\omega_{ie}^n$ can be obtained from:

$$w_{en}^n + 2w_{ie}^n = \begin{pmatrix} 2w_e \cos\hat{\varphi} + \frac{\hat{v}_E}{(N + \hat{h})} \\ \frac{-\hat{v}_N}{(N + \hat{h})} \\ -2w_e \sin\hat{\varphi} - \frac{\hat{v}_E \tan\hat{\varphi}}{N + \hat{h}} \end{pmatrix}. \quad (6.18)$$

The first and second error terms on the right hand side can be defined as functions of position and velocity errors. Thus, the error term of the vectors can be given by

$$\delta w_{en}^n + 2\delta w_{ie}^n = \delta\Omega_r \delta r^n + \delta\Omega_v \delta v^n \quad (6.19)$$

with

$$\delta\Omega_r = \begin{pmatrix} -2w_e \sin\hat{\varphi} & 0 & \frac{-\hat{v}_E}{(N+\hat{h})} \\ 0 & 0 & \frac{\hat{v}_N}{(N+\hat{h})} \\ -2w_e \cos\hat{\varphi} - \frac{\hat{v}_E}{(N+\hat{h})\cos^2\hat{\varphi}} & 0 & \frac{\hat{v}_E \tan\hat{\varphi}}{(N+\hat{h})^2} \end{pmatrix}, \quad (6.20)$$

and

$$\delta\Omega_v = \begin{pmatrix} 0 & \frac{1}{(N+\hat{h})} & 0 \\ \frac{-1}{(M+\hat{h})} & 0 & 0 \\ 0 & \frac{-\tan\hat{\varphi}}{(N+\hat{h})} & 0 \end{pmatrix}. \quad (6.21)$$

Now the equations 6.17 and 6.19 can be used to obtain the functions of the velocity errors [Shi01]:

$$(v_{eb}^n \times)(2\delta w_{ie}^n + \delta w_{en}^n) = (v_{eb}^n \times)\delta\Omega_r \delta r^n + (v_{eb}^n \times)\delta\Omega_v \delta v^n. \quad (6.22)$$

The terms at the right hand side of the equation 6.22 can be noted as:

$$(v_{eb}^n \times)\delta\Omega_r = \begin{pmatrix} -2\hat{v}_E w_e \cos\hat{\varphi} - \frac{\hat{v}_E^2}{(N+\hat{h})\cos^2\hat{\varphi}} & 0 & \frac{-\hat{v}_E \hat{v}_D}{(M+\hat{h})^2} + \frac{\hat{v}_E^2 \tan\hat{\varphi}}{(N+\hat{h})^2} \\ -2w_e(\hat{v}_N \cos\hat{\varphi} - \hat{v}_D \sin\hat{\varphi}) + \frac{\hat{v}_E \hat{v}_D}{(N+\hat{h})\cos^2\hat{\varphi}} & 0 & \frac{-\hat{v}_E \hat{v}_D}{(N+\hat{h})^2} - \frac{\hat{v}_N \hat{v}_E \tan\hat{\varphi}}{(N+\hat{h})^2} \\ 2\hat{v}_E w_e \sin\hat{\varphi} & 0 & \frac{\hat{v}_E^2}{(N+\hat{h})^2} + \frac{\hat{v}_N^2}{(M+\hat{h})^2} \end{pmatrix}, \quad (6.23)$$

and

$$(v_{eb}^n \times)\delta\Omega_v = \begin{pmatrix} \frac{\hat{v}_D}{M+\hat{h}} & \frac{-\hat{v}_E \tan\hat{\varphi}}{N+\hat{h}} & 0 \\ 0 & \frac{\hat{v}_D}{N+\hat{h}} + \frac{\hat{v}_N \tan\hat{\varphi}}{N+\hat{h}} & 0 \\ \frac{-\hat{v}_N}{M+\hat{h}} & \frac{-\hat{v}_E}{N+\hat{h}} & 0 \end{pmatrix}. \quad (6.24)$$

The vector g^n is calculated with good approximation by the normal gravity vector $(0, 0, \gamma)^T$. The vector γ changes with the height of the point. This vector can be formulated as follows:

$$\gamma = \gamma_0 \left(\frac{R}{R + \hat{h}} \right)^2, \quad (6.25)$$

where γ_0 is the normal gravity vector at zero elliptical height with $R = \sqrt{MN}$ and

$$\delta\gamma = -2 \left(\frac{\gamma}{R + \hat{h}} \right) \delta h. \quad (6.26)$$

Therefore, the velocity error propagation equation 6.17 can be formulated as follows:

$$\delta\dot{v}^n = F_{vr} \delta r^n + F_{vv} \delta v^n + (f_{ib}^n \times) \epsilon^n + \delta f_{ib}^n + \delta g^n \quad (6.27)$$

where

$$F_{vr} = \begin{pmatrix} -2\hat{v}_E w_e \cos\hat{\varphi} - \frac{\hat{v}_E^2}{(N + \hat{h}) \cos^2\hat{\varphi}} & 0 & \frac{-\hat{v}_E \hat{v}_D}{(M + \hat{h})^2} + \frac{\hat{v}_E^2 \tan\hat{\varphi}}{(N + \hat{h})^2} \\ -2w_e (\hat{v}_N \cos\hat{\varphi} - \hat{v}_D \sin\hat{\varphi}) + \frac{\hat{v}_E \hat{v}_D}{(N + \hat{h}) \cos^2\hat{\varphi}} & 0 & \frac{-\hat{v}_E \hat{v}_D}{(N + \hat{h})^2} - \frac{\hat{v}_N \hat{v}_E \tan\hat{\varphi}}{(N + \hat{h})^2} \\ 2\hat{v}_E w_e \sin\hat{\varphi} & 0 & \frac{\hat{v}_E^2}{(N + \hat{h})^2} + \frac{\hat{v}_N^2}{(M + \hat{h})^2} - \frac{2\gamma}{R + \hat{h}} \end{pmatrix}, \quad (6.28)$$

and

$$F_{vv} = \begin{pmatrix} \frac{\hat{v}_D}{M + \hat{h}} & -2w_e \sin\hat{\varphi} - 2 \frac{\hat{v}_E \tan\hat{\varphi}}{N + \hat{h}} & \frac{\hat{v}_N}{M + \hat{h}} \\ 2w_e \sin\hat{\varphi} + \frac{\hat{v}_E \tan\hat{\varphi}}{N + \hat{h}} & \frac{\hat{v}_D + \hat{v}_N \tan\hat{\varphi}}{N + \hat{h}} & 2w_e \cos\hat{\varphi} + \frac{\hat{v}_E}{N + \hat{h}} \\ -2 \frac{\hat{v}_N}{M + \hat{h}} & -2w_e \cos\hat{\varphi} - 2 \frac{\hat{v}_E}{N + \hat{h}} & 0 \end{pmatrix}. \quad (6.29)$$

The attitude matrix derivative in the n-frame is obtained by multiplying the transformation matrix C_b^n by the differences between Earth-rate and gyroscope measurements [Shi01]. The attitude dynamics can be followed as:

$$\dot{C}_b^n = C_b^n \Omega_{nb}^b = C_b^n (\Omega_{ib}^b - \Omega_{in}^b), \quad (6.30)$$

where Ω_{ib}^b denotes the measurements of the gyroscopes. The equations 6.9 and 6.30 can be combined to obtain the attitude derivative $\dot{\epsilon}^n$:

$$\begin{aligned} -\dot{\epsilon}^n \times C_b^n + (I - \epsilon^n \times) \dot{C}_b^n &= (I - \epsilon^n \times) C_b^n (\Omega_{ib}^b - \Omega_{in}^b + \delta\Omega_{ib}^b - \delta\Omega_{in}^b) \\ &= (I - \epsilon^n \times) C_b^n \Omega_{nb}^b + (I - \epsilon^n \times) C_b^n (\delta\Omega_{ib}^b - \delta\Omega_{in}^b) \end{aligned} \quad (6.31)$$

The above equation can be reduced to $\dot{\epsilon}^n$ [Shi01]

$$\dot{\epsilon}^n \times = -C_b^n (\delta\Omega_{ib}^b - \delta\Omega_{in}^b) C_n^b. \quad (6.32)$$

or in vector form:

$$\dot{\epsilon}^n = -C_b^n (\delta\omega_{ib}^b - \delta\omega_{in}^b). \quad (6.33)$$

The relationship of $\omega_{in}^b = C_n^b w_{in}^n$ can be used to obtain the $\delta\omega_{in}^b$. This can be expanded into

$$w_{in}^b + \delta w_{in}^b = C_b^n (I + \epsilon^n \times) (w_{in}^n + \delta w_{in}^n). \quad (6.34)$$

and

$$\delta w_{in}^b = C_b^n (\delta w_{in}^n + (\epsilon^n \times) w_{in}^n). \quad (6.35)$$

Substituting equations 6.34 and 6.35 in equation 6.33, $\dot{\epsilon}^n$ can be expressed as:

$$\begin{aligned} \dot{\epsilon}^n &= \delta w_{in}^n + (\epsilon^n \times) w_{in}^n - C_b^n \delta w_{ib}^b \\ &= \delta w_{in}^n - (w_{in}^n \times) \epsilon^n - C_b^n \delta w_{ib}^b. \end{aligned} \quad (6.36)$$

Now, the first term on the right hand side of the equation above is expanded into the position and velocity error terms, and the attitude error propagation is defined by using the equations 4.26 and 6.19, which can be expressed as:

$$\dot{\epsilon}^n = F_{er} \delta r^n + F_{ev} \delta v^n - (w_{in}^n \times) \epsilon^n - C_b^n \delta w_{ib}^b, \quad (6.37)$$

where

$$F_{er} = \begin{pmatrix} -w_e \sin \hat{\varphi} & 0 & \frac{-\hat{v}_E}{(N + \hat{h})^2} \\ 0 & 0 & \frac{\hat{v}_N}{(M + \hat{h})^2} \\ -w_e \cos \hat{\varphi} - \frac{-\hat{v}_E}{(N + \hat{h}) \cos^2 \hat{\varphi}} & 0 & \frac{\hat{v}_E \tan \hat{\varphi}}{(N + \hat{h})^2} \end{pmatrix}, \quad (6.38)$$

and

$$F_{ev} = \begin{pmatrix} 0 & \frac{1}{(N + \hat{h})} & 0 \\ \frac{-1}{(N + \hat{h})} & 0 & 0 \\ 0 & \frac{-\tan \hat{\varphi}}{(N + \hat{h})} & 0 \end{pmatrix}. \quad (6.39)$$

The residual biases, b_a and b_g , which represent the uncertainty of the IMU device, are also considered. The relationship between both vectors can be expressed as

$$\delta f_{ib}^b = b_a, \quad \delta w_{ib}^b = b_g, \quad (6.40)$$

As assumed, the biases do not have any variation over the time. Therefore, the time derivatives of the biases can be assumed to be zero [Gro13]:

$$\dot{b}_a = 0, \quad \dot{b}_g = 0. \quad (6.41)$$

6.4. GNSS/IMU system model

The KF system model equation 5.11 can be implemented by using all differential error equations 6.11, 6.27 and 6.37, all equations have to be transformed into a matrix- and state based description as follows [Shi01]:

$$\dot{x} = Fx. \quad (6.42)$$

The system transition matrix F is determined for each epoch and this matrix (15×15)

can be defined as follows:

$$F = \begin{pmatrix} F_{rr} & F_{rv} & 0 & 0 & 0 \\ F_{vr} & F_{vv} & (f^n \times) & C_b^n & 0 \\ F_{er} & F_{ev} & -(w_{in}^n \times) & 0 & C_b^n \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (6.43)$$

The system noise term describes the uncertainties of the dynamic system behavior in the system model. The covariance matrix of the system noise Q of the IMU device consists of the variances of the accelerometers and the gyroscopes, which are random walks of the velocity error due to noise on specific force measurements and the orientation error due to the angular rate measurements [Gro13]. Such system noises are the main noise sources in the navigation solution calculated from the IMU measurements. The unmodeled residual errors in position, velocity, attitude, gravity model and sensor biases are added to the process noise matrix Q . Each of these six residual errors has three dimensions and belongs to the components of the system noise vector or rather the diagonal components of Q . Therefore, matrix Q has a dimension of 18×18 .

$$Q = \text{diag}(\sigma_{r^e}^2 \quad \sigma_{v^n}^2 \quad \sigma_{\epsilon^n}^2 \quad \sigma_{g^n}^2 \quad \sigma_{b_a}^2 \quad \sigma_{b_g}^2). \quad (6.44)$$

The IMU in strapdown mechanization provides the measurements with the high data rate. A discrete Kalman filter algorithm is chosen for the integration of the GNSS and the IMU. The F matrix is substituted through the transition matrix, Φ , by applying equation 5.14. The system noise covariance matrix Q_k can be calculated by means of the equation 5.15, where the G matrix (the system noise distribution matrix) is represented as shown in equation 6.45 below with the dimension of 15×18 due to the gravity model [Bla08]:

$$G = \begin{pmatrix} F_{rv} & 0 & 0 & 0 & 0 & 0 \\ I & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \end{pmatrix}, \quad (6.45)$$

If the norm of Q_k is significant and the obtained measurements are fairly accurate, the Kalman filter trusts Kalman gain computation in the observation and suite less in the system prediction. But the estimated states do not have any time lag. In contrast, if the norm of Q_k is

small, the estimated states experience time lags. Especially, if the norm of Q_k is much smaller, the filter diverges. This causes numerical instabilities of the estimated solutions [Shi01]. In this thesis, the components of Q_k are chosen to be large enough until the filter is stable and the trajectory can follow the GNSS and IMU measurements.

6.5. Loosely coupled GNSS/IMU/ODO integration algorithm

The GNSS-based position and velocity are used to correct the navigation solution estimated by the IMU measurements with the system uncertainty. When the vehicle moves into a tunnel, and the GNSS signal is lost, the vehicle's trajectory can be determined based on the IMU standalone processing. Hence, the GNSS/IMU/ODO integration is considered to improve the positioning accuracy. In general, the odometer provides more accurate velocity information, which is always used as a part of measurements to update the state vector of a Kalman filter. Furthermore, the odometer can provide a measurement with a higher data rate than GNSS. The loosely coupled GNSS/IMU/ODO integration covers the following 16 components [Hem08] :

$$\delta x^n = \left(\delta r_{eb}^n \quad \delta v_{eb}^n \quad \delta \psi_{nb}^n \quad \delta b_a \quad \delta b_g \quad \delta k_{ODO} \right). \quad (6.46)$$

where δk_{ODO} is the scale factor error of the odometer.

As mentioned before, the velocity obtained by the odometer can be calculated by multiplying these pulses by a scale factor. The scale factor is applied to convert the number of pulses to the current velocity of the vehicle, which is the traveling distance per time of a unit [Lee06]. Now, the scale factor error of the odometer must be considered in the integrated system, and the sensor error is usually assumed to be a random constant bias. The velocity measured by the odometer in the n-frame can be defined as follows:

$$v_{ODO}^n = (1 + \delta k_{ODO}) \cdot v_{ODO}^n. \quad (6.47)$$

The block diagram of a loosely coupled GNSS/IMU/ODO sensor integration is depicted in figure 12. The vehicle's position derived by the GNSS signal processing and the velocity measured by the odometer are used to create the measurement model in the loosely coupled Kalman filter. Then they are blended with the measurements from the IMU device to estimate the error state vector. The estimated error state variables are fed back to the navigation equation within the Kalman filter to mitigate the instrumental errors that affect the IMU measurements. The equations 6.8 and 6.9 are used to correct the position, velocity, and attitude solution.

Following the corrections, the attitude error, velocity error, position error and sensor biases of the Kalman filter are set to zero for the next epoch, because the corrections has already been transferred to the navigation solution. However, the error covariance matrix, P , does not change for the next epoch, because the error in the state estimates has not changed [Gro13]. The estimated sensor biases are fed also back to correct the IMU measurements, because they are input for the navigation equations. The attitude solution update of the inertial equation uses the angular rate measurements (see equation 4.20), and the measured specific forces are used to update the velocity and position solution (see equations 4.12a and 4.12b). Furthermore, the odometer can be utilized to handle the GNSS outages in the GNSS/IMU/ODO integrated system. Unlike GNSS, the odometer measures the vehicle's velocity without an auxiliary signal. When the vehicle moves into a tunnel or under a bridge, and the GNSS signal is not available, the vehicle's velocity can be still recovered from the odometer measurements.

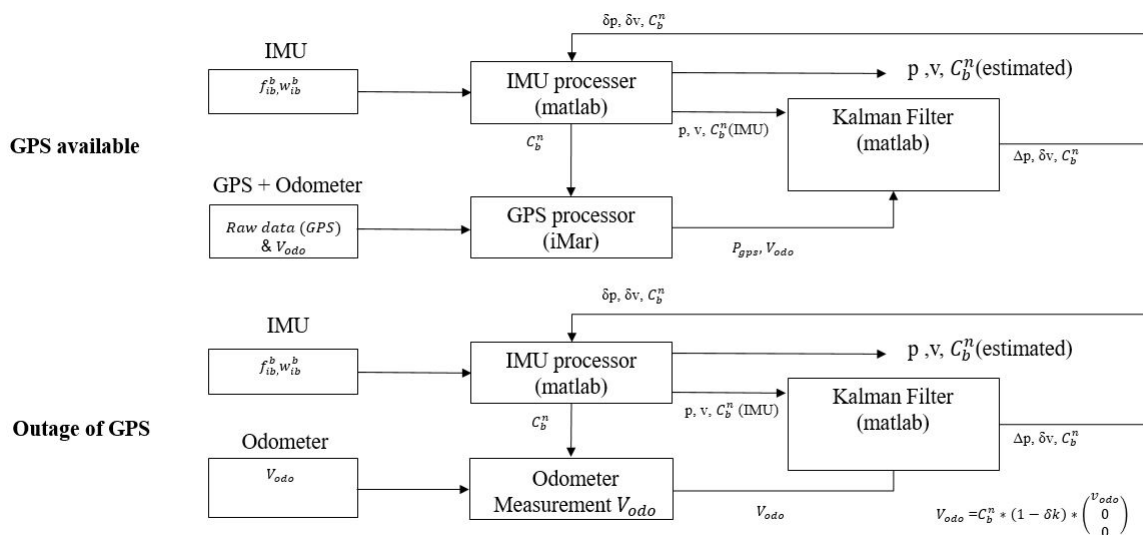


Figure 12: Architecture of a loosely coupled GNSS/IMU/ODO integration. The block diagram at the bottom shows the architecture of the IMU/ODO integration in case of an outage of the GNSS signal

The system model (see equation 6.42) is used for the loosely coupled approach. The dynamic model provides information about the changeable motion of the body at each epoch. The transition matrix F is already designed (see equation 6.43). However, one cell more must be added for the odometer scale factor error into the F matrix in the GNSS/IMU/ODO integrated system. This additional cell only contain zero values [Lee06]. As assumed, the odometer scale factor error is constant. Therefore, the dynamic equations are zero.

Then, the measurement model of the GNSS/IMU/ODO integration bases on the

equation 5.23. The measurement vector z_k in the loosely coupled Kalman filter realized in this thesis is made of the difference between the coordinates calculated by GNSS processing and velocity measured by the odometer with the respect to the corresponding values derived by the IMU measurements.

$$z_k = \begin{pmatrix} r_{IMU}^e - r_{GPS}^e \\ v_{IMU}^n - v_{ODO}^n \end{pmatrix} = \begin{pmatrix} \varphi_{IMU} - \varphi_{GPS} \\ \lambda_{IMU} - \lambda_{GPS} \\ h_{IMU} - h_{GPS} \\ v_{E,IMU} - v_{E,ODO} \\ v_{N,IMU} - v_{N,ODO} \\ v_{D,IMU} - v_{D,ODO} \end{pmatrix}, \quad (6.48)$$

Here, H_k denotes the measurement design matrix at epoch k . The measurements in the GNSS/IMU/ODO loosely coupled integration directly correspond to the position and velocity error states in the n -frame. Therefore, the H_k matrix can be designed as follows[Lee06].

$$H_k = \begin{pmatrix} h_{\varphi,\lambda,h} * I_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{1x3} \\ 0_{3x3} & I_{3x3} & v_{IMU}^n \times & 0_{3x3} & 0_{3x3} & v_{IMU}^n \end{pmatrix}, \quad (6.49)$$

with

$$h_{\varphi,\lambda,h} = \begin{pmatrix} -(M + h) \\ -(N + h)\cos\varphi \\ 1 \end{pmatrix}, \quad (6.50)$$

where h is the ellipsoidal height,

The vector $h_{\varphi,\lambda,h}$ is used to convert from radiants to meter. The coordinate differences in φ and λ between the IMU and the GNSS are compared in radiant [Shi01].

M and N are radius of curvature in meridian and radius of curvature in the prime vertical, which are defined by the equations 4.14 and 4.15. Here, $v_{IMU}^n \times$ is a skew-symmetric matrix for the velocity vector estimated by IMU, which is defined by:

$$v_{ODO}^n \times = \begin{pmatrix} 0 & -v_{D,ODO}^n & v_{E,ODO}^n \\ v_{D,ODO}^n & 0 & -v_{N,ODO}^n \\ -v_{E,ODO}^n & v_{N,ODO}^n & 0 \end{pmatrix} \quad (6.51)$$

It is well-known that the measurement noise is environmentally dependent. The measurement noise covariance matrix, R , models the noise-like errors on the pseudo-range and pseudo-range-rate measurements, such as tracking errors, multipath variations, and satellite clock noise [Gro13]. Each component in the diagonal matrix, R , depends on the accuracy of the GNSS estimates and odometer measurements, which can be obtained from GNSS processing and odometer processing [Shi05].

Finally, the R matrix is calculated as:

$$R_k = \begin{pmatrix} \sigma_\varphi^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_\lambda^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_h^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{v_N}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{v_E}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{v_D}^2 \end{pmatrix}. \quad (6.52)$$

Furthermore, the measurement model of a loosely coupled IMU/ODO integrated system is defined as follows.

$$z_k = \begin{pmatrix} v_{E,IMU} - v_{E,ODO} \\ v_{N,IMU} - v_{N,ODO} \\ v_{D,IMU} - v_{D,ODO} \end{pmatrix}, \quad (6.53)$$

and

$$H_k = \begin{pmatrix} 0_{3 \times 3} & I_{3 \times 3} & v_{IMU}^n \times & 0_{3 \times 3} & 0_{3 \times 3} & v_{IMU}^n \end{pmatrix}. \quad (6.54)$$

The GNSS/IMU/ODO integrated system is reduced to an IMU/ODO integrated system, in times while the GNSS signal is blocked or the quality of the GNSS solution is terrible. The algorithm checks the quality of the GNSS solution through the measurement standard deviation from the GNSS signal processing, and this information is usually provided with the GNSS solution. Finally, the error state vector is reduced to zero after every measurement update.

6.6. Tightly coupled GNSS/IMU integration algorithm

This subsection describes how a GNSS/IMU integration is implemented through a tightly coupled Kalman filter. The tightly coupled integration directly uses the differences between the measured GNSS raw range observations and the ranges derived from the IMU's observation data to estimate the error states, which are fed back into the navigation solution predicted from the last epoch, and corrects this solution for sensor biases. The structure of a tightly coupled algorithm is designed in figure 13. The different GNSS observation processing strategies (DD or SPP) can be implemented in the algorithm. A big advantage of the developed Kalman filter algorithm is that it still works even when the number of the visible GNSS satellites is less than four.

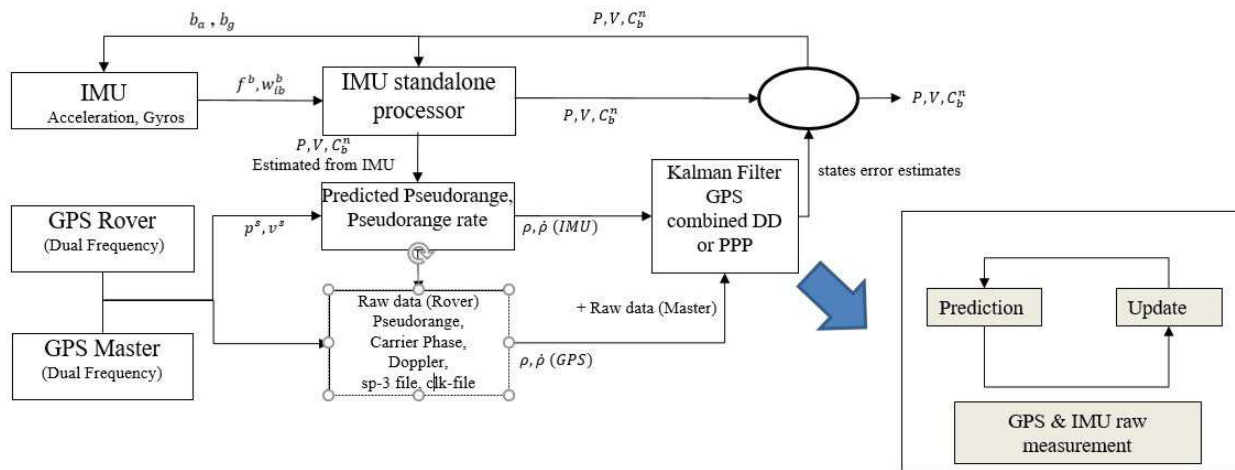


Figure 13: Structure of a tightly Coupled Kalman filter based on DD/SPP combined code and phase observation

Defining the H_k matrix and the measurement vector z_k from the equation 5.23 are the important steps for integrating GNSS/IMU within a Kalman filter. H_k and z_k together make up the measurement model based on the difference in measurements from at least two different sensors. In a tightly coupled approach, the GNSS code range, carrier-phase range and pseudo-range rate observations, obtained from code and carrier-phase and doppler shift tracking, are used for the measurement update. With carrier-phase observables a centimeter-level positioning accuracy in relative positioning mode can be achieved. The differences between the GNSS code ranges, carrier-phase ranges, pseudo-range rates and the corresponding predicted IMU pseudo-ranges, pseudo-range rates at the same epoch constitute the Kalman filter innovation. Therefore, the measurement vector at epoch k comprises the combined code, carrier-phase and

pseudo-range rates differences between both sensors for m satellites tracked.

$$z_k = \begin{pmatrix} \delta z_{\rho,k} \\ \delta z_{\phi,k} \\ \delta z_{r,k} \end{pmatrix}, \quad (6.55)$$

where $\delta z_{\rho,k}$ denotes the code range part of the measurement vector. Here, $\delta z_{\phi,k}$ and $\delta z_{r,k}$ are the carrier-phase and pseudo-range rate parts.

$$\begin{aligned} \delta z_{\rho,k} &= \left(\rho_{r,GNSS}^1 - \rho_{r,IMU}^1, \rho_{r,GNSS}^2 - \rho_{r,IMU}^2, \dots, \rho_{r,GNSS}^m - \rho_{r,IMU}^m \right) \\ \delta z_{\phi,k} &= \left(\lambda \phi_{r,GNSS}^1 - \rho_{r,IMU}^1, \lambda \phi_{r,GNSS}^2 - \rho_{r,IMU}^2, \dots, \lambda \phi_{r,GNSS}^m - \rho_{r,IMU}^m \right). \\ \delta z_{r,k} &= \left(\dot{\rho}_{r,GNSS}^1 - \dot{\rho}_{r,IMU}^1, \dot{\rho}_{r,GNSS}^2 - \dot{\rho}_{r,IMU}^2, \dots, \dot{\rho}_{r,GNSS}^m - \dot{\rho}_{r,IMU}^m \right) \end{aligned} \quad (6.56)$$

The predicted code ranges and carrier-phase ranges $\rho_{r,GNSS}^s$ and $\phi_{r,GNSS}^s$ as measured by the GNSS receiver can be calculated by using the observation equations 3.7 and 3.8. In this research, the GNSS code range and carrier-phase range DD observations are applied as parts of the measurement vector. A few significant error terms such as clock offsets are removed in the DD observation equations. The code range and carrier-phase range measurement equations based on the DD approach are described by the equations 3.28 and 3.29. The measurement vector is described in subsection 6.6.1. Moreover, the ionospheric delay is eliminated by using the dual-frequency ionosphere-free combination. Next, the Saastamoinen model can be used to remove the tropospheric delay. However, the receiver measurement noise and multi-path error cannot be reduced through the DD approach. At last, $\dot{\rho}_{r,GNSS}^s$, the pseudo-range rates, can be calculated by the Doppler shift measurements (see equation 3.21).

$\rho_{r,IMU}^s$, $\phi_{r,IMU}^s$ and $\dot{\rho}_{r,IMU}^s$ are the code range, carrier-phase range and pseudo-range rate observations calculated by the predicted IMU measurements, such as the corrected navigation solution. The distance offset (lever-arm vector) between the GNSS antenna and the IMU device has to be accounted for to refer to the same reference point. Again, the n -frame is chosen for determining the navigation solution, which means that the position vector is defined in the e -frame, and the velocity vector is expressed in the NED coordinate system. For a n -frame implementation, it is easier to compute the code range and pseudo-range rate observations using the position in Cartesian coordinates and the velocity in the e -frame [Gro13]. The position of the sensor in the Cartesian coordinate system is calculated by using equation 4.13, and the

conversion of the velocity into the e-frame can be carried out by:

$$v^n = C_n^e v^e, v^e = C_n^e v^n. \quad (6.57)$$

where C_n^e is the transformation matrix expressed in equation 2.33.

The predicted code ranges, ρ_r^s and pseudo-range rates, $\dot{\rho}_r^s$, in the Cartesian e-frame, are given by [Gro13]:

$$\begin{aligned} \rho_{r,IMU}^s &= \sqrt{(r^{e,s}(t^s) - r_r^e(t_r))^T (r^{e,s}(t^s) - r_r^e(t_r))} + \delta_r + \delta\rho_{ie,r}^s, \\ \dot{\rho}_{r,IMU}^s &= u^{eT} (v^{e,s}(t^s) - v_r^e(t_r)) + \dot{\delta}_r + \delta\dot{\rho}_{ie,r}^s, \end{aligned} \quad (6.58)$$

where:

$r^{e,s}(t^s)$ = the calculated satellite position in the e-frame

$v^{e,s}(t^s)$ = the calculated satellite velocity in the e-frame

$r_r^e(t_r)$ = the position of GNSS antenna phase center related to those of IMU center

$v_r^e(t_r)$ = the velocity of GNSS antenna phase center related to those of IMU center

$\delta\rho_{ie,r}^s$ = the sagnac correction (see equation 6.59)

$\delta\dot{\rho}_{ie,r}^s$ = the range rate Sagnac correction(see equation 6.60)

u^{eT} = the transposed line of sight vector (see equation 6.62)

δ_r = the receiver clock offset

$\dot{\delta}_r$ = the receiver clock drift.

The position of the rover is normally computed in the ECEF, and the current satellite position is calculated by using the ephemeris information contained in the navigation message. Therefore, the range should be computed in the e-frame. But the range in the e-frame neglects the Earth's rotation during the signal transit time, which is known as the Sagnac correction. The Sagnac correction must be compensated, and this error can be up to about 40 m at the equator [Gro13]. The Sagnac correction can be calculated by using the following equation:

$$\delta\rho_{ie,r}^s = \frac{w_e}{c} [y^{e,s}(t^s)x_r^e(t_r) - x^{e,s}(t^s)y_r^e(t_r)]. \quad (6.59)$$

where:

w_e = the rate of the Earth rotation

c = the speed of light

The range rate Sagnac correction is approximated by using the following equation.

$$\delta \dot{\rho}_{ie,r}^s = \frac{w_e}{c} [v_y^{e,s}(t^s)x_r^e(t_r) + y^{e,s}(t^s)v_{r,x}^e(t_r) - v_x^{e,s}(t^s)y_r^e(t_r) - x^{e,s}(t^s)v_{r,y}^e(t_r)]. \quad (6.60)$$

The unit vector describes the direction from the satellite to the user antenna, resolved about the arbitrary frame. An example of this vector can be shown as follows:

$$u_{ba}^d = C_c^d u_{ba}^c. \quad (6.61)$$

The vector u may be formulated either in the e-frame (see equation 6.62) or in the n-frame (see equation 6.64):

$$u^e \approx \frac{r^{e,s}(t^s) - r_r^e(t_r)}{|r^{e,s}(t^s) - r_r^e(t_r)|}. \quad (6.62)$$

In the n-frame, the elevation θ and azimuth ψ define the direction of the line of sight vector with respect to the north, east and down axes. The elevation and azimuth may be obtained by the line-of-sight vector in the n-frame

$$\theta = -\arcsin(u_D^n), \psi = \arctan_2(u_E^n, u_N^n) \quad (6.63)$$

and

$$u^n = \begin{pmatrix} \cos\theta \cos\psi \\ \cos\theta \sin\psi \\ -\sin\theta \end{pmatrix}. \quad (6.64)$$

where:

$$u^n = (u_N^n, u_E^n, u_D^n)$$

In simple phrase, $\rho_{r,GNSS}^s$, $\phi_{r,GNSS}^s$, as well as $\dot{\rho}_{r,GNSS}^s$ denote the GNSS observations. $\rho_{r,IMU}^s$, $\phi_{r,IMU}^s$ as well as $\dot{\rho}_{r,IMU}^s$ are the corresponding predicted observations using corrected states of the IMU navigation solution from the last epoch. These quantities are defined as computed observations. The measurement vector is thus based on observed GNSS-minus-computed IMU observations.

6.6.1. Combined GNSS DD observation

In this research, the double difference code range, carrier-phase observations as well as the single differenced Doppler observations are implemented in the Kalman filter algorithm. The observation model, combines both the code and carrier phase measurements. The observation equation describes pseudo-range and carrier-phase obtained by differencing of master and rover observations for the short baseline, with respect to the pivot satellite and other satellites [Rea09]. There is a set of GNSS satellites simultaneously tracked at dual-frequency with the antenna mounted on a vehicle and at a permanent base station. When using the tightly coupled integration the satellite with the highest elevation has to be defined as the pivot satellite for both stations, which is denoted as p . The code range observation equation in DD mode can be formulated as follows (see chapter 3.5):

$$P(k)_{a,b}^{A,B} = \rho(k)_{a,b}^{A,B} + \delta t_{DP_{a,b}}^{A,B} + \epsilon_{DP_{a,b}}^{A,B}, \quad (6.65)$$

The range difference at DD level between the master and the rover station can be represented as follows:

$$\begin{aligned} \rho(k)_{a,b}^{A,B} &= \rho(k)_{r,m}^{p,s} \\ &= [\rho(k)_r^p - \rho(k)_m^p] - [\rho(k)_r^s - \rho(k)_m^s]. \end{aligned} \quad (6.66)$$

where

k = the epoch number

$s = 1, 2, \dots, s$ for other satellites

p = the pivot satellite

r = the rover

m = the master station

Note, that due to a chosen pivot satellite, the number of available GNSS observation is reduced to $s-1$.

The carrier-phase range observation equation at the DD level is described by the equation 3.29. In case that the baseline between both stations is less than 10 km, the equation can be approximated as:

$$\lambda \phi(k)_{a,b}^{A,B} = \rho(k)_{a,b}^{A,B} + \lambda N_{a,b}^{A,B} + \epsilon_{a,b\phi}^{A,B}. \quad (6.67)$$

where $N_{a,b}^{A,B}$ is the DD ambiguity value.

$$\begin{aligned} N_{a,b}^{A,B} &= N_{r,m}^{p,s} \\ &= [N_r^p - N_m^p] - [N_r^s - N_m^s]. \end{aligned} \quad (6.68)$$

One thing to note here is that the DD ambiguity value of the pivot satellite $N_{r,m}^{p,p}$ is set to zero. Now, $N_{r,m}^{p,s}$ can be represented as follows [Rea09]:

$$\left(\begin{array}{l} N_{r,m}^{p,1} = [N_r^p - N_m^p] - [N_r^1 - N_m^1] \\ N_{r,m}^{p,2} = [N_r^p - N_m^p] - [N_r^2 - N_m^2] \\ \vdots \\ N_{r,m}^{p,p} = 0 \\ \vdots \\ N_{r,m}^{p,s} = [N_r^p - N_m^p] - [N_r^s - N_m^s] \end{array} \right). \quad (6.69)$$

6.6.2. Kalman filter initialization

The Kalman filter initialization is a quite important step, which is performed to initialize the state vector. The initialization starts with a first estimation of the rover position and velocity, also the initial ambiguities can be estimated. Approximate rover coordinates can be obtained from the header of the related rinex file or can be estimated in real-time by an initial LSQ algorithm. Assuming that the GNSS rover is at rest, the initial velocity of the rover is set to zero. The approximate position and velocity are used as the inputs to perform the linearized LSQ (see chapter 5.1) based on the code-only double difference. The few decimeter-level positioning accuracy could be achieved by using the GNSS code observations in the relative positioning mode. Hence, the estimated position, velocity and ambiguities can be defined as the initial state variables in the Kalman filter.

The implemented algorithm accepts the float solution due to the sensor's dynamics. Generally, there are two approaches to calculate approximated float values. The first approach is based on the geometry-free technique, the non-ambiguous code and carrier phase measurements are used as an additional wavelength to resolve the float ambiguity [See03] and [Rea09]:

$$\rho_{r,m}^{p,s} - \phi_{r,m}^{p,s} = \lambda N_{r,m}^{p,s}. \quad (6.70)$$

The LSQ based float ambiguity estimation is proposed as the second approach, which

provides a more accurately estimated float solution. The ambiguities are usually fixed when the values are close to integer within a 3σ variance range.

The developed algorithm starts automatically with the determination of the initial position and velocity of the rover by using a double difference linearized LSQ algorithm based on the combined GNSS observation (see chapter 5.1). At the same time, the initial float ambiguity values are also estimated as part of this LSQ, which are very close to integer values. In case that there are only observations of two satellites available, the first approach will be activated for calculating the initial ambiguities.

In the DD approach, error terms such as $\delta t_{r,m}^{p,s}$ (see equations 3.6 and 3.9) in the observation equation can be corrected by using the ionosphere free linear combination (see chapter 3.3.2) and the correction models [e.g. Saastamoinen model (see chapter 3.3.1)]. The DD GNSS code observation equation (see equation 6.65) can be simplified as:

$$\rho_{r,m}^{p,s} = \rho(\bar{k})_{r,m}^{p,s} + \epsilon_{r,m}^{p,s}, \quad (6.71)$$

On the one hand, the coordinates of the master station, r_m , are accurately known. On the other hand, the coordinates of the rover, r_r , are usually unknown. Here, $x(\bar{k})^s$ is the satellite coordinate vector at the epoch \bar{k} , calculated by using the ephemeris. Now, $\rho(\bar{k})_{r,m}^{p,s}$ can be obtained by using the equation 6.66. In the LSQ algorithm, the equation above must be linearized as follows [Rea09]:

$$\begin{aligned} \rho(k)_{r,m}^{p,s} = & \left[\frac{x_r(\bar{k}) - x^p(k)}{\rho_r^p} - \frac{x_r(\bar{k}) - x^s(k)}{\rho_r^s} \right] (x_r(\bar{k}) - x_r(k)) + \\ & \left[\frac{y_r(\bar{k}) - y^p(k)}{\rho_r^p} - \frac{y_r(\bar{k}) - y^s(k)}{\rho_r^s} \right] (y_r(\bar{k}) - y_r(k)) + \\ & \left[\frac{z_r(\bar{k}) - z^p(k)}{\rho_r^p} - \frac{z_r(\bar{k}) - z^s(k)}{\rho_r^s} \right] (z_r(\bar{k}) - z_r(k)) + \rho(\bar{k})_{r,m}^{p,s} + \epsilon_{r,m}^{p,s} \end{aligned} \quad (6.72)$$

The abbreviation $a_k = \left[\frac{r_r(\bar{k}) - r^p(k)}{\rho_r^p(k)} - \frac{r_r(\bar{k}) - r^s(k)}{\rho_r^s(k)} \right]$ are used to simplify the equation above, which becomes,

$$\rho(k)_{r,m}^{p,s} = a_{x_r}^{p,s} (x_r(\bar{k}) - x_r(k)) + a_{y_r}^{p,s} (y_r(\bar{k}) - y_r(k)) + a_{z_r}^{p,s} (z_r(\bar{k}) - z_r(k)) + \rho(\bar{k})_{r,m}^{p,s} + \epsilon_{r,m}^{p,s}. \quad (6.73)$$

where:

$r_r(k)$ = the rover position

$r_r(\bar{k})$ = the rover position predicted from this value at time k-1 (see figure 10)

$r^s(k)$ = the satellite position

$r^p(k)$ = the position of the pivot satellite

The measurement vector in the LSQ estimation consists of the code part and carrier-phase part. The code part can be defined as follows:

$$z_{k,code} = \begin{pmatrix} \rho(k)_{r,m}^{p,1} + a_{x_r}^{p,1}(x_r(\bar{k})) + a_{y_r}^{p,1}(y_r(\bar{k})) + a_{z_r}^{p,1}(z_r(\bar{k})) - \rho(\bar{k})_{r,m}^{p,1} \\ \rho(k)_{r,m}^{p,2} + a_{x_r}^{p,2}(x_r(\bar{k})) + a_{y_r}^{p,2}(y_r(\bar{k})) + a_{z_r}^{p,2}(z_r(\bar{k})) - \rho(\bar{k})_{r,m}^{p,2} \\ \vdots \\ \tilde{\rho}(k)_{r,m}^{p,s} + a_{x_r}^{p,s}(x_r(\bar{k})) + a_{y_r}^{p,s}(y_r(\bar{k})) + a_{z_r}^{p,s}(z_r(\bar{k})) - \rho(\bar{k})_{r,m}^{p,s} \end{pmatrix}, \quad (6.74)$$

where p is the pivot satellite and s is the visible satellites.

$\rho(\bar{k})_{r,m}^{p,s}$ and $\rho(k)_{r,m}^{p,s}$ are the estimated and measured observations. Both represent the code range differences between the master and the rover station. The calculation of the first observation term, $\rho(\bar{k})_{r,m}^{p,s}$, which can be carried out based on the calculated satellite position and the estimated position of the rover and master station. The second term, $\rho(k)_{r,m}^{p,1}$ can simply be obtained from the GNSS pseudo-range and carrier-phase measurement.

The H matrix can be defined as follows:

$$H_{k,code} = \begin{pmatrix} a_{x_r}^{p,1} & a_{y_r}^{p,1} & a_{z_r}^{p,1} \\ a_{x_r}^{p,2} & a_{y_r}^{p,2} & a_{z_r}^{p,2} \\ \vdots & \vdots & \vdots \\ a_{x_r}^{p,s} & a_{y_r}^{p,s} & a_{z_r}^{p,s} \end{pmatrix}. \quad (6.75)$$

The carrier-phase observation equation based on DD approach (see equation 6.67) can be simplified as:

$$\lambda\phi(k)_{r,m}^{p,s} = \rho(k)_{r,m}^{p,s} + \lambda N_{r,m}^{p,s} + \epsilon_{r,m\phi}^{p,s}. \quad (6.76)$$

The following linearized equation is obtained:

$$\lambda\phi(k)_{r,m}^{p,s} = a_{x_r}^{p,s}(x_r(\bar{k}) - x_r(k)) + a_{y_r}^{p,s}(y_r(\bar{k}) - y_r(k)) + a_{z_r}^{p,s}(z_r(\bar{k}) - z_r(k)) + \lambda N_{r,m}^{p,s} + \rho(\bar{k})_{r,m}^{p,s} + \epsilon_{r,m}^{p,s}. \quad (6.77)$$

Thus carrier-phase contribution of the measurement vector can be expressed by:

$$z_{k,phase} = \begin{pmatrix} \lambda\phi(k)_{r,m}^{p,1} + a_{x_r}^{p,1}(x_r(\bar{k})) + a_{y_r}^{p,1}(y_r(\bar{k})) + a_{z_r}^{p,1}(z_r(\bar{k})) - \rho(\bar{k})_{r,m}^{p,1} \\ \lambda\phi(k)_{r,m}^{p,2} + a_{x_r}^{p,2}(x_r(\bar{k})) + a_{y_r}^{p,2}(y_r(\bar{k})) + a_{z_r}^{p,2}(z_r(\bar{k})) - \rho(\bar{k})_{r,m}^{p,2} \\ \vdots \\ \lambda\phi(k)_{r,m}^{p,s} + a_{x_r}^{p,s}(x_r(\bar{k})) + a_{y_r}^{p,s}(y_r(\bar{k})) + a_{z_r}^{p,s}(z_r(\bar{k})) - \rho(\bar{k})_{r,m}^{p,s} \end{pmatrix}. \quad (6.78)$$

Now, the measurement vector is completed by combining the code part and carrier-phase part, $z_k = [z_{k,code}; z_{k,phase}]$. The H_k matrix for carrier-phase observations also includes the ambiguity values, which can be defined as follows:

$$H_{k,phase} = \begin{pmatrix} a_{x_r}^{p,1} & a_{y_r}^{p,1} & a_{z_r}^{p,1} & \lambda & 0 & \cdots & 0 \\ a_{x_r}^{p,2} & a_{y_r}^{p,2} & a_{z_r}^{p,2} & 0 & \lambda & \vdots & 0 \\ \vdots & \vdots & \vdots & \vdots & 0 & \vdots & \vdots \\ a_{x_r}^{p,s} & a_{y_r}^{p,s} & a_{z_r}^{p,s} & 0 & \cdots & 0 & \lambda \end{pmatrix}. \quad (6.79)$$

The wavelength λ is set to the suitable column and rows in the matrix, where the visible satellite in the observation refers to. The right choice for columns and rows depends on the order of number of the corresponding satellite.

Here, the matrices, $H_{k,code}$ and $H_{k,phase}$, have different dimensions. The dimension of $H_{k,code}$ is $m \times 3$ and $H_{k,phase}$ is a $m \times (3+32)$ matrix, where m is the number of the visible satellite. To combine the both parts, the $H_{k,code}$ has to be changed by adding the 32 columns with zero values. Now, both parts can be combined to complete the H matrix.

$$H_k = \begin{pmatrix} H_{k,code} \\ H_{k,phase} \end{pmatrix}. \quad (6.80)$$

After this processing, the estimated position, velocity and float ambiguity values are used as initial state variables for the Kalman filter algorithm.

6.6.3. Re-initialization

Changing of the tracked satellite signals within the GNSS observations occur frequently due to the receiver's dynamics, attitude change and environmental factors [Rea09]. Sometimes, one or more GNSS satellites could be lost temporarily in the observation vector. Sometimes, one or more new satellites are obtained in the next epoch. This causes the number of ambiguities

to vary within the Kalman filter processing, and the satellite additions/losses or changing of the pivot satellite in the rover and master station must be managed. When the signal of an individual satellite is lost or a new satellite observation has to be added, the corresponding ambiguity in the observation equation must be reinitialized. The ambiguity initialization means that the ambiguity value becomes unknown again, and the value must be initialized by using the LSQ estimation. Furthermore, cycle slips also lead to a re-initialization in the processing. GNSS signal blockage causes the receiver temporarily losing signal lock on the carrier of a signal. This can cause a discontinuity in the number of cycles in the time series of the measured carrier phase, known as a cycle slip [Lan14]. In this case, the previously known ambiguity value of the affected satellite is not valid any longer. Hence, the satellite needs to be initialized once more for the processing. The software algorithm will accept a new estimated set of the ambiguities when there is a change in the number of observed satellites. This always causes a small jump in the coordinates when the new ambiguities are reinitialized.

The first case considered is the loss of one satellite in the GNSS observation vector. This satellite was available in the observation at the epoch t_{k-1} . However, this satellite disappeared at the epoch t_k for any reason. In this case, the developed algorithm would change the H_k matrix, the transition matrix ϕ_k . Of course, the ambiguity value of the lost satellite is deleted. In the H_k matrix the two rows of code and phase measurements of the corresponding satellite are deleted. The ambiguity value of the corresponding satellite will be also set to zero. The rows in the ϕ_k matrix will not be changed, but the row of the corresponding satellite is set to zero, so that the lost satellite would not affect the ambiguity of the other satellites. The algorithm carries out this process automatically, as soon as the satellite is lost.

In the second case, a new satellite has to be added to the GNSS observation vector. The new satellite was not available in the observation at the epoch t_{k-1} . However, this satellite becomes visible in the observation at the epoch k . The algorithm would also change the H_k matrix and the ambiguity value. The H_k matrix is expanded by two new rows covering in the combined code and phase measurements. The float ambiguity value of the new satellite will also be reinitialized.

The third case considers the change of the pivot satellite in the DD carrier phase observations in the Kalman filter. This means that a different satellite p_1 may reach a higher elevation than the current pivot satellite p , and this one would upgrade to the new pivot satellite. In order to adjust the model after changing the pivot satellite, the ambiguity of the p_1 satellite must be set to zero (see equation 6.81). Q, P, R and H matrices in the Kalman filter algorithm (see chapter 5.3) have to be adjusted such that the new pivot satellite is used. The rows

of p_1 in the matrix are removed, and the new rows based on the p satellite are added. Also the observations of all satellites in the z_k vector are subtracted from the new pivot satellite observations. Furthermore, to change the pivot satellite requires all ambiguities to be referred to the new pivot [Rea09].

$$\begin{pmatrix} N_{r,m}^{p_1,1} = [N_r^{p_1} - N_m^{p_1}] - [N_r^1 - N_m^1] \\ N_{r,m}^{p_1,p_1} = 0 \\ \vdots \\ N_{r,m}^{p_1,p} = [N_r^{p_1} - N_m^{p_1}] - [N_r^p - N_m^p] \\ \vdots \\ N_{r,m}^{p_1,s} = [N_r^{p_1} - N_m^{p_1}] - [N_r^s - N_m^s] \end{pmatrix}. \quad (6.81)$$

where:

p = the current pivot satellite

p_1 = the new pivot satellite

In order to avoid a frequent variation of the pivot satellite, the elevation of the corresponding satellite has to be at least 1 degree higher than the current p satellite to become the new pivot satellite.

6.6.4. Cycle slip detection

A cycle-slip occurs between two epochs when the GNSS satellite signals to the receiver are blocked due to e.g. trees or buildings. The second reason for a cycle slip is that the signal-to-noise ratio is low due to changing the ionospheric conditions (rapid changes of electron density along a signal path from a satellite to a GNSS receiver), multipath, high receiver dynamics, or low satellite elevation (< 10 degrees) [See03] and [Dac15]. If the GNSS receiver loses the carrier-phase lock of the satellite signal, the carrier-phase in the GNSS observation at the current epoch can suddenly jump. The cycle counting has to restart due to signal interruption.

The developed algorithm uses the Doppler based detection to detect a cycle slip with a threshold of 1m in the DD carrier-phase observations. This detection bases on the Doppler integration predicted range from the last epoch compared with range measured at the current epoch. This can be implemented by using the following equation [CP14].

$$\phi(t) - (\phi(t-1) + \int_{t-1}^t \delta_{f_D} d_t) < Threshold. \quad (6.82)$$

where

δ_{fD} = the Doppler shift measurement [Hz]

$\phi(t - 1)$ = the carrier phase observation at t-1 epoch [m]

$\phi(t)$ = the carrier phase observation at t epoch [m]

d_t = the GNSS measurement sampling interval [s]

$\phi(t)$ denotes the ionosphere-free linear combination of dual-frequency signal carrier phase observations. When the GNSS rover is processing in static mode, cycle slips are easily detected. However, when the rover is operating in kinematic mode, the cycle slips can not be easily detected, and the threshold should be set a magnitude of several cycles [CP14]. While the difference between the predicted range using Doppler integration and the measured range is less than the empirical threshold 1m (e.g. 10 cycles \approx 2 m error [Rea09]), the cycle slip detection indicates that a cycle slip occurs at the current epoch.

If a cycle slip of a satellite is detected, the corresponding satellite can be treated as a new satellite to the current observation. Thus the initial ambiguities of this satellite should be reinitialized. If a cycle slip on the pivot satellite is detected, all initial ambiguities should be reinitialized.

6.6.5. Combined GNSS DD observation/IMU integration

The state vector (see equation 6.1) within a tightly coupled integration based on the combined GNSS observation is expanded to a 15 components vector including the ambiguity offset of all tracked GNSS satellites [Rea09]:

$$\delta x^n = \left(\delta r_{eb}^e \quad \delta v_{eb}^n \quad \psi_{eb}^n \quad b_a \quad b_g \quad \delta N_{r,m}^{p,1}, \dots, \delta N_{r,m}^{p,s} \right). \quad (6.83)$$

δN are the estimated ambiguity offset from carrier-phase observation of all satellites. The ambiguity offset parameter is set to zero for those satellites currently not visible.

The measurement model of GNSS/IMU integration within the tightly coupled Kalman filter is constructed by differencing the observations between the GNSS measured pseudo-ranges, carrier-phases and pseudo-range rates and the quantities calculated from the IMU observations at every epoch (see chapter 6.6). The IMU based observations are calculated from predicted observations using estimated states of IMU navigation solution from the last epoch. Like the GNSS DD combined approach the measurement vector z_k in the GNSS DD combined/IMU integration also comprises of three measurement parts (see equation 6.56), the first part $\delta z_{\rho,k}$

concerns the DD code measurements created by differencing two measurements based on the equations 6.56, 6.65 and 6.66.

$$\begin{aligned} \delta z_{\rho,k} = & [\rho(k)_{r,GNSS}^p - \rho(k)_{m,GNSS(obs)}^p] - [\rho(k)_{r,GNSS}^s - \rho(k)_{m,GNSS(obs)}^s] \\ & - [\rho(k)_{r,IMU}^p - \rho(k)_{m,GNSS(cal)}^p] - [\rho(k)_{r,IMU}^s - \rho(k)_{m,GNSS(cal)}^s]. \end{aligned} \quad (6.84)$$

where

$\rho(k)_{r,GNSS}^s$ = the pseudo-ranges from the rover to the visible satellites measured by the rover

$\rho(k)_{r,IMU}^s$ = the calculated pseudo-ranges from the rover position derived by IMU to the visible satellites

$\rho(k)_{m,GNSS(obs)}^s$ = the pseudo-ranges from the reference station to the visible satellites measured by the reference station

$\rho(k)_{m,GNSS(cal)}^s$ = the calculated pseudo-ranges based on the known position of the reference station and the calculated satellite position at current epoch

The IMU measurements are used to calculate a IMU-rover position, which is subsequently used for deriving the $\rho(k)_{r,IMU}^s$ and $\rho(k)_{r,IMU}^p$ by means of the equation 6.58. Next, the current position of the corresponding satellites can be calculated by the given ephemeris information (n-file). Furthermore, the coordinate of the reference station is normally known, or the position can be calculated by using a static GNSS data processing software such as the raPPPid software developed at TU Vienna. Therefore, $\rho(k)_{m,GNSS(cal)}^s$ and $\rho(k)_{m,GNSS(cal)}^p$ can be easily obtained. The subscript p denotes the pivot satellite at the current epoch.

$\delta z_{\phi,k}$ concerns the DD carrier-phase measurements created by differencing two measurements based on the equations 6.67, 6.68 and 6.69.

$$\begin{aligned} \delta z_{\phi,k} = & ([\lambda\phi(k)_{r,GNSS}^p - \lambda\phi(k)_{m,GNSS(obs)}^p] - [\lambda\phi(k)_{r,GNSS}^s - \lambda\phi(k)_{m,GNSS(obs)}^s] + \lambda N_{r,m}^{p,s}) \\ & - [\rho(k)_{r,IMU}^p - \rho(k)_{m,GNSS(cal)}^p] - [\rho(k)_{r,IMU}^s - \rho(k)_{m,GNSS(cal)}^s]. \end{aligned} \quad (6.85)$$

where

$\lambda\phi(k)_{r,GNSS}^s$ = the carrier-phase measurements from the rover to the visible satellites measured by the rover

$\lambda\phi(k)_{m,GNSS(obs)}^s$ = the carrier-phase measurements from the reference station to the visible satellites measured by the reference station

$N_{r,m}^{p,s}$ = the float double difference ambiguities for each visible satellites with respect to the pivot satellite

$\lambda\phi(k)_{r,GNSS}^s$ and $\lambda\phi(k)_{m,GNSS}^s$ are the carrier-phases measurements based on the ionosphere-free linear combination by using the equation 6.67.

The third observation $\delta z_{r,k}$ concerns the differences between the GNSS measured pseudo-range rates and the IMU estimated pseudo-range rates based on the equations 3.21, 3.22, 6.53, 6.56 and 6.58.

$$\delta z_{r,k} = \dot{\rho}_{r,GNSS}^S - \dot{\rho}_{r,IMU}^S. \quad (6.86)$$

All three parts create the integrated measurement vector to estimate the error states. The developed algorithm does not only support the combination of GNSS code ranges with IMU measurements, but also the GNSS code and carrier-phase combined processing. When the code-only processing option is chosen, the algorithm is more simple and faster than the combined processing, but the quality of the calculated solution is limited.

The H_k matrix bases on the combined code and carrier phase approach for a n-frame and is defined by

$$H_k = [H_{k,1}; H_{k,2}], \quad (6.87)$$

where $H_{k,1}$ is a $[m \times 3, 9]$ matrix, which can be expressed as:

$$H_{k,1} = \begin{pmatrix} h_\phi u_{a1,N}^n & h_\lambda u_{a1,E}^n & u_{a1,D}^n & 0 & 0 & 0 & 0 & 0 & 0 \\ h_\phi u_{a2,N}^n & h_\lambda u_{a2,E}^n & u_{a2,D}^n & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_\phi u_{am,N}^n & h_\phi u_{am,E}^n & u_{a0,D}^n & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & h_\phi u_{a1,N}^n & h_\lambda u_{a1,E}^n & u_{a1,D}^n & 0 & 0 & 0 \\ 0 & 0 & 0 & h_\phi u_{a2,N}^n & h_\lambda u_{a2,E}^n & u_{a2,D}^n & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & h_\phi u_{am,N}^n & h_\lambda u_{am,E}^n & u_{a0,D}^n & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -u_{a1,N}^n & -u_{a1,E}^n & -u_{a1,D}^n \\ 0 & 0 & 0 & 0 & 0 & 0 & -u_{a2,N}^n & -u_{a2,E}^n & -u_{a2,D}^n \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & -u_{am,N}^n & -u_{am,E}^n & -u_{am,D}^n \end{pmatrix}. \quad (6.88)$$

where

$$h_\phi = -(M + h), h_\lambda = -(N + h)\cos\varphi. \quad (6.89)$$

Furthermore, $H_{k,2}$ is a $[m \times 3, 32]$ matrix, which is displayed below. The 32 columns

refer to the maximum number of GNSS satellites.

$$H_{k,2} = \begin{pmatrix} 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \vdots & 0 & \vdots & 0 & \vdots & 0 & \vdots & 0 \\ 0 & \vdots & 0 & \vdots & 0 & \vdots & 0 & \vdots & 0 \\ 0 & \vdots & 0 & \vdots & 0 & \vdots & 0 & \vdots & 0 \\ -\lambda & 0 & 0 & \vdots & 0 & \vdots & 0 & \vdots & 0 \\ 0 & 0 & 0 & -\lambda & 0 & 0 & 0 & \vdots & 0 \\ 0 & \vdots & 0 & \vdots & 0 & -\lambda & 0 & 0 & 0 \\ 0 & \vdots & 0 & \vdots & 0 & 0 & 0 & -\lambda & 0 \\ 0 & \vdots & 0 & \vdots & 0 & \vdots & 0 & \vdots & 0 \\ 0 & \vdots & 0 & \vdots & 0 & \vdots & 0 & \vdots & 0 \\ 0 & \vdots & 0 & \vdots & 0 & \vdots & 0 & \vdots & 0 \\ 0 & \vdots & 0 & \vdots & 0 & \vdots & 0 & \vdots & 0 \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \end{pmatrix}, \quad (6.90)$$

The $-\lambda$ in the suitable column in the carrier-phase observation rows in the $H_{k,2}$ matrix refers to the visible m satellites.

In the developed algorithm, an observation weighting is considered. The measurement noise covariance matrix R is elevation-angle-dependent. This a priori elevation dependent weighting matrix can be described as follows [HWD17]:

$$R_k = \begin{pmatrix} \sigma_{\rho_1}^2 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_{\rho_2}^2 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{\rho_m}^2 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \sigma_{\phi_1}^2 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & \sigma_{\phi_2}^2 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & \sigma_{\phi_m}^2 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \sigma_{\rho_1}^2 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & \sigma_{\rho_2}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & \sigma_{\rho_m}^2 \end{pmatrix}, \quad (6.91)$$

and

$$\sigma_{\rho}^2 = \frac{\sigma_{code}^2}{\sin^2(el)}, \sigma_{\phi}^2 = \frac{\sigma_{phase}^2}{\sin^2(el)}, \sigma_{\dot{\rho}}^2 = \frac{\sigma_{Doppler}^2}{\sin^2(el)}. \quad (6.92)$$

where

σ_{code} = the double differenced pseudo-range noise standard deviation (typically about 0.1 m)

σ_{phase} = the double differenced carrier-phase noise standard deviation (typically about 0.008 m)

$\sigma_{doppler}$ = the Doppler measurement noise standard deviation (typically about 0.1 m)

$\sin^2(el)$ = the elevation-based weighting function

The quality of the GNSS observations depends on the quality of satellite products and precision of the receiver's signal tracking [Gla22]. The approach cannot remove receiver measurement noise and multi-path error, the measurement noise of GNSS observations is environmentally dependent. The components of the R matrix were computed by using measurement's standard deviation and mapping this quantity according to the current elevation angle. Ordinarily, all GNSS satellites are weighed equally.

Finally, the state errors are estimated within the Kalman filter, the corrections have to be fed back to the IMU state error propagation algorithm by using the equations 6.8 and 6.9.

7. Test drives and results

This chapter describes two test field scenarios concerning vehicle positioning performance based on the previously discussed Kalman filter approaches.

The first scenario concerns a test drive in the area of Kaiserebersdorf near Vienna, which was carried out with a train from the Austrian Federal Railway company. The train has been equipped with a GNSS antenna and a dual-frequency, dual-system receiver from the NovAtel Company (OEM7700). An IMU device of the iMar company (iRailLoc-C) and an odometer has been mounted on the train. The data sets are composed of about half an hour of recorded sensor measurements, which contain static and kinematic parts. The test drive covers more than 7 km driving distance. The test data set contains the GNSS solution, odometer data, as well as the corresponding IMU measurements. Sometimes, a limited positioning quality is expected when the GNSS signal fails (e.g., tunnels and urban areas) in this scenario. The wheel speed sensor (odometer) is added as the third sensor to ensure the availability of a reliable trajectory in these areas. There are two time periods when the GNSS data are unavailable during the test drive. The integrated system can switch to IMU/odometer mode in periods of GNSS outages. Finally, a loosely coupled filter solution has been processed. The solution is compared to a commercial reference solution generated by the iXCOM-CMD (iMAR).

The second scenario concerns three test drives in the area of Graz. These test drives were carried out by the working group Navigation from the institute of geodesy at Graz University of technology. The car has also been equipped with a GNSS antenna and a dual-frequency, dual-system receiver from the JAVAD company. In addition an IMU device of the iMar company has been mounted on the car's roof. A permanent EPOSA station located on the roof of the Graz main train station was chosen as the GNSS reference station. The data sets are composed of about 1 hour of recorded sensor measurements covering static and kinematic parts. The test drive covers more than 11 km driving distance. The reference station is located within 12 km of all rover points along the test route, and delivers reference GNSS observables at 1 Hz update rate. Each data set contains the raw GPS measurements of the rover and reference station as well as the corresponding IMU measurements. Also the reference solution processed by the waypoint software was provided.

7.1. The 1st test drive scenario

7.1.1. The iMar inertial measurement system iRailLoc-C

The employed IMU device is a strapdown high accurate navigation-grade device from the iMar family type iRailLoc-C. This system is a member of the advanced iNAT (iMar Navigation and Timing) system family (see figure 14). It provides the required information such as positioning, navigation, timing and communication all in one box.



Figure 14: IMU device type iRailLoc-C. [iMa21]

This box especially assembled for train applications was installed on a train of the Austrian Federal Railway company ÖBB for the first time (see figure 15). The box offers sensor data fusion and radio communication within one enclosure. The iRailLoc-C can provide the motion and position data for train location. Depending on the required real-time accuracy, iRailLoc-C supports the integration combining IMU, GNSS, and other external sensors such as an odometer to obtain position and/or velocity of the train in real time. All kinematic measurements like acceleration, angular rate, attitude, true heading, velocity and position of the rail vehicle can be collected inclusive standard deviation with data rate of up to 500 Hz.



Figure 15: ÖBB tests of the inertial measurement system iRailLoc-C [iMa21]

iMar iRailLoc-C is equipped with three robust MEMS gyroscopes and three accelerometers. The sensor characteristics, such as noise, bias and scale factor stability, linearity, acceleration sensitivity are listed in technical data sheets (see appendix B). Table 4 gives an overview of the most important specifications of the IMU device.

iNAV iRailLoc-c	
Gyroscope performance	
Random walk	$< 0.15^\circ/\sqrt{h}$
Drift	$< 0.5^\circ/\text{hr}$
Accelerometer performance	
Random walk	$< 23 \mu g\sqrt{Hz}$
Bias	$< 1 mg$
Data Rate	500 Hz

Table 4: Specifications of the device iRailLoc-C [iMa21]

GNSS antenna and IMU device were mounted at the calibrated location on the train (see figure 15). The IMU device and the odometer have been synchronized to GPS-time. The GNSS signals are measured with a NovAtel GNSS receiver at 1 Hz update rate. Also the odometer update rate was set to 1 Hz. At the same time, the IMU output data rate is set to 500 Hz. All those data are applied as input data for data fusion to provide the train's trajectory.

The iMar company provides the information of the gyro random noise with 4.19860963934^{-6} [$rad/s^2/\sqrt{Hz}$] and the accelerometer random noise with 7.0797576759^{-6} [$m/s/\sqrt{Hz}$] to calculate the reference solution trajectory of the train.

```

X
Root Noise [X: 4.36332312599e-05 [rad/s/sqrt(Hz)] [Y: 4.36332312599e-05 [rad/s/sqrt(Hz)] [Z: 4.36332312599e-05 [rad/s/sqrt(Hz)] ]
Quantization [X: 0.00000000000e+00 [rad/s] [Y: 0.00000000000e+00 [rad/s] [Z: 0.00000000000e+00 [rad/s] ]
Bias Standard Deviation [X: 6.98131700798e-03 [rad/s] [Y: 6.98131700798e-03 [rad/s] [Z: 6.98131700798e-03 [rad/s] ]
Bias Random Walk [X: 4.19860963943e-06 [rad/s^2/sqrt(Hz)] [Y: 4.19860963943e-06 [rad/s^2/sqrt(Hz)] [Z: 4.19860963943e-06 [rad/s^2/sqrt(Hz)] ]
Scale Factor Std Dev [X: 6.00000000000e-03 [rad] [Y: 6.00000000000e-03 [rad] [Z: 6.00000000000e-03 [rad] ]
Scale Factor Random Walk [X: 1.00000000000e-09 [1/s/sqrt(Hz)] [Y: 1.00000000000e-09 [1/s/sqrt(Hz)] [Z: 1.00000000000e-09 [1/s/sqrt(Hz)] ]
Misalign Std Dev [X: 1.00000000000e-03 [rad] [Y: 1.00000000000e-03 [rad] [Z: 1.00000000000e-03 [rad] ]
Misalign Random Walk [X: 0.00000000000e+00 [rad/s/sqrt(Hz)] [Y: 0.00000000000e+00 [rad/s/sqrt(Hz)] [Z: 0.00000000000e+00 [rad/s/sqrt(Hz)] ]

```

Figure 16: Noise of the gyroscopes provided by the iMar company

```

Y
Root Noise [X: 2.94300000000e-04 [m/s^2/sqrt(Hz)] [Y: 2.94300000000e-04 [m/s^2/sqrt(Hz)] [Z: 2.94300000000e-04 [m/s^2/sqrt(Hz)] ]
Quantization [X: 0.00000000000e+00 [m/s^2] [Y: 0.00000000000e+00 [m/s^2] [Z: 0.00000000000e+00 [m/s^2] ]
Bias Standard Deviation [X: 4.90500000000e-02 [m/s^2] [Y: 4.90500000000e-02 [m/s^2] [Z: 4.90500000000e-02 [m/s^2] ]
Bias Random Walk [X: 7.07975767594e-06 [m/s^3/sqrt(Hz)] [Y: 7.07975767594e-06 [m/s^3/sqrt(Hz)] [Z: 7.07975767594e-06 [m/s^3/sqrt(Hz)] ]
Scale Factor Std Dev [X: 3.00000000000e-03 [rad] [Y: 3.00000000000e-03 [rad] [Z: 3.00000000000e-03 [rad] ]
Scale Factor Random Walk [X: 1.00000000000e-09 [1/s/sqrt(Hz)] [Y: 1.00000000000e-09 [1/s/sqrt(Hz)] [Z: 1.00000000000e-09 [1/s/sqrt(Hz)] ]
Misalign Std Dev [X: 1.00000000000e-03 [rad] [Y: 1.00000000000e-03 [rad] [Z: 1.00000000000e-03 [rad] ]
Misalign Random Walk [X: 0.00000000000e+00 [rad/s/sqrt(Hz)] [Y: 0.00000000000e+00 [rad/s/sqrt(Hz)] [Z: 0.00000000000e+00 [rad/s/sqrt(Hz)] ]

```

Figure 17: Noise of the accelerometer provided by the iMar company

7.1.2. The GNSS device

The GNSS receiver used for the first test drive scenario is the multi-frequency OEM7700 from NovAtel. This receiver can track all current GNSS signals for civilian users. The observations from GNSS satellite constellations GPS, Galileo, Glonass, Beidou are processed. The OEM7700 receiver is enabled to offer the position solution with sub-meter to centimeter level accuracy. Table 5 gives an overview of the most important specifications of the GNSS receiver.

NovAtel OEM7700	
Position accuracy	
Single point L1-only	1.5 m RMS
Single point L1/L2	1.2 m RMS
NovAtel CORRECT DGPS	40 cm RMS
NovAtel CORRECT PPP	from 40 to 4 cm RMS
NovAtel CORRECT RTK	1 cm + 1 ppm RMS
Other accuracies	
Velocity accuracy	0.03 m/s RMS
Time accuracy	20 ns RMS
Data Rate	up 100 Hz

Table 5: Specifications of the GNSS receiver OEM7700

7.1.3. The iMar software

The iXCOM command software (iXCOM-CMD) was developed by the iMar company, and is used to configure the iMAR's iNAT system. This software also supports monitoring of measurement results for all iNAT-based IMU/GNSS strapdown inertial measurement systems like product family type iNAT-xxx, the inertial measurement system iRailLoc-C and other systems. This software enables users to provide an accurate trajectory of the moving train calculated by a loosely coupled GNSS/IMU/Odometer integrated system. This trajectory calculated by means of the iMar software is defined as the reference solution to assess the quality of the TU software developed by the doctoral candidate.



Figure 18: Graphical interface of iXCOM-CMD (GUI)

The iXCOM command software requires also the information about the lever-arm vector between the sensors. A surveying company has measured the lever-arm vector between the GNSS antenna and IMU device as well as the distance between the odometer and IMU device. Figure 19 shows the lever-arm vector from the IMU device to the GNSS antenna. The finally applied lever-arm vector amounts to: [1.450 m, 2.510 m, 0.540 m].

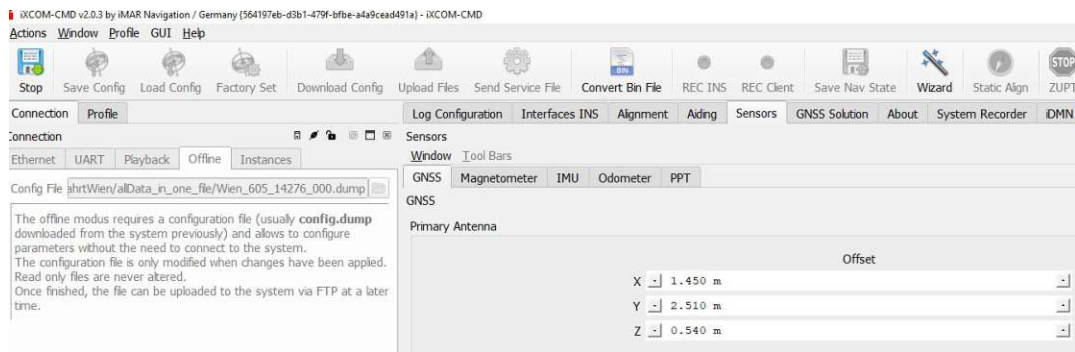


Figure 19: Lever-arm vector between the IMU device and the GNSS antenna

Figure 20 shows the lever-arm from the IMU device to the odometer computed by the software iXCOM-CMD, which amounts to: [-0.210 m, -1.080 m, 0.360 m].

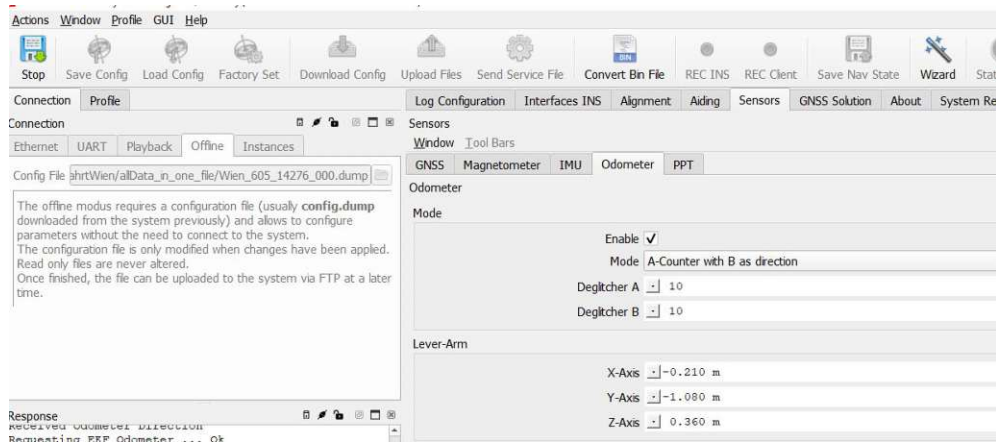


Figure 20: Lever-arm vector between the IMU device and the Odometer

The iMar software also provides information about the relationship between the IMU axes and the sensor directions mounted on the train. In this test case, the sensor coordinate frame has been already set by the iMar company and realizes a NED coordinate system. The x-axis of the IMU device is defined as the forward direction of the train, the y-axis as the right direction, and the z-axis as the down direction of the train's movement.

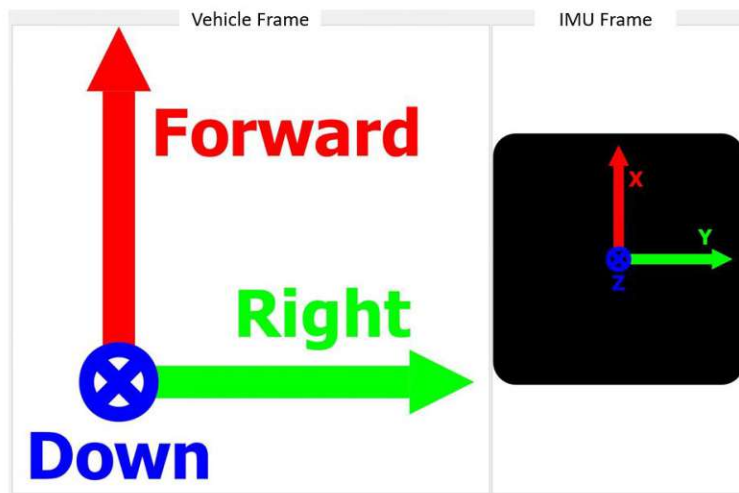


Figure 21: Relationship between the IMU axes and the sensor directions placed on the train

7.1.4. TU Filter software development

This test case aimed to develop a loosely coupled GNSS/IMU/ODO integration algorithm. This algorithm has been employed to determine the navigation solution of the moving vehicle. The algorithm was coded in Matlab (R2019a).

Figure 22 shows a flowchart of the overall process. The software uses the position derived by GNSS standalone signal processing and the velocity measured by the odometer as updates of the estimated position and velocity derived from the IMU data. Therefore, once the GNSS and odometer data has been acquired, the Kalman filter algorithm starts initializing at the first epoch. These data are merged with IMU data at every epoch in the iterative part. By applying the lever-arm vector between the GNSS antenna and the IMU device, as well as between the odometer and the IMU device the initial position and velocity of the IMU device is determined. Furthermore, the roll, pitch and yaw values at the first epoch from the iMar reference solution are defined as the initial alignment of the vehicle. An coarse alignment algorithm for calculating the initial alignment of the sensor was implemented in the software. However, the rest time of the train used for this initialization processing is short, which took about 5 minutes. The iterative part of the algorithm starts after reading the IMU measurements. The system model within the Kalman filter is described in equations 6.42 and 6.43. The measurement model comprises the difference between the position calculated by GNSS as well as velocity measured by the odometer and the IMU estimated position and velocity (see chapter 6.5). As soon as the GNSS signal is blocked, or rather to say that the GNSS solution provides an unsatisfying position quality, the system switches to a loosely coupled IMU/ODO integrated system immediately. The measurement model is then based on the difference between the velocity measured by the odometer and the IMU estimated velocity.

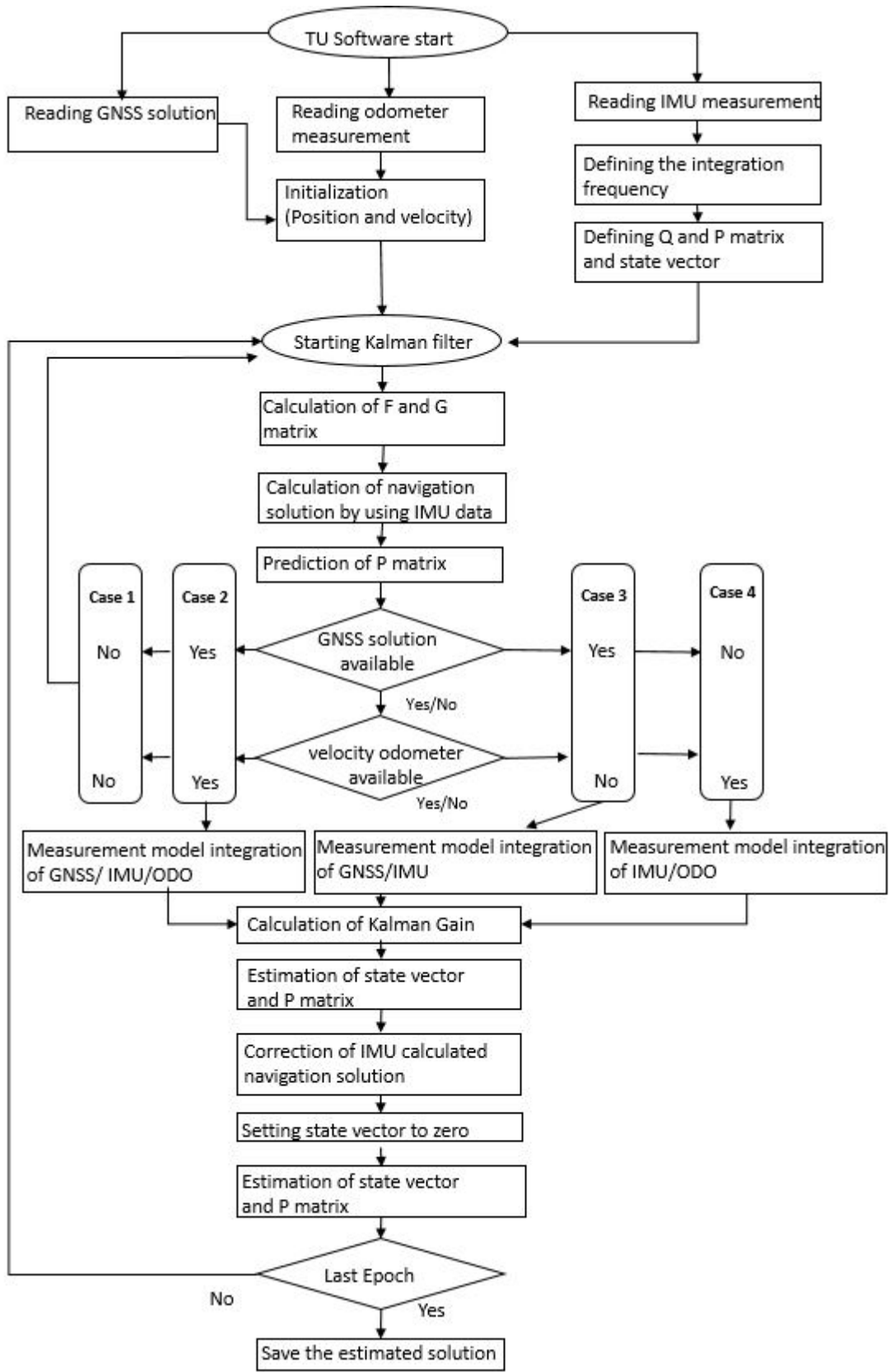


Figure 22: Flow chart of the TU developed loosely coupled GNSS/IMU/ODO integration algorithm

Within the loosely coupled processing, the measurement noise covariance matrix R determines the weight given to new measurements during the prediction step (see chapter 5.3). R is described by the equation 6.52. The larger the noise, the less the data will influence the calculation. The iMar company provided the GNSS navigation solutions (position and velocity) and the covariance of the GNSS solutions, these data are used to calculate the reference trajectory. The covariance of the GNSS position solutions were also applied to define the position components of the R matrix in the TU-Processing developed by the doctoral candidate. The unit of the measurement noise variance is m^2 . However, the noise of the odometer measurement is unknown. Nevertheless, different values of noise were tested, until the noise variance has been set to 0.08^2 (unit: $[m/s]^2$), which provides the most optimal result. The R matrix within the IMU/ODO system depends on the general accuracy of the measured odometer velocity:

$$R_{odo} = \begin{pmatrix} 0.08^2 & 0 & 0 \\ 0 & 0.08^2 & 0 \\ 0 & 0 & 0.08^2 \end{pmatrix}, \quad (7.1)$$

7.1.5. The IMU measurement data

The iMar software iXCOM-CMD saves IMU measurements in a CSV file. The IMU provides the measurement time as GPS week, GPS seconds of the week, sampling interval, date, GPS time and three-dimensional data measured by the accelerometers and the gyroscopes (see figure 23). The acceleration outputs are provided in $m\ s^{-2}$ and the angular rate is provided in $^\circ\ s^{-1}$.

E	F	G	H	I	J	K	L	M	N	O
GPS Week Number	GPS Seconds	Relative GPS Time [s]	GPS Date [Y-M-D]	GPS Time	Acceleration X [m/s/s]	Acceleration Y [m/s/s]	Acceleration Z [m/s/s]	Angular Rate X [deg/s]	Angular Rate Y [deg/s]	Angular Rate Z [deg/s]
2180	129410.5201	0	10/18/2021	11:56:50	-0.078231461	9.693691254	0.07790298	0.248591245	-1.01159061	-0.002544022
2180	129410.5221	0.002	10/18/2021	11:56:50	-0.029801099	9.809360504	-0.07832849	0.056055069	-0.766465366	-0.062608716
2180	129410.5241	0.004	10/18/2021	11:56:50	-0.152866006	9.78374958	-0.562804639	-0.116228489	0.286547086	-0.021810913
2180	129410.5261	0.006	10/18/2021	11:56:50	-0.018170152	9.688797951	-0.335240394	0.132809453	0.820977426	-0.184590918
2180	129410.5281	0.008	10/18/2021	11:56:50	0.214905679	9.803555489	0.048766188	0.189029314	0.509066752	-0.119918445
2180	129410.5301	0.01	10/18/2021	11:56:50	0.150316224	9.941650391	0.392942667	-0.109568407	-0.370087424	0.036709595
2180	129410.5321	0.012	10/18/2021	11:56:50	0.169339776	9.881893158	-0.155028075	0.00588226	-0.775146856	0.102194978
2180	129410.5341	0.014	10/18/2021	11:56:50	-0.157840848	9.743944168	-0.48977381	0.141787724	-0.532454707	-0.000325394
2180	129410.5361	0.016	10/18/2021	11:56:50	-0.055059019	9.710337639	-0.345278114	0.158787157	0.165596002	-0.091946793
2180	129410.5381	0.018	10/18/2021	11:56:50	-0.045192055	9.72886467	0.050596699	-0.078173449	0.622700503	0.087683107
2180	129410.5401	0.02	10/18/2021	11:56:50	0.260755956	9.859782219	0.132725537	0.01799965	0.44815139	0.04599991

Figure 23: IMU measurements of the first test drive performed by the train

When the train is stationary, the acceleration in the z-direction should measure the effect of the Earth's gravity $9.81\ m\ s^{-2}$, and the gyroscopes measure the values due to the impact of the Earth's rotation. The relationship between the IMU device coordinate system and the train body system was described in section 7.1.3. In general, the IMU provides its measurements in the body frame. As mentioned in sections 2.5, the b-frame of the train is defined by three axes pointing to the front (x-axis), right (y-axis) and down (z-axis) directions. However, the IMU

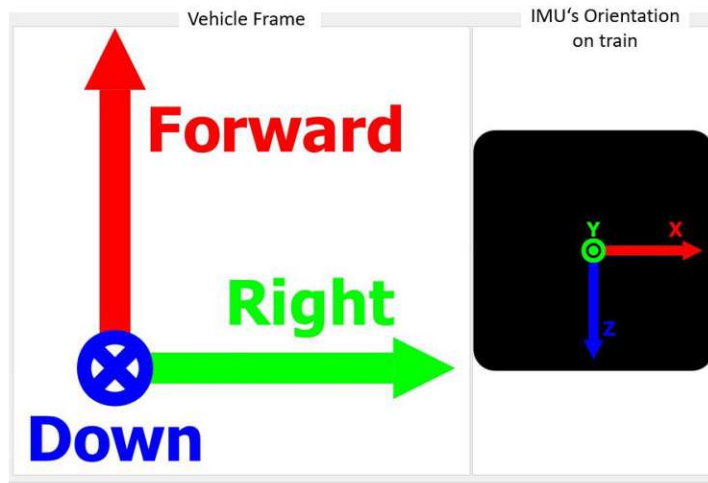


Figure 24: Relationship between sensor's regular orientation and its true orientation of the test scenario

First of all, a transformation matrix must be designed to convert the IMU measurements from the IMU device coordinate system $(-y, z, -x)$ to the train body system (x, y, z) . The transformation matrix between both systems can be formulated as follows:

$$C_{imu}^b = \begin{pmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}. \quad (7.2)$$

The test drive took approximately 32 minutes, with a driving distance of about 7.27 km on an flat rail track. Figure 25 shows the complete IMU measurement time serie during the test drive. In the beginning the train was stationary for about 5 minutes. After transformation between the IMU system and the body system, the measured acceleration in x and y directions and angular rate swing forth and back near zero, the acceleration in z-direction swings forth and back near -9.8 m s^{-2} . Later the train started to move for about 3 minutes, and then the train stopped again. After a pause of about 7 minutes, the train began to move again for 6 minutes. Then the train stopped again and started moving after 5 minutes. The train finished the test drive after about 5 minutes drive, and the data logger on board stopped logging and shut off.

It is striking that the angular rate values in z-direction are significantly large compared to the values along the other coordinate axes, even though the train is motionless (e.g. the first 400 seconds). The gyroscope output in the z-direction varies between 2 and -2 °/s. However, the values in the x and y direction vary between 0.3 and -0.3 °/s. There are not much differences between the x- and y-direction. However, the variations of the magnitude in the z direction is about 7 times larger than other both directions. The gyroscope in the z direction provides the output measurements with the noise. The gravity acts in the direction opposite to the z-axis and this parameter generally is constant over a small region [KM15]. The magnetometer can cancel out the noise in an angular rate sensor and stabilise the gyroscope in the z-direction. and this magnetic sensor is not included in iRailLoc-C [iMa21]. Hence, the noise of the gyroscope output measurements in the z direction is not eliminated. A practical gimbal-lock problem could also occur in this case [KM15].

Furthermore, the vibration of the iRailLoc-C box, which is mounted on the train directly next to the diesel engine, plays a big role as additional noise. Indeed, the yaw angle predicted by the algorithm may get larger continuously over time, even when the train is at rest. Therefore, the noise, which is contained in the measured angular rates, could make the estimation of the train's yaw angle worse.

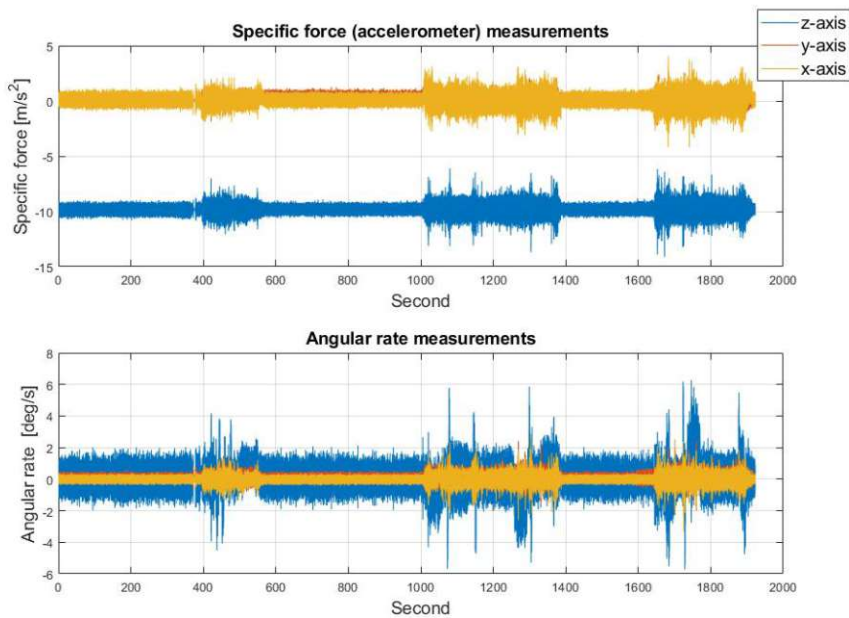


Figure 25: Figure on the top presents the measured accelerations, the bottom plot presents the measured angular rates

7.1.6. Track of the train

The data was recorded on October, 18th, 2021, starting at 11:56h (UTC). Firstly, the estimated yaw angle is analysed to match this with the route traveled by train (see figure 26). The starting point and the end of the test drive is visible, where the train stopped on a rail track at the railway station Kaiserebersdorf. Later the train drove along the rail track in the south-west direction.

A first short drive took about 3 minutes, afterwards the train has stopped. At 12:13 the train started again moving in north-east direction. The train passed by a bridge and at 12:19, the GNSS signal was very weak in a gap zone around the bridge and the quality of the GNSS solution was bad during 30 seconds. After passing the bridge, the train stopped again and moved back in the south-west direction to the starting point. After 5 minutes drive at 12:29, the train stopped at the same place as it started at the railway station Kaiserebersdorf. On the way back to the railway station, the train crossed the same bridge again at 12:24 and is the gap zone of the GNSS coverage for 45 seconds. At about 12:29 the sensors were turned off and the data logging stopped.

Figure 26 shows the two-dimensional horizontal trajectories of the vehicle estimated by both the reference software and TU-Wien solution. The trajectories were represented in Open street map. The blue line denotes the estimated horizontal coordinates employing the iMar software, and the red line represents the estimated vehicle trajectory using the TU-Wien solution. However, as both lines overlap all the time, the coordinate differences are shown more clearly in figure 29.

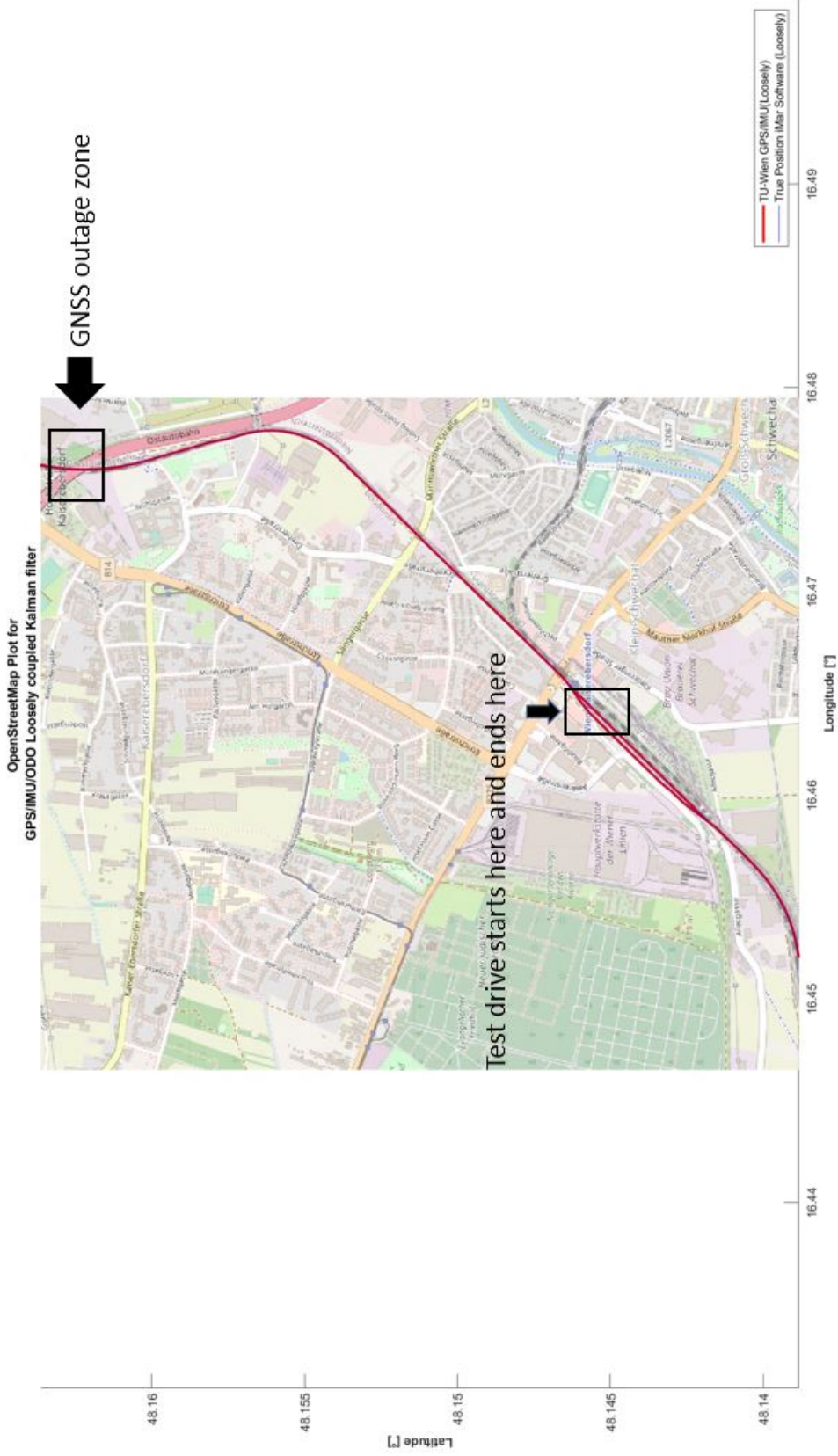


Figure 26: Train's trajectory

7.1.7. The results

In the following coordinates, velocity and attitude angles calculated by the reference software and the algorithm established in this thesis are compared. The navigation solution and associated parameters, including the odometer scale factor and the inertial sensor biases, are estimated (16-state variables). Here the chosen approach is consistent with the approach used by the iMar reference software. The results in figures 26, 28, 31 and 33 show that both software packages provide similar navigation solutions along the test drive.

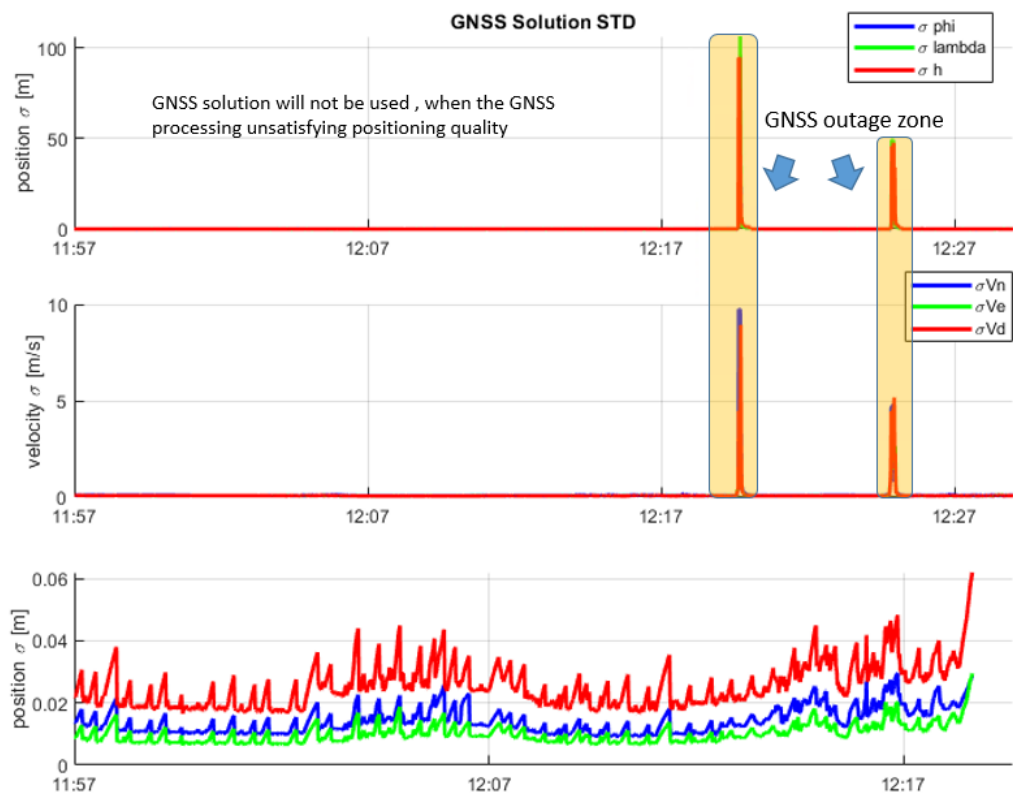


Figure 27: Standard deviation of GNSS processing. Upper one shows the position error, and the middle one shows velocity error, the bottom one shows the positioning quality before the GNSS outage

Figure 27 is the plot of the three-dimensional standard deviations σ of positioning and velocity solutions, which are calculated by GNSS signal processing. The values of standard deviations on the graphic are significantly small except two periods. Before the train moved under the bridge. The GNSS solution quality of σ_{phi} and σ_{lambda} are not exactly zero, but they are close to 0.01 meter. However, the height values vary between 0.02 and 0.04 meter.

Later, it can be clearly seen that there are two big jumps. One of them reaches to 100 meter position error and 10 m/s velocity error, and another one reaches to about 50 meter error in the positioning solution and 5 m/s error in the velocity solution. When the train travels under a bridge, the GNSS signals are blocked and the parameter solution is harmed, causing large standard deviations. Therefore, the GNSS data should be omitted during these periods, and the threshold values have to be determined to assess the GNSS positioning solution is good or bad. For this analysis, the box plot method is used to determine the lower bound and the upper bound for a bad quality of the GNSS solution. Table 6 represents the results, and [0.0265 0.0183 0.0461] are defined as the threshold values in φ , λ and h . These values indicate the GNSS unsatisfying positioning quality, when the algorithm recognizes that the standard deviations from the GNSS processing are worse than these values, the GNSS/IMU/ODO system will change to the IMU/ODO system.

bound	φ [m]	λ [m]	h [m]
lower bound	0.0016	0.0018	0.0062
upper bound	0.0265	0.0183	0.0461

Table 6: The determined bound for a bad GNSS solution (standard deviation)

The first time the train travelled under the bridge (12:19), the GNSS signals are lost about 30 seconds. The second time of loss of the GNSS signals is about 30 seconds (12:34). Figure 28 shows the three-dimensional positioning solution. The red line shows the solution calculated by reference software, and the blue line represents the trajectory from the TU-Wien software. The unit of the horizontal positioning solution are represented in degrees, the differences between the solutions calculated by both softwares is detailed in figure 29.

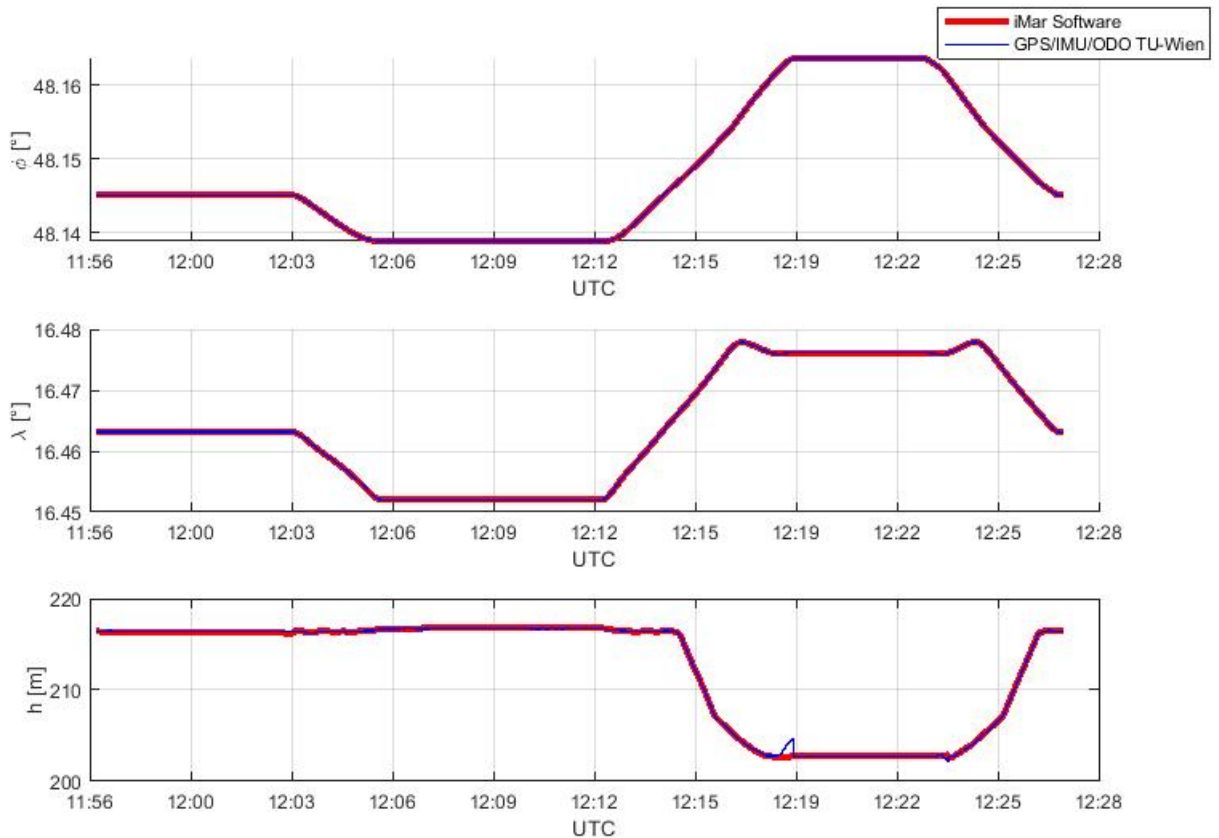


Figure 28: 3D positioning solutions processed by TU-Wien Software and reference software (Both solutions are estimated by using the loosely coupled approach)

Figures 29 and 30 displays the three-dimensional position differences between the TU-Wien software solution compared to the iMar software solution. This allows to assess the quality of TU-Wien software. The graphic shows that the differences in North, East and Height components are relatively small. The deviation is within a range of a few decimeters up to 3 m when bridging tunnels. Interestingly the graphic indicates that the differences increases in East and Height components, when the train does not move (up to 2 dm in both direction). However, when the train is moving, the differences are stable (see figure 30). As already mentioned, the angular rate values in the z-direction are significantly larger compared with the angular rates along the horizontal axes. Different noises impact strongly the estimation of the train's trajectory. The yaw angle with noise can cause small errors of a few cm in position [Bar96]. When the train is in motion, the deviations between both software solutions of course diminish and the results are more stable.

Moreover, only the IMU and odometer measurements are available during two GNSS

signal outages, the odometer data was applied to limit the IMU sensor errors. So the regular GNSS/IMU/ODO system data processing switched to a IMU/ODO processing. During these periods, the positioning accuracy reaches up to 2 meters in horizontal direction, up to 1.5 meters in vertical direction.

In summary, the differences between the TU-Wien software and the iMar are relatively slight except when the GNSS signal failed. The deviations were generally at the cm-level only. But figure 29 pretty well shows, when the GNSS signal failed, the system IMU/ODO decreases the positioning accuracy.

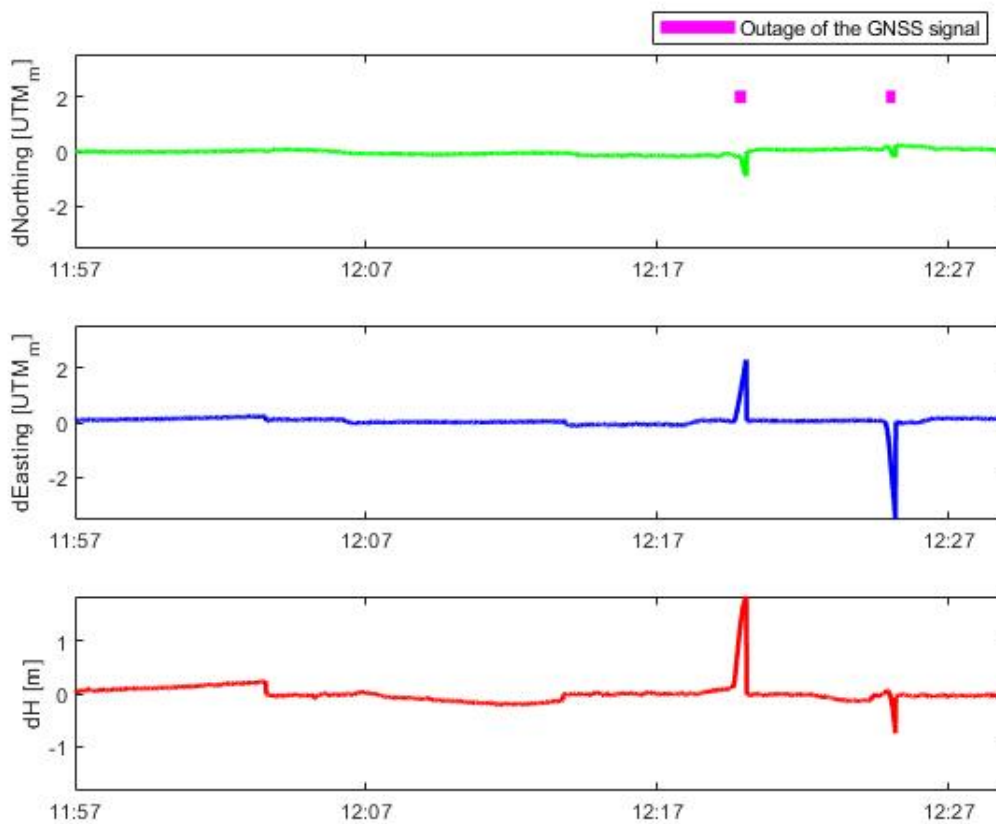


Figure 29: Positioning differences between solutions processed by iMar Software and TU-Wien software

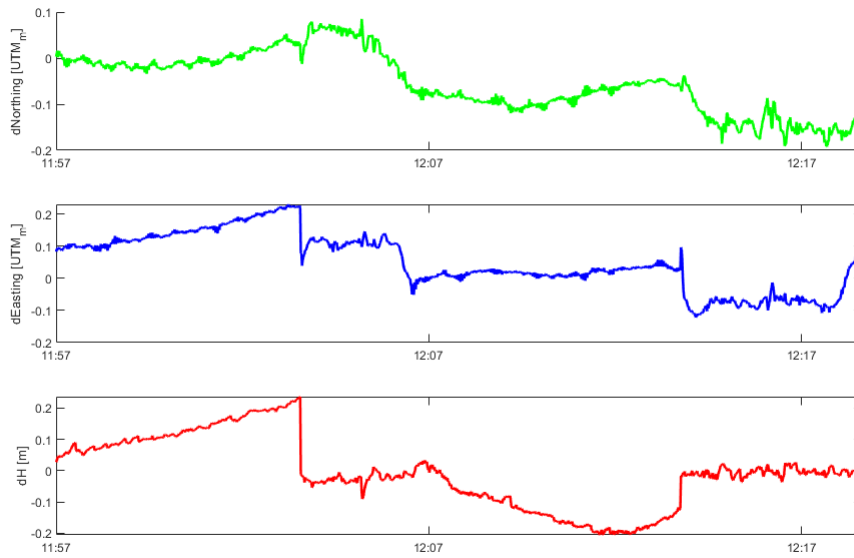


Figure 30: Positioning differences between both solutions before the GNSS outage

Figure 31 shows the three-dimensional velocity solutions of both softwares in North, East and Down directions. The estimated velocities are reported in meter per second (m/s). The blue line represents the solution processed by TU-developed algorithm, and the red line shows the solution processed by iMar software. It is clearly visible that the differences in all directions are minimal. The train always moves on a flat railway track, so the train's velocity in the vertical direction should be close to zero all the time.

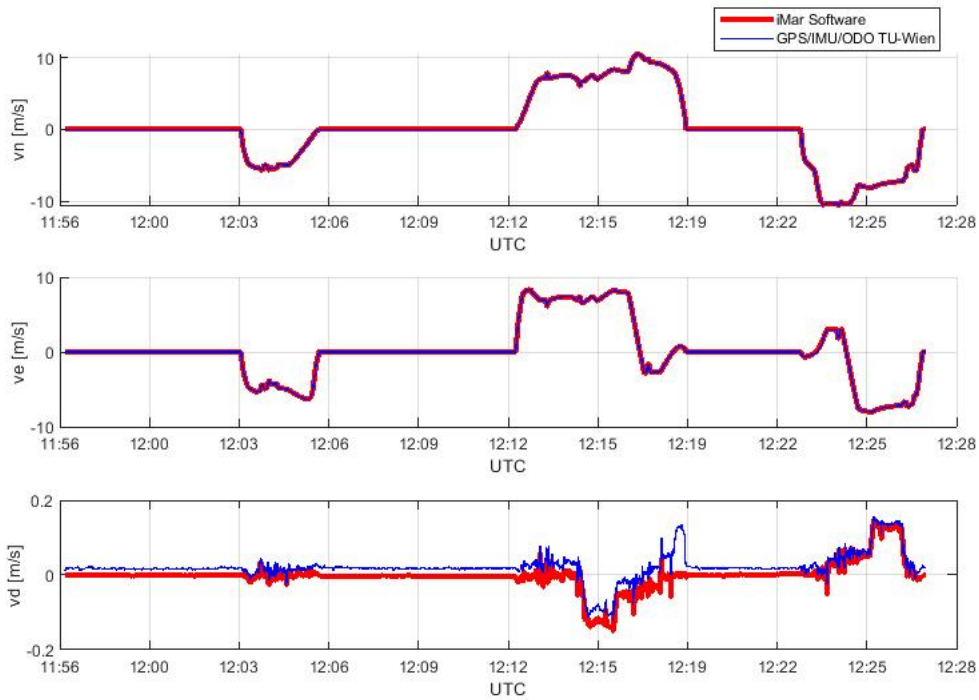


Figure 31: 3D velocity solutions processed by TU-Wien software and reference software

Concerning the velocity components, the TU-Wien solution delivers also a stable solution as expected except when the GNSS signal is not available. Figure 32 shows three-dimensional velocity differences between the TU-Wien solution and the reference software. It can be seen clearly that the deviations in the North direction are within a range of 0.1 m/s, in East direction up to 0.3 m/s and in Down direction up to 0.15 m/s, when the GNSS signal is not available. When the GNSS/IMU/ODO combination is activated, the deviations are much smaller.

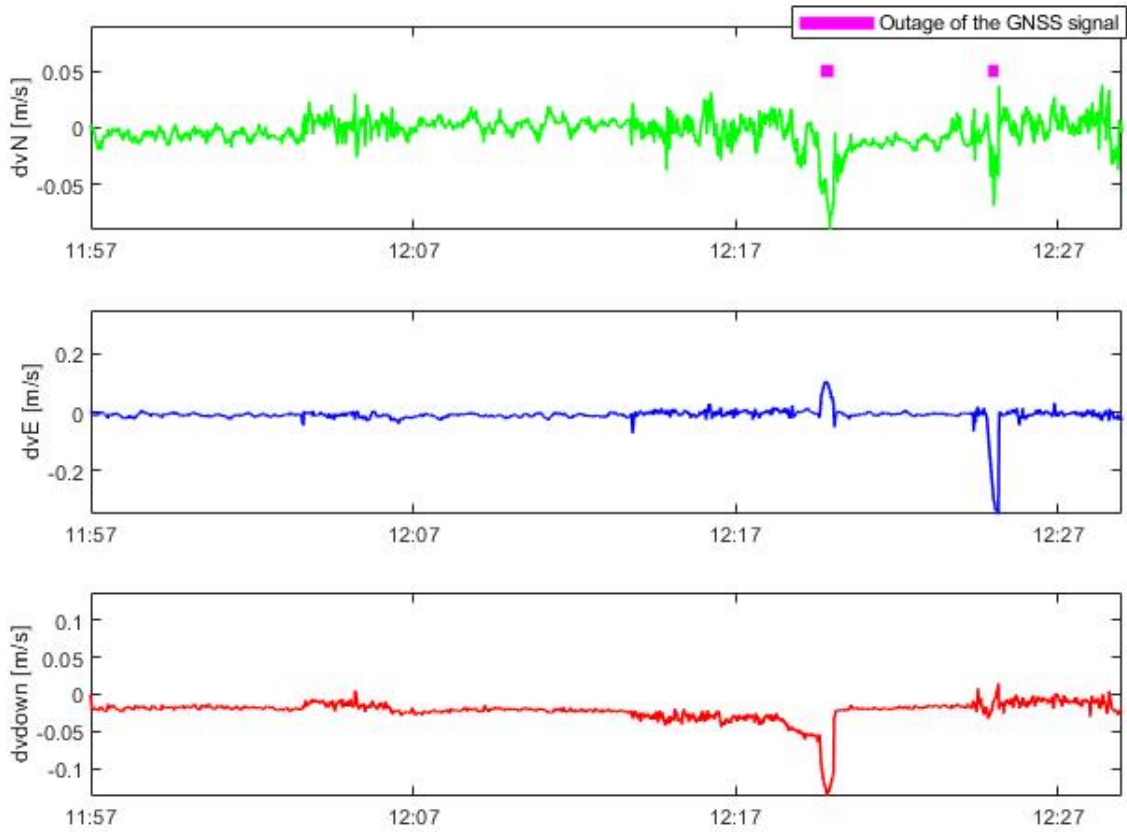


Figure 32: Velocity differences between solutions processed by iMar software and TU-Wien software

Figure 33 represents pitch, roll and yaw solutions from both softwares in degrees. Similar to figures 29 and 31, the differences between both solutions are small.



Figure 33: Roll/Pitch/Yaw solutions processed by TU-Wien Software and reference software

Figure 34 shows pitch, roll and yaw angles differences estimated by means of TU-Wien software and reference software. As mentioned before, the measured angular rates in the z-direction contain increased noise. While the train does not move, deviations in roll and pitch angles within 0.1 degrees show up, while as expected the yaw angle estimates deviate within a range of 2 degrees over time (e.g. the first 400 second). It can be seen from the results that the IMU for stationary train gives high drift in yaw angle and stable values of roll and pitch angles. However, while the train is moving, the differences between both solutions are very small. While the integrated system changes to sole IMU/ODO mode, the deviations between both solutions are bigger. Two peaks in the graphic are clearly visible. The deviations in roll and pitch angles increase up to 0.2 degree, and it increases up to about 1.5 degree in the yaw angle.

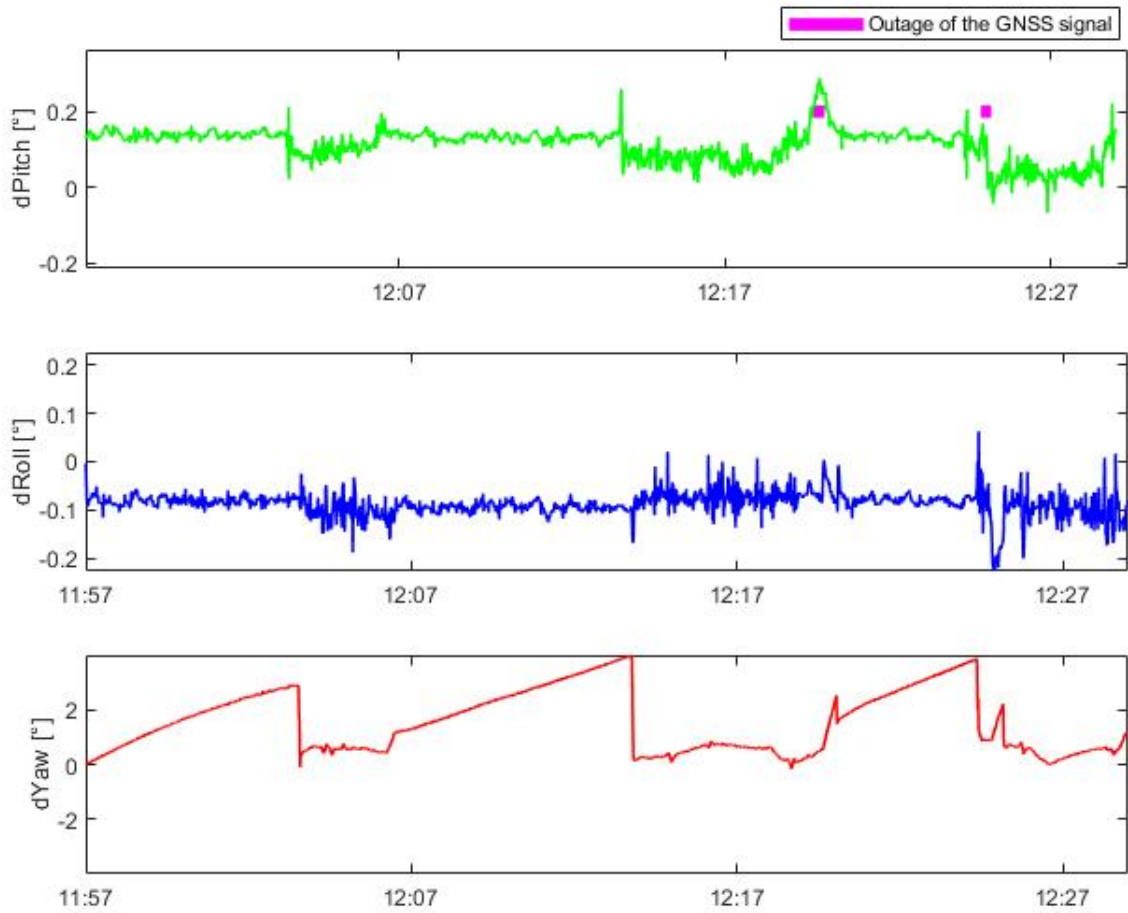


Figure 34: Pitch/Roll/Yaw differences processed by iMar software and TU-Wien software

Last, but not least. In this test case, the odometer scale factor error is also considered. Figure 35 shows the scale factor error differences estimated by means of TU-Wien software and reference software. The blue line denotes the estimated scale factor error employing the iMar software. The red line represents the estimated scale factor error using the TU-Wien software. The development of both time series is similar. Moreover, the deviation between both solutions is within 0.0001, which is quite good solution.

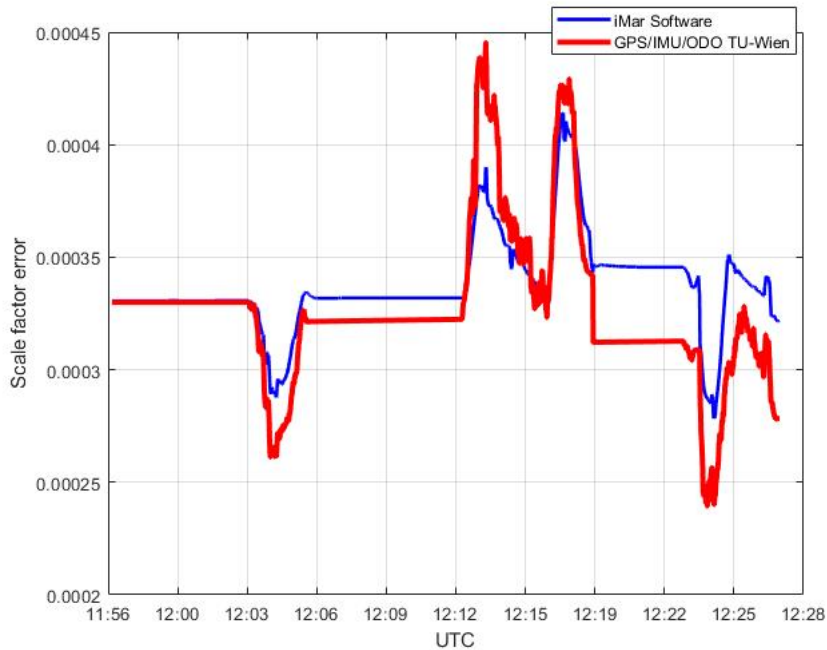


Figure 35: Odometer scale factor error estimated by iMar software and TU-Wien software

7.2. The 2nd test drive scenario

7.2.1. The IMU Device iNAV-RQH-003

The car and navigation sensors for the test drives were provided by the working group navigation of the institute of Geodesy TU Graz. The IMU device used for the second test drive is a strapdown high accurate navigation-grade device from the iMar family type iNAV-RQH-003 (see figure 36).



Figure 36: IMU device type iNAV-RQH-003 [Wie]

The device carries three servo-accelerometers type QA2000-40 (Q-Flex) and three

high precision ring laser gyroscopes type GG1320. This device is characterized by improved performance parameters, such as bias and scale factor stability, linearity, and acceleration sensitivity. Table 7 gives an overview of the most important specifications of this IMU device.

iNAV-RQH-003	
Gyroscope performance	
Random walk	$< 0.0025^\circ/\sqrt{h}$
Drift	$< 0.003^\circ/\text{hr}$
Accelerometer performance	
Random walk	$< 8 \mu g\sqrt{Hz}$
Bias	$< 100 \mu g$
Data Rate	2000 Hz
Price of the sensor	160.000 Euro

Table 7: The specification of the device iNAV-RQH-003

The IMU device is mounted on the tailored calibrated platform of a car's roof. At the same time, the GNSS antenna and receiver has been mounted at this platform for the simultaneous measurements (see figure 37 and figure 38). The car's moving direction is as shown in figure 38 (the red arrow), and "Fahrriichtung" can be translated as "driving direction" in German. The IMU device output has been synchronized to GPS-time. If the IMU and the GNSS receiver are accurately time-synchronized, the IMU device measures data calibrated via the GNSS receiver's external PPS signal (pulse per second). The data rate of the IMU observations was set to 2000 Hz for the test drives.



Figure 37: GNSS and IMU sensors mounted on the car's roof for a test drive of TU Graz in the past [Lae10]

The lever-arm vector between the GNSS antenna and IMU device was measured by the working group within a bachelor thesis. The finally applied lever-arm between the IMU device and the GNSS antenna amounts to: $[-0.001 \text{ m}, 1.097 \text{ m}, 0.153 \text{ m}]$.

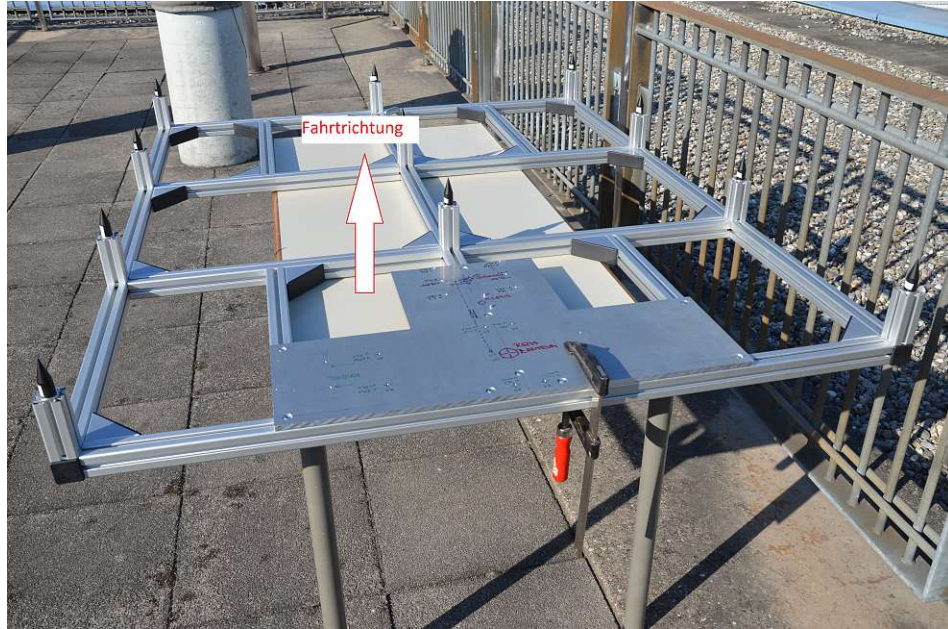


Figure 38: Tailored calibrated platform applied for the sensor fusion [Wie]

For the test drives performed by the TU Graz, the y-axis of the IMU device was defined as the forward direction, and the z-axis as the up direction of the vehicle's movement. The x-axis turning to the right completes a positive (right-handed) 3D Cartesian system.

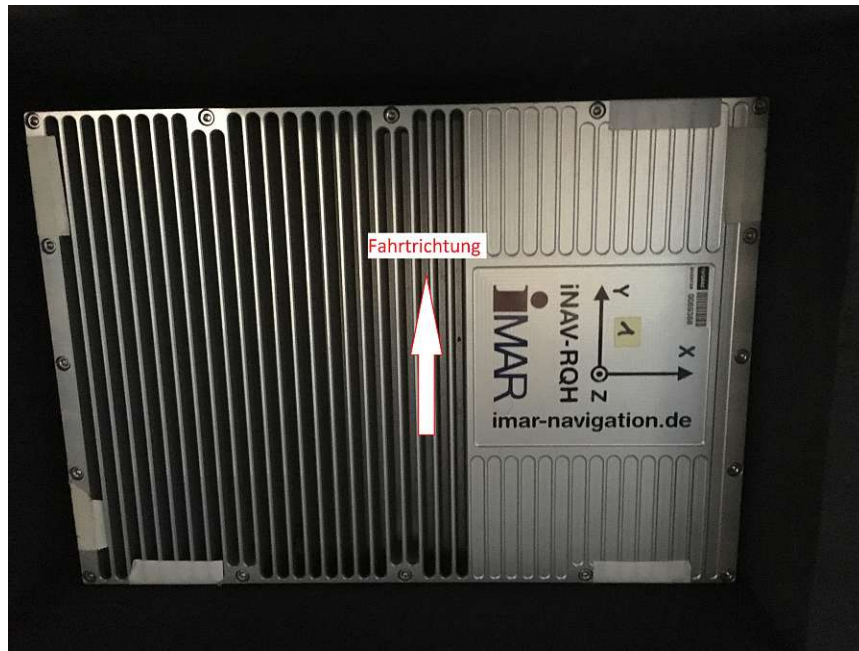


Figure 39: Relationship between the IMU axes and the vehicle axes [TU Graz,2019]

7.2.2. Coordinate system of the applied IMU device

In this test case, the iMar's IMU device is specially used for a land vehicle. The n-frame of the final navigation solution was defined as a NED-frame. Using NED simplifies the calculation in the algorithm due to its positive direction in the clockwise direction. The NED axes (North, East, Down or rather in x,y,z) coincide with the body fixed coordinates (roll, pitch, yaw), while the vehicle is leveled and heading North [GWA06].

The IMU's output measurements in the b-frame can be preconfigured that respect to the orientations of the coordinate axes of the n-frame and the vehicle's frame (see chapter 2.5). In this test scenario, the obtained IMU measurement data were preconfigured, which refers to the ENU reference system. Thus, a coordinate transformation has to be carried out to transform the measured IMU data to the NED. The IMU data rotates from the ENU so that the rotated system can coincide with the NED [Gra07]. As assumed, the ENU rotates in roll direction about π , in yaw direction about $\frac{-\pi}{2}$ and no rotation in pitch direction (see Figure below).

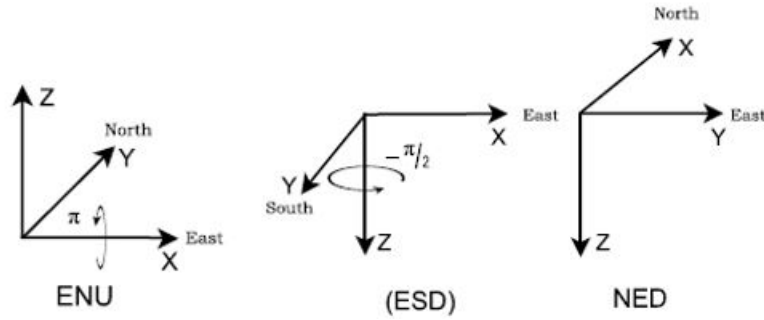


Figure 40: Transformation from the ENU system to the NED system carries two rotations in Roll and Yaw directions (from the left side to the right side) [Gra07]

The transformation matrix between both systems can be formulated as follows [GWA06]:

$$C_{NED}^{ENU} = C_{ENU}^{NED} = (R_{yaw}R_{yaw})R_{roll} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \quad (7.3)$$

Now, the sensed accelerations and angular rates in the NED system can be obtained by using $f_{ib\text{ENU}}^b C_{NED}^{ENU}$ and $w_{ib\text{ENU}}^b C_{NED}^{ENU}$.

7.2.3. The GNSS device

The GNSS data sets used as the rover for the test drive were recorded by a high-quality geodetic dual-frequency GPS/GLONASS receiver from Javad type Sigma TRE-G3TH with a Javad GrAnt-G3T antenna. This receiver is capable of receiving L1, L2, and L5 GPS signals [Lae10]. It can transform and output the raw observation data (pseudo-range, carrier phase, Doppler measurement and ephemerides data) in RINEX format. Javad GrAnt-G3TH is a versatile high-performance antenna that can track GPS, Glonass, Galileo, BeiDou, WAAS, EGNOS, MSAS, GAGAN and QZSS signals.



Figure 41: Javad GrAnt-G3TH antenna(left) and Javad Sigma TRE-G3TH receiver(right) [Glo]

The observations of the JAVAD receiver were recorded on 19.05.2019, starting at 10:12. The logging data rate of the GNSS receiver is set to 5 Hz during the test drive. In this research, the GPS code and carrier-phase observation (C1, C2, P1, P2, L1 and L2) and the Doppler observations (D1 and D2) are used.

7.2.4. The Waypoint software

Inertial Explorer from the Novatel's waypoint products Group is a post-processing software supported DD, PPP based tightly coupled processing for integration of GNSS and IMU measurement dataset [Exp20]. In this test case, a reference solution processed by the Inertial Explorer software is used to check the quality of the TU software's solution. Therefore, the same measurement data, noise values and processing method are used to obtain the reference trajectories. A tightly-coupled approach with DD GPS/Glonass observation was chosen to achieve the reference solution of centimeter accuracy (see figure 42).

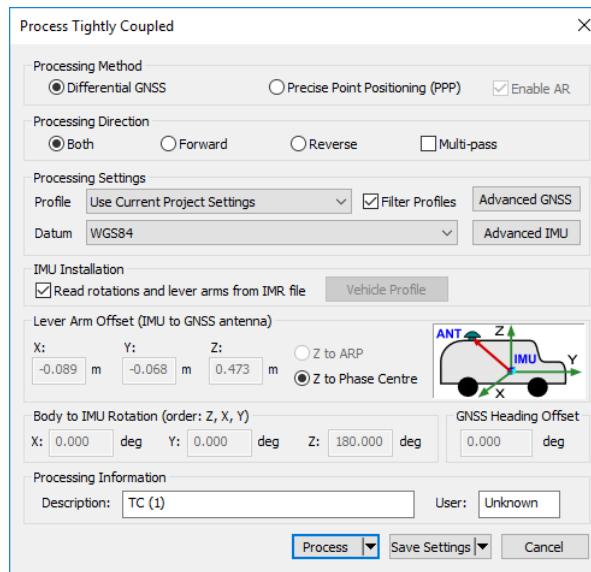


Figure 42: Processing Settings [TU Graz, 2020]

The GNSS observation processing was set to the option "Dual-frequency carrier phase", which means that the software activates the Ionosphere-free linear combination for eliminating the first order (up to 99.9%) ionospheric effect (see figure 42). This option should provide the best results if both reference station and rover measurement database on dual frequency [Exp20]. The elevation mask was set to 5 degrees, thus, the software removed all observations below this elevation.

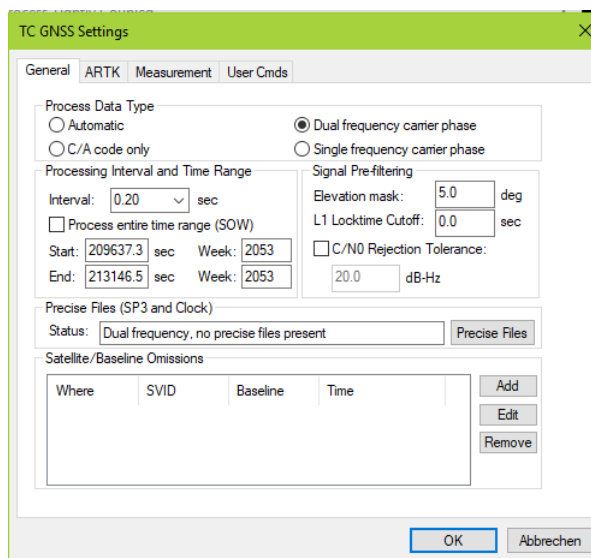


Figure 43: GNSS observation processing Settings [TU Graz, 2020]

Standard deviations have to be entered for the code, carrier and doppler measurements

(see figure 44). These values are required for an appropriate weighting of the GNSS DD pseudo-range, carrier phase and pseudo-range rate measurements. The reference software assigns the relatively high measurement standard deviations of 7m to the code ranges, 0.02 m to phase ranges and 1 m/s to doppler measurement applied for the DD processing. Setting these values is necessary for generating the measurements noise covariance matrix R . If ambiguities are not fixed, the GPS measurement STD weighting can affect the float solution convergence [Exp20].

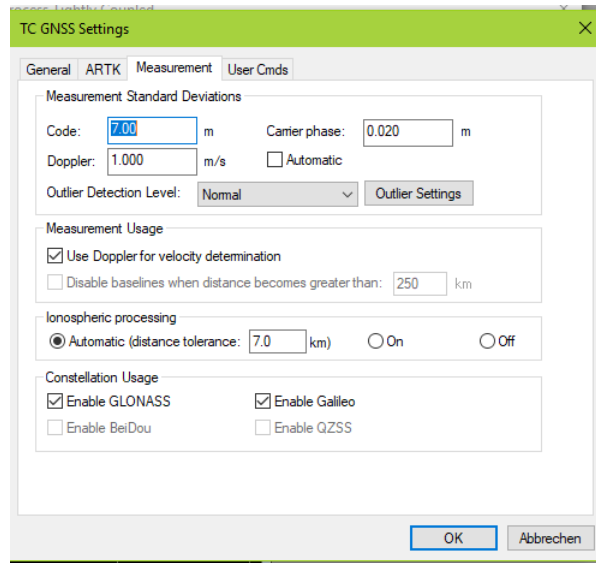


Figure 44: GNSS measurement noise Settings [TU Graz, 2020]

Also quantities to model the IMU sensor have to be entered (see figure 45). "Initial Standard Deviation Values" represents the initial acceleration biases of the accelerometers and initial angular rates. The "Spectral Densities Values" comprise the uncertainty components for calculating the system noise covariance matrix in the Kalman filter. ARW is the abbreviation for angular random walk, VRW stands for velocity random walk. Reasonable numbers for sensor uncertainties can be found in the specification documents published by the IMU device manufacturer.

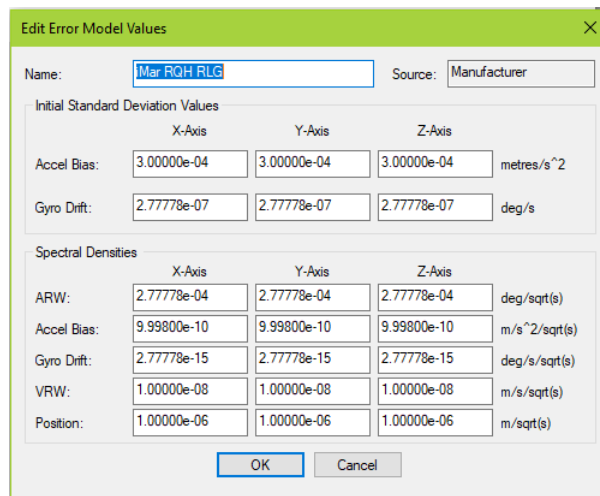


Figure 45: Setting "Error Model Values" [TU Graz, 2020]

7.2.5. The TU developed tightly coupled GPS/IMU algorithm

As already mentioned in previous chapter, a tightly coupled Kalman filter algorithm based on the fusion of GPS and IMU data was developed within this thesis. The algorithm has been employed to estimate the trajectories of slow and fast-moving vehicles like cars or trains in post-processing. The developed algorithm has been coded in Matlab language (R2019a). The software supports input of GPS raw data in Rinex format and IMU measurements in txt format.

Figure 46 shows the overall process of the developed software to determine the trajectory of a moving vehicle. The software allows different GPS observation processing strategies based on the SPP with code-only processing, DD with code-only processing and combined code plus carrier phase processing. In SPP mode only the observation file of the rover is required for the processing. The SPP algorithm can determine the receiver clock offset and zenith wet delay. Still, satellite orbit (sp-3) and clock (clk) files have to be manually downloaded, provided by International GNSS Service. At least data of one reference station has been added when the DD processing mode is activated. Also navigation files of the rover and the reference station have to be available.

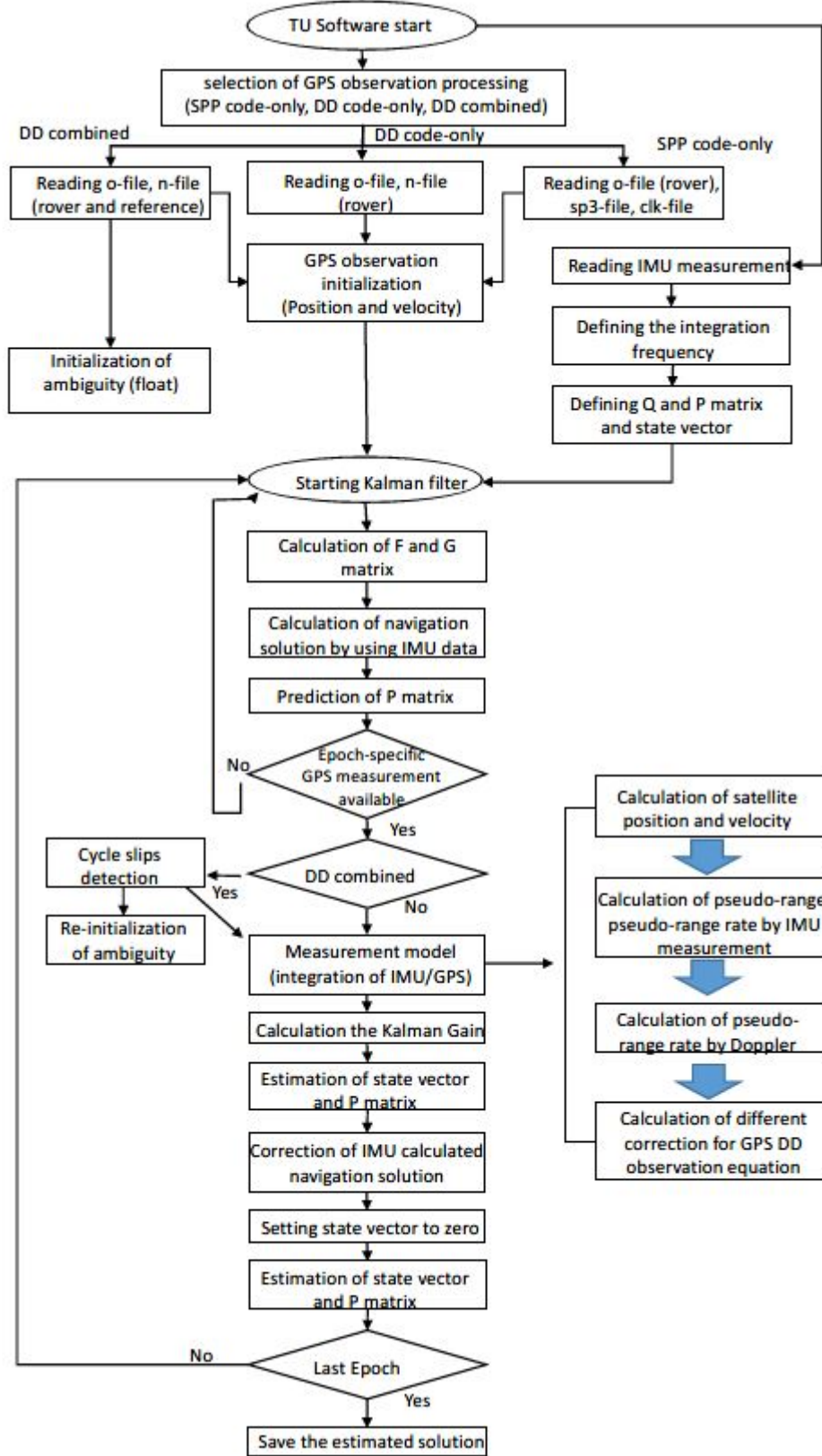


Figure 46: Flow chart of the TU developed tightly coupled GPS/IMU integration algorithm

There are two processing options implemented for the DD observation approach. One is with code-only processing, and the other is combined code plus carrier phase processing. Once raw GPS data of the rover and the reference station have been acquired, the Kalman filter algorithm has to be initialized. After initialization the algorithm operates with the IMU measurement data of each epoch in the iterative processing.

For the initialization, satellites with an elevation angle above 5° visible for both the rover and the reference station are selected. To speed up convergence of the Kalman filter later on, the initial position and velocity of the rover, initial float ambiguity values and their error covariance are estimated using an least-squares adjustment by processing GPS DD code plus phase data over a few epochs. (see chapter 6.6). Similar to the first test scenario, the algorithm takes at the first epoch the initial alignment from the reference solution of the vehicle. The calculated float solution does not solve for integer ambiguities. When the vehicle moves, the number of visible satellites is also changing frequently. The ambiguities must be reinitialized very often. On the other hand, the float solution deteriorates the accuracy of the estimated solution to dm-level instead of cm-level. In a dynamical scenario, the float solution is often the best alternative, although the fixed solution could provide more accurate results.

The iterative part of the GPS/IMU integration algorithm starts after reading the IMU measurements. The IMU data contains the accelerations and angular rate observations in the x, y and z directions recorded in GPS-time. By applying the lever-arm vector between the GPS antenna and the IMU device also the initial position of the IMU device is determined.

The system model within the Kalman filter is described in chapter 6.4. The measurement model within the tightly coupled Kalman filter makes use of the difference between the GPS measured DD observation and the IMU estimated DD observation (see chapter 6.6.1). The 3D coordinates of the EPOSA reference station in Graz is calculated by using the high accuracy PPP software raPPPid (developed at the TU Vienna). The developed tightly Kalman filter directly uses the difference between the measured GPS raw observations and corresponding 'pseudo-observations' derived from the IMU data. These differences are fed back into the navigation solution of the IMU and allow correcting also for sensor biases. Therefore, these state errors from the IMU measurements can be eliminated, and the IMU calculated navigation solution becomes more accurate. The algorithm calculates the estimated error states for each epoch, and the previous estimates will be the new input for the current states.

Also further processing settings are required. Table 8 shows an overview of the parameter settings within the TU software to process data of the test drive. The software uses the ionosphere-free combination to process dual-frequency data to eliminate the ionospheric

delay and applies the Saastamoinen model to account for the tropospheric delay. The software also considers error corrections in single point mode (PPP-code) such as phase wind-up, solid earth tides, relativistic corrections, etc. When the code-only observation processing option (SPP or DD) is chosen, the processing algorithm is simpler and faster than the combined DD processing. Still, the calculated solution is limited to sub-meter accuracy.

Parameter	Settings
Stations	Rover, rover + the reference station
Device	GPS + IMU
GPS observation	PPP code-only, DD code-only & combined code+phase
Processing mode	Post-processing, kinematic
Data Rate	IMU: 2000 Hz & GPS: 5 Hz
Integration frequency	200 Hz
Observation weighting model	Satellite elevation ($\sin(el)^2$)
STD GPS raw observation	Code: 20 cm, phase: 0.8 cm & doppler: 20 cm
Cut-off angle	Elevation: 5°
Satellite orbit and clock correction	GFZ final product for the PPP processing
Satellite and receiver antenna	IGS Antex igs14.atx
Troposphere model	Saastamoinen model
Ionosphere model	Ionosphere-free linear combination (L1, L2)
Phase ambiguity	Float solution
Cycle slip detection	Doppler measurement based
Other correction model	Phase wind-up, solid earth tides, relativistic correction
Min.number of satellites	2

Table 8: General processing settings used for the tightly coupled GPS/IMU integration

7.2.6. The IMU measurement data

The IMU device records the measurements in a binary file. To enter the data in the established software, converting the IMU binary format into a txt format is necessary. The converted txt file normally provides the measurement logging time in GPS seconds of the week as well as the accelerometer and the gyroscope measurements (see figure 47). The acceleration output is provided in m s^{-2} . As the vehicle usually moves on an almost flat street, the accelerations in horizontal directions swing forth and back near zero plus the additional small deviations caused by uneven ground or noise due to the motor vibration. The accelerations in the z-axis are

overlaid by the local gravity. The gravity of this region is 980.69 Gal. This value was calculated by using the software Topograv, which is developed by the research unit Higher geodesy TU Wien. This software can calculate the terrain effects and synthetic gravity functionals. The angular rate is provided in $^{\circ}\text{s}^{-1}$, the output swings forth and back near zero. When the car is motionless, the acceleration in z-direction should measure the positive values of the effect of the Earth's gravity 9.81 m s^{-2} , and the gyroscopes measure the values due to the effect of the Earth's rotation. The IMU iNAV-RQH-003 prefigured the ENU coordinate system as its output coordinate system (see chapter 7.2.2). As the calculation within the TU algorithm is based on a NED coordinate system. The IMU outputs have to be transformed to the NED coordinate system using equation 7.3.

Time [SOW]	Gyro_X [deg/s]	Gyro_Y [deg/s]	Gyro_Z [deg/s]	Accel_X [m/s ²]	Accel_Y [m/s ²]	Accel_Z [m/s ²]
209635.907721	0.00144	0.61843	-0.61882	0.20447	0.76023	10.42594
209635.908244	0.00104	1.23680	-0.61927	0.11434	-0.08673	9.64447
209635.908727	1.23713	-0.61691	0.00408	0.21770	-0.56307	9.27427
209635.909250	-1.23593	-1.23820	-0.00272	0.26822	0.43759	10.11865
209635.909732	0.00000	0.00000	0.00000	0.16566	0.46192	10.47003
209635.910255	1.23633	0.61982	0.00317	0.17032	-0.54239	9.36123

Figure 47: An example of the applied IMU's measurement data saved in a txt file

The test drive took approximately 56 minutes, with a driving distance of about 15 km. Figure 47 shows the complete IMU measurement time series during the test drive. The vehicle was stationary and kept a record of sensor data for at least 10 minutes in the beginning. The measured accelerations in x and y directions are back near zero, the acceleration in z-direction swings forth and back near 9.81 m s^{-2} . This stationary period is necessary for determining the initial alignments of the IMU device. Later the car started to move for about 2 minutes, and turned back to a static position. After a pause of about 15 minutes, the car started to move again for 20 minutes. At the end of the test drive, the vehicle stopped and was static for 10 minutes.

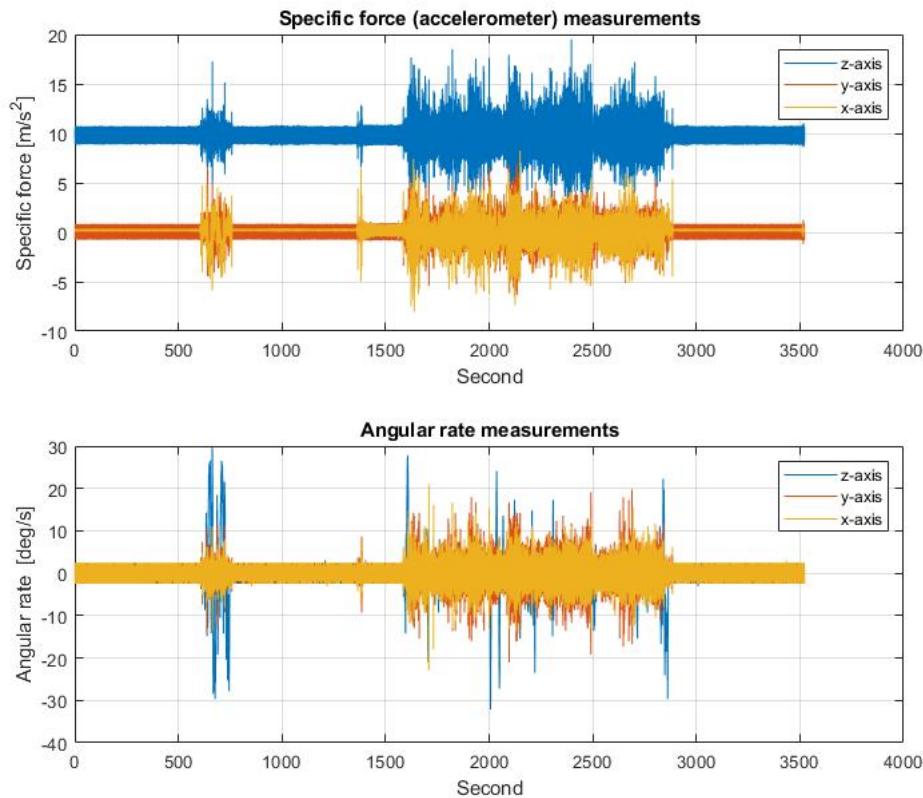


Figure 48: Measured acceleration time series (top), measured angular rate time series (bottom)

7.2.7. Track of the car

The travelled track of the car is shown in Figure 48. Data was recorded on May, 19th, 2019, starting at 10:13h (GPS time). Firstly, the estimated yaw angles are analyzed to match these with the route traveled by car.

The starting point and the end of the test drive is visible on the left side of Figure 49, where the car stopped on a parking lot for at least 10 minutes. The vehicle was placed at a location about 11km away from the reference station in Graz. Later, the car drove along the roundabout and finally getting back to the parking lot.

As noted in the previous chapter, the car started again moving in the north direction to the Graz airport at 10:42. After passing the airport the vehicle moved back in the south direction to the starting point. Thus the test drive was a 25 minutes ride around the Graz airport. At about 11:10 (UTC) the sensors were turned off and the data logging stopped. Figure 49 shows the two-dimensional horizontal trajectories of the vehicle estimated by both the reference

software and TU software. The trajectories were represented in Open street map. The blue line denotes the estimated horizontal coordinates employing the Waypoint software, and the red line represents the estimated vehicle trajectory using the TU software. The differences in position between both software solutions are just in the few dm (2dm) range, both lines overlap all the time. Coordinate differences are shown more clearly later on in figure 54.

The route of the test drive

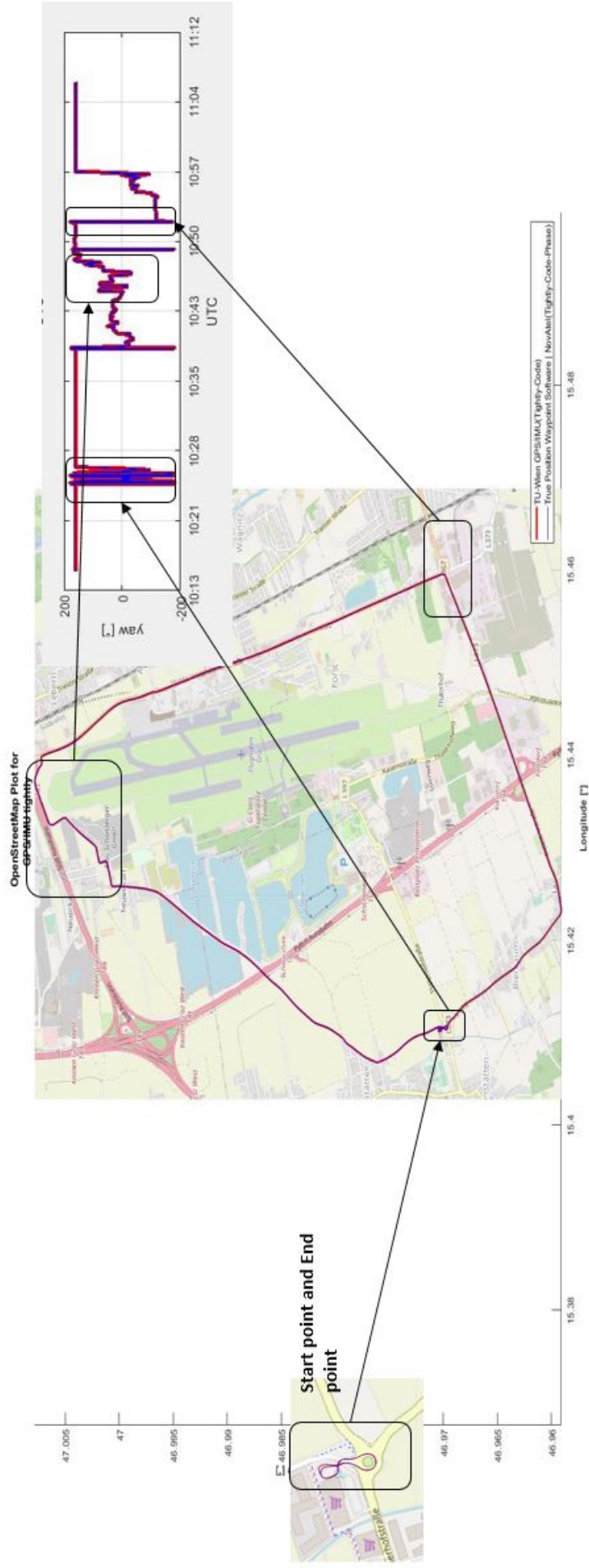


Figure 49: Consistency between the estimated yaw angles and the car's trajectory

7.2.8. The GPS measurement data

In this test case, the obtained rinex file from the rover is based on version 2.11. The original dual-frequency observation file contains the pseudo-range (C1/P1, C2/P2) and carrier phase (L1, L2) measurements. The original raw code measurements are very noisy and unstable. In section 3.6, the method carrier-smoothed-code has already been introduced. Actually, the algorithm firstly uses the carrier-smoothed-code observables for the GPS observation processing to reduce the noise in the pseudo-range measurements. Figures 50 and 51 show the range differences between the code measurements and after the-carrier-smoothed-code measurements along the second test drive.

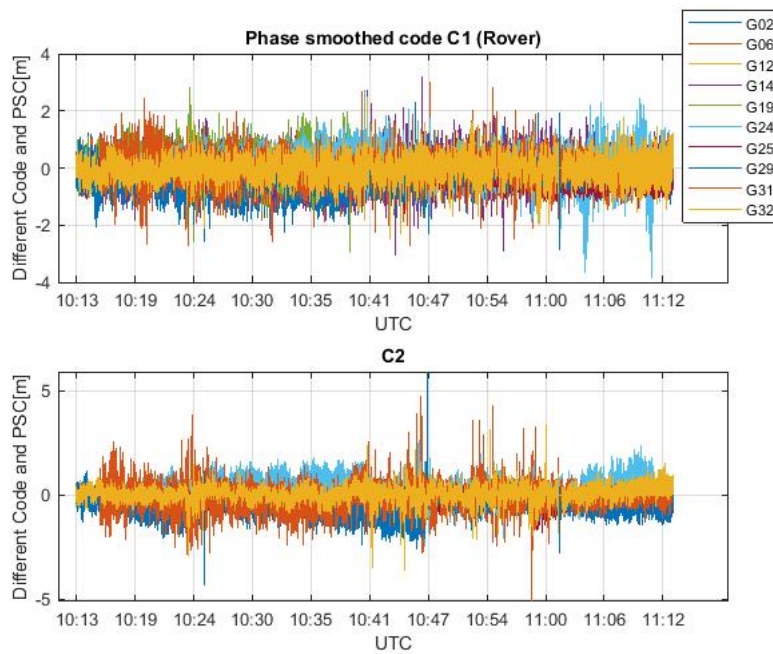


Figure 50: Difference of C1 & C2 range measurements between before and after carrier smoothing in [m], shown for all visible satellites tracked by rover

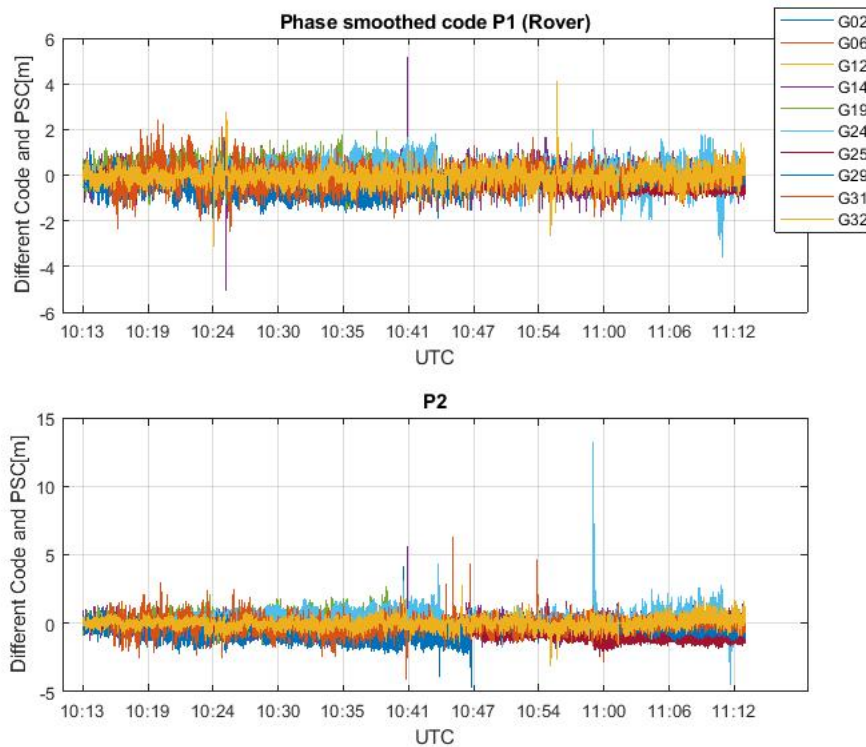


Figure 51: Difference of P1 & P2 range measurements between before and after carrier smoothing in [m], shown for all visible satellites tracked by rover

Using the carrier smoothed code measurements reduces the noise of the pseudo-range measurement. Processing smoothed ranges improves the positioning accuracy and makes the solution more stable, as shown in chapter 7.2.11.

Furthermore, figure 52 shows the number of the visible satellites and the computed Position Dilution of Precision (PDOP values) along the test drive. All chosen satellites have elevations higher than 5 degrees. The maximal number reaches ten satellites, and the minimal number is 5 along the test drive. The PDOP value indicates the quality of the GPS satellites geometry. Usually, when the calculated PDOP values are low such as in the range of 3 or less, the satellites geometry is good. A PDOP value bigger than 7 indicates bad geometry. The PDOP values in the second test drive were mainly below 2. This number indicates an excellent geometry, and the received signals from the satellites are strong. Figure 52 also shows us that when satellite availability decreases, the PDOP values increase.

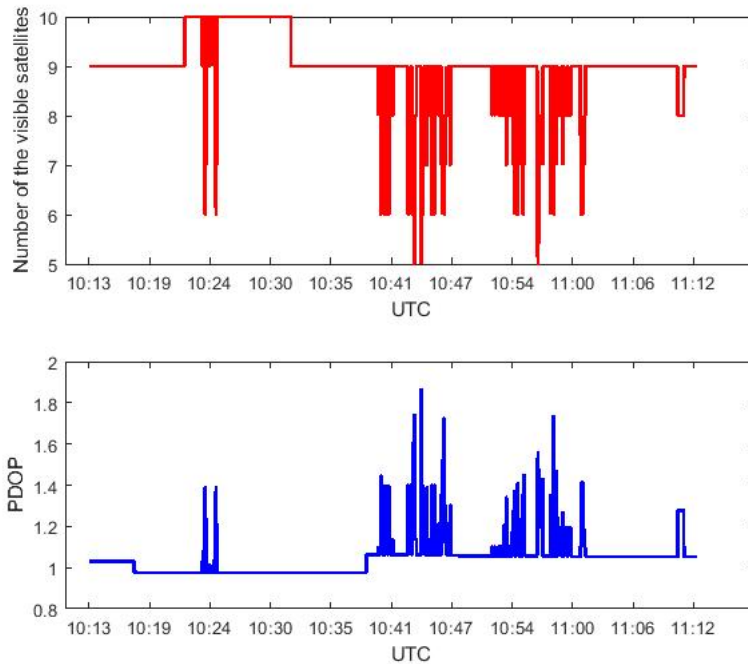


Figure 52: Figure on the top represents the number of visible satellites, figure on the bottom represents the PDOP values along the test drive

7.2.9. The results

In the following, a more detailed look at the results obtained with the TU-developed GPS/IMU integration algorithm (15-states + phase biases Kalman filter) is presented. The software allows to select between three modes for GPS data processing, namely a) DD Phase + Code observations, b) DD Code observations and c) just a zero-difference code solution (SPP). The DD combined code and carrier phase approach including IMU measurements at the raw data level is consistent with the approach used by reference software. The only difference is that the reference software processes GPS/GLONASS observations, while the TU-Wien solution is just based on GPS observations. The results in figures 53, 55 and 57 show that both softwares provide similar estimated navigation solutions along the test drive. Figure 53 displays the three-dimensional position solution of the test drive. The red line shows the solution calculated by reference software, and the blue line represents the trajectory calculated by TU-Wien software. The estimated trajectory is represented in a geographic coordinate system in Latitude, Longitude and Height.

It should also be noted that there are two big jumps in the height time series at 10:43 h and 10:55 h of about 5 meters. The reason is that the car traveled up to two bridges used for

crossing the highway. The height of the bridges is estimated to be about 5 meters.

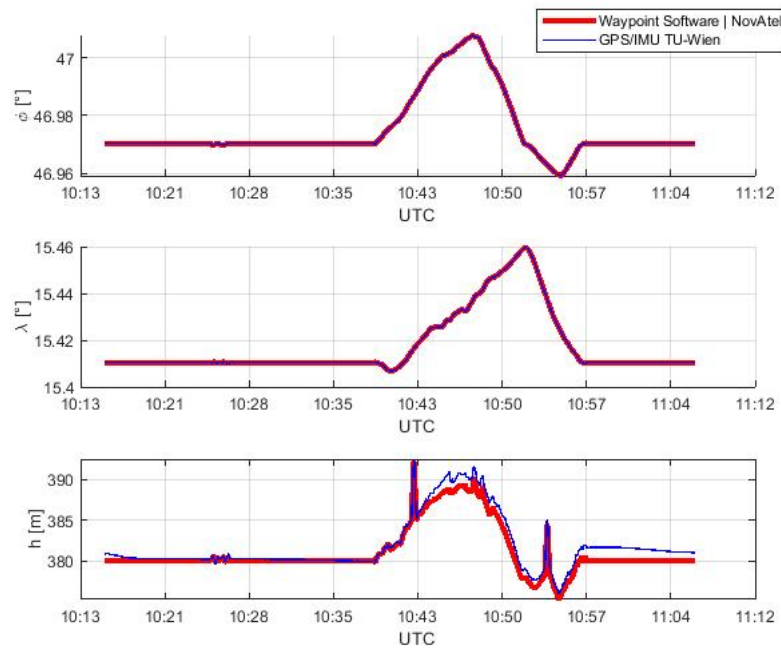


Figure 53: 3D positioning solutions processed by TU-Wien software and reference software (Both solutions are estimated by using the tightly coupled approach based on the GPS DD combined code-and carrier-phase observation)

The differences in North and East components are relatively small at the few dm-level. In the height, the deviation is within a range of a few decimeters up to 1.5 meter.

Due to the rover's movement and therefore varying signal blockings, the software has to account for permanent changes of visible satellites. A problem, especially evident, when the reference satellite has to be switched. The software switches the reference satellite in case the new reference satellite's elevation is at least 1 degree higher than the elevation of the current one.

At 10:46h, the signal of the reference satellites has been lost several times for short periods. Each signal loss causes a switch to new reference satellites and subsequently requires that all ambiguities are reinitialized and to be referred to the new reference. This problem makes the results at these epochs worse and more unstable, clearly seen in figure 54. Another problem occurs at 10:42. There is a big jump of all visible satellites in the carrier-phase measurements of the reference station at this epoch. The implemented cycle slip detector could identify these jumps at the current epoch, and the software reinitialized the ambiguity for all satellites to fix this problem automatically.

Furthermore, the golden line represents the number of visible satellites. Usually a change in the number of visible satellites causes coordinate difference between both software solutions.

When the car was at rest, the deviations between both software solutions of course diminished and the results are more stable.

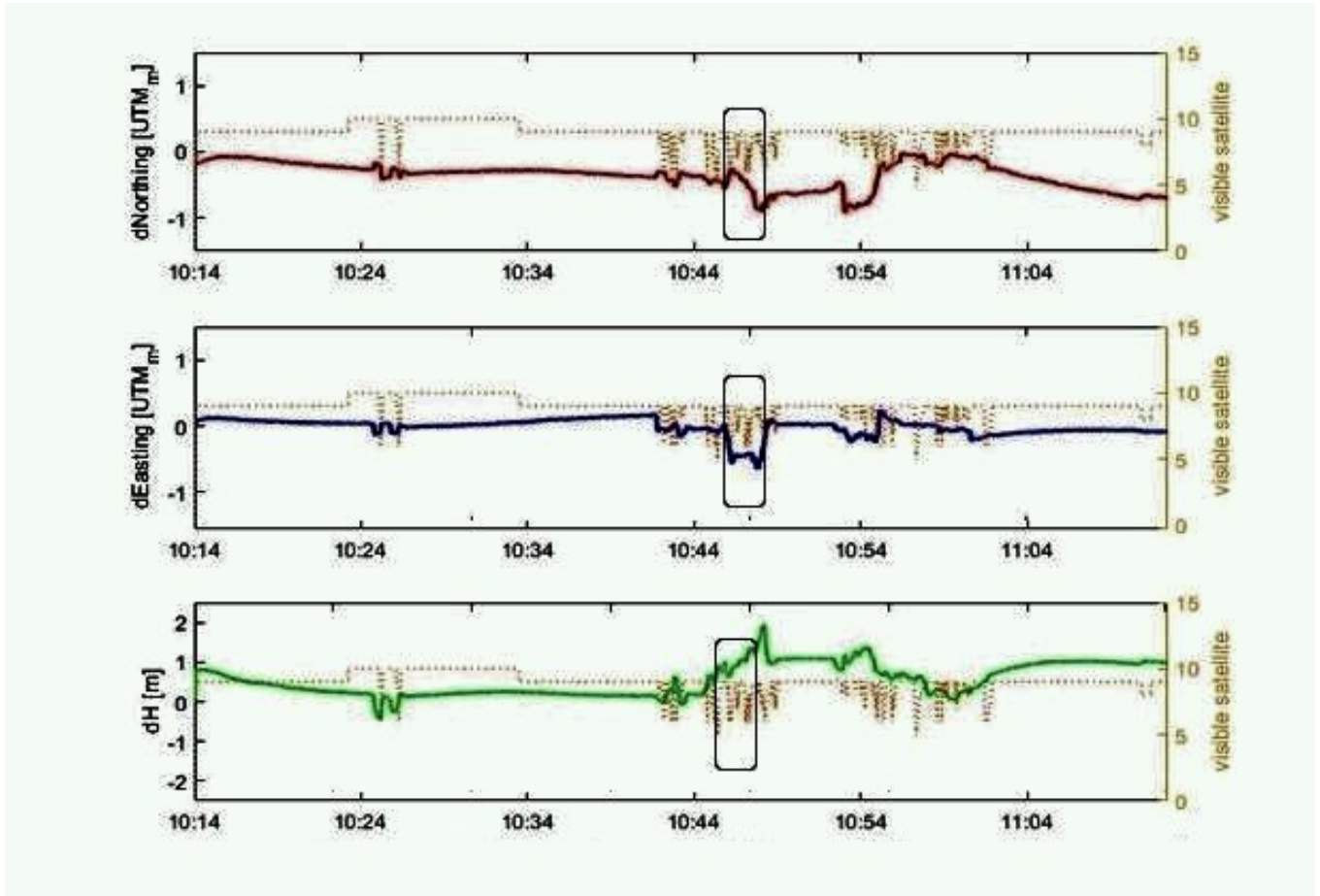


Figure 54: Positioning differences between solution processed by Waypoint software and TU-Wien software

Figure 55 shows the calculated three-dimensional velocity solutions of both softwares in North, East and Down directions. The estimated velocities are reported in meter per second (m/s). The differences between the new filter and the Waypoint solution are obviously minimal.

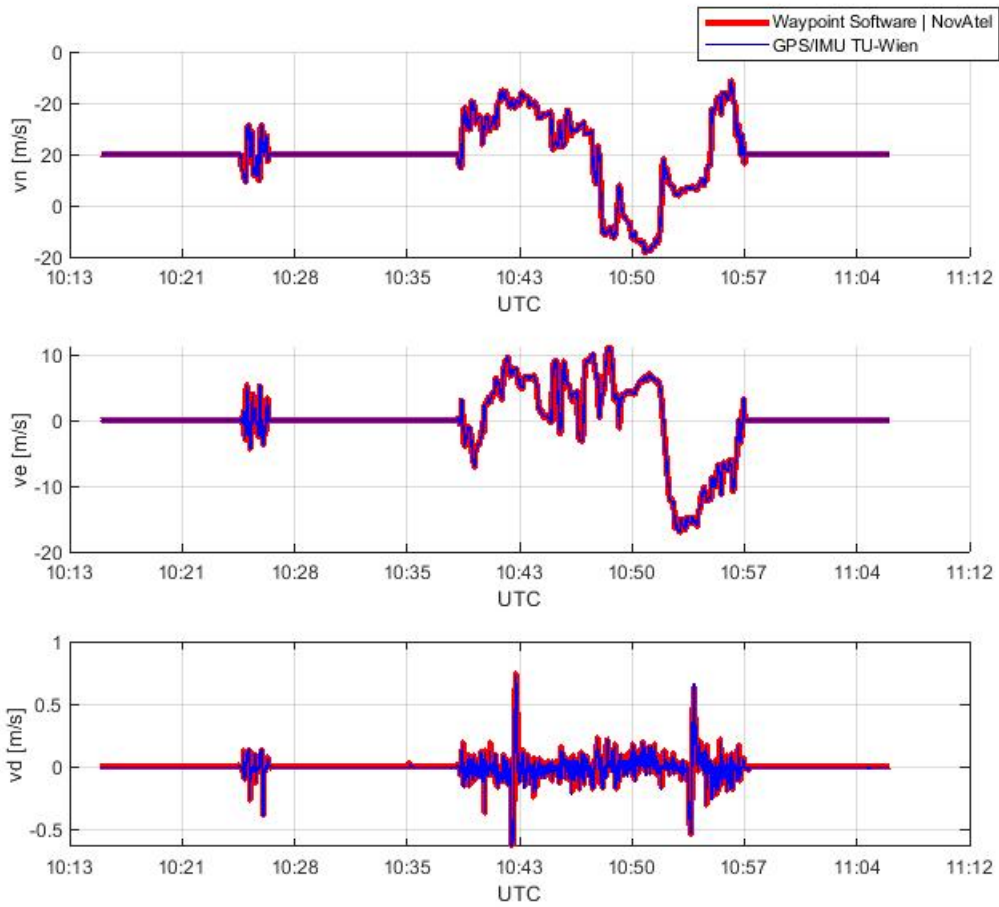


Figure 55: 3D velocity solutions processed by TU-Wien Software and reference software (both solutions are estimated by using the tightly coupled approach based on the GPS DD combined observation)

To assess the quality of the velocity components, the differences between both solutions are depicted in figure 56. The figure clearly shows that the deviations in all three directions are within a range of 0.05 m/s.

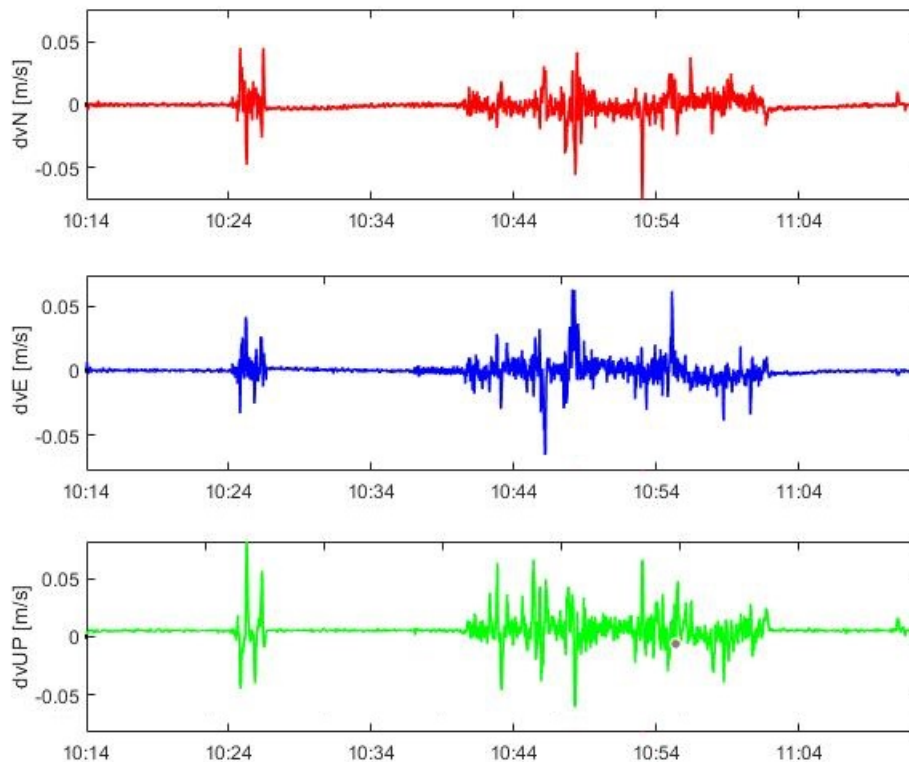


Figure 56: Velocity differences between solutions processed by Waypoint software and TU-Wien software

Last, but not least, figure 57 represents time series of pitch, roll and yaw angles (degree) along the test drive. Similar to figures 53 and 55, the difference between both solutions are quite small.

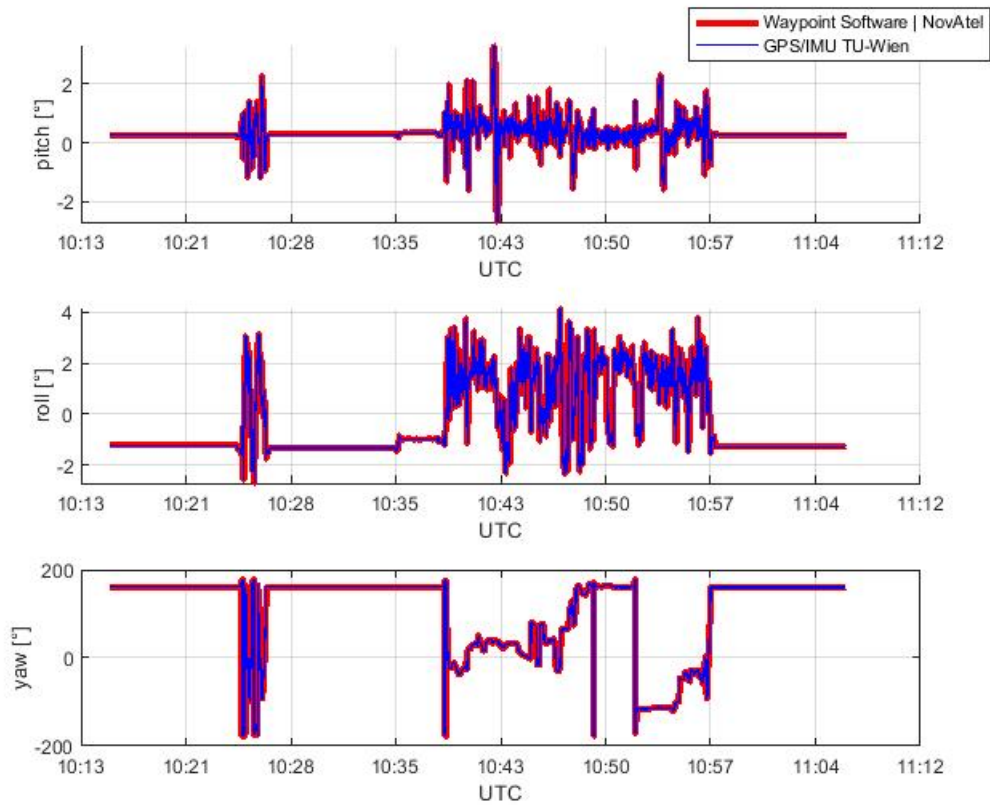


Figure 57: Roll/Pitch/Yaw solutions processed by TU-Wien Software and reference software

Figure 58 shows pitch, roll and yaw angles differences estimated by means of TU-Wien software and reference software. While the car is in movement, deviations in roll and pitch angles within 0.05 degrees show up, while as expected the yaw angle estimates deviate within a larger range of 0.2 degrees.

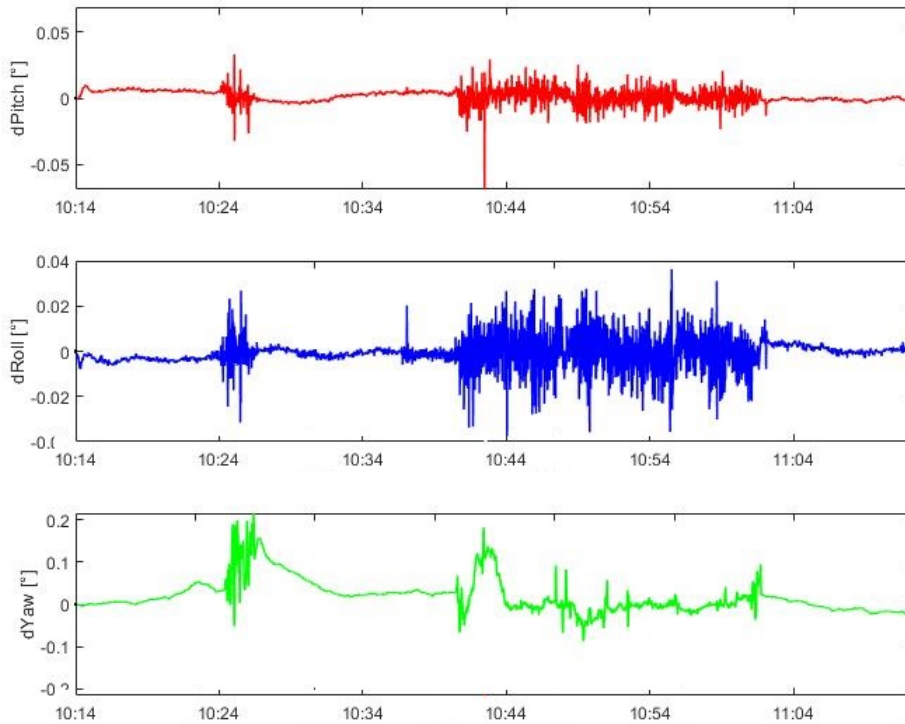


Figure 58: Pitch/Roll/Yaw differences from Waypoint software and TU-Wien

7.2.10. Combined processing versus other GPS observation processing

As noted above, the TU-Wien software allows to select between three modes for GPS data processing. figures 59, 60 and 61 below display the difference time series of all approaches with respect to the reference solution. The red line always refers to the DD code+phase approach. The blue line represents the differences between the TU-Wien solution based on the DD code-only processing algorithm compared with the reference solution. The green line shows the differences of the SPP solution with respect to the Waypoint reference solution.

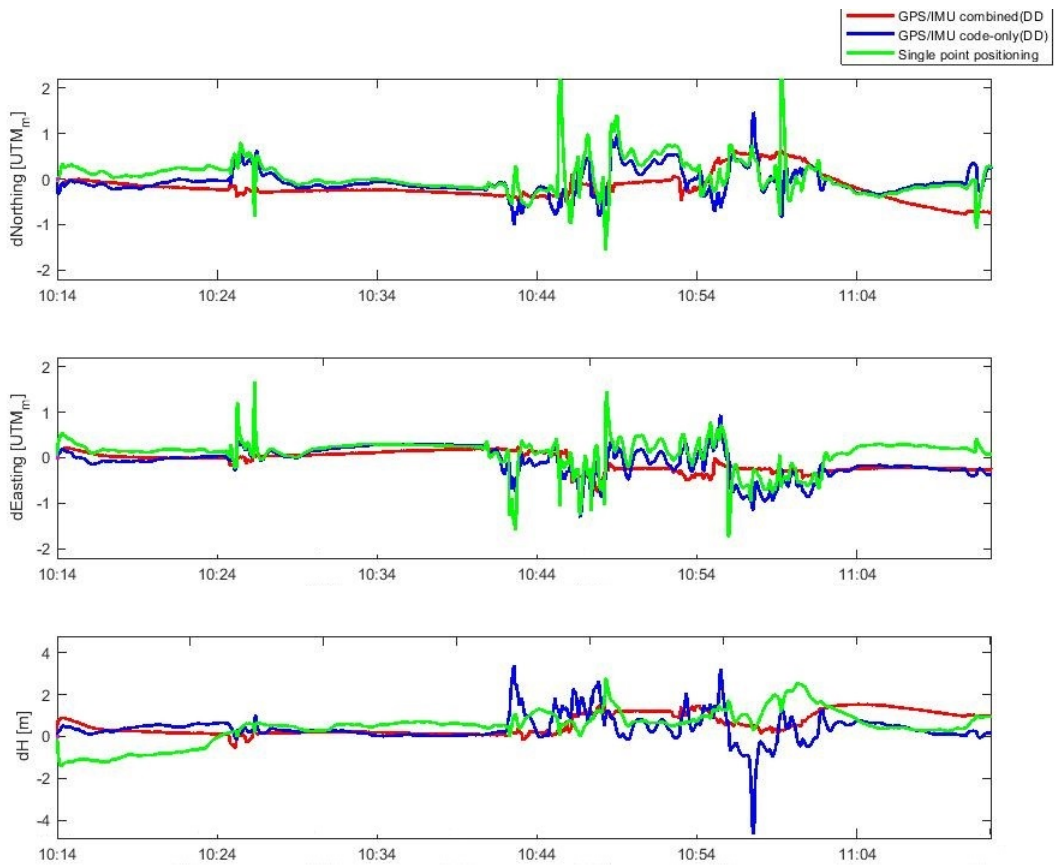


Figure 59: Position differences with respect to Waypoint reference solution (red: DD code+phase; blue: DD code only; green: code SPP)

As assumed, the differences between the TU-Wien solution based on the DD phase + code processing and the reference solution are smallest and most stable over the whole test drive (see figure 59). The SPP time series experience the most significant differences in the horizontal position (up to meter level), and interestingly the DD code-only approach indicates the biggest differences in the vertical.

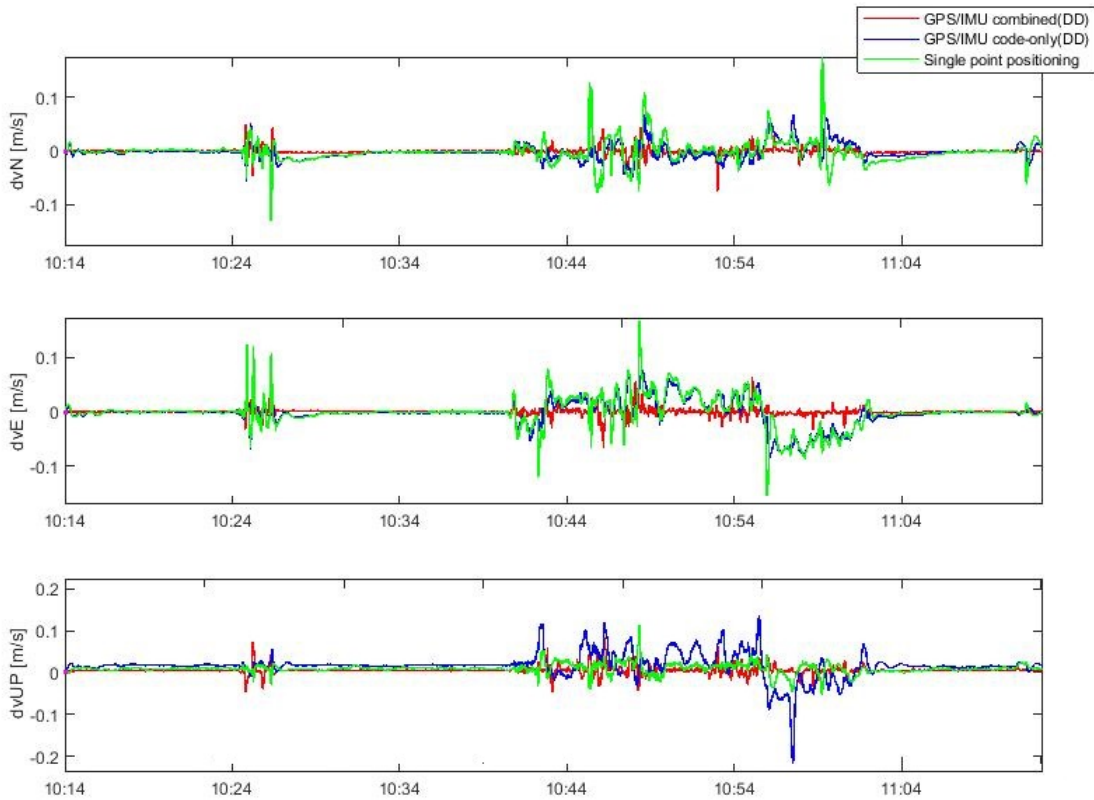


Figure 60: Velocity differences with respect to Waypoint reference solution (red: DD code+phase; blue: DD code only; green: code SPP)

Concerning the velocity components again the DD code+phase approach delivers the best and most stable solution as expected (see figure 60), the red line represents the slightest differences in the velocity solutions in all three directions. More noisy but with only slightly larger velocity differences the DD code solution performs better than expected in the plane coordinates. The SPP solution experiences peaks in the velocity differences up and slightly above 0.1 m/s.

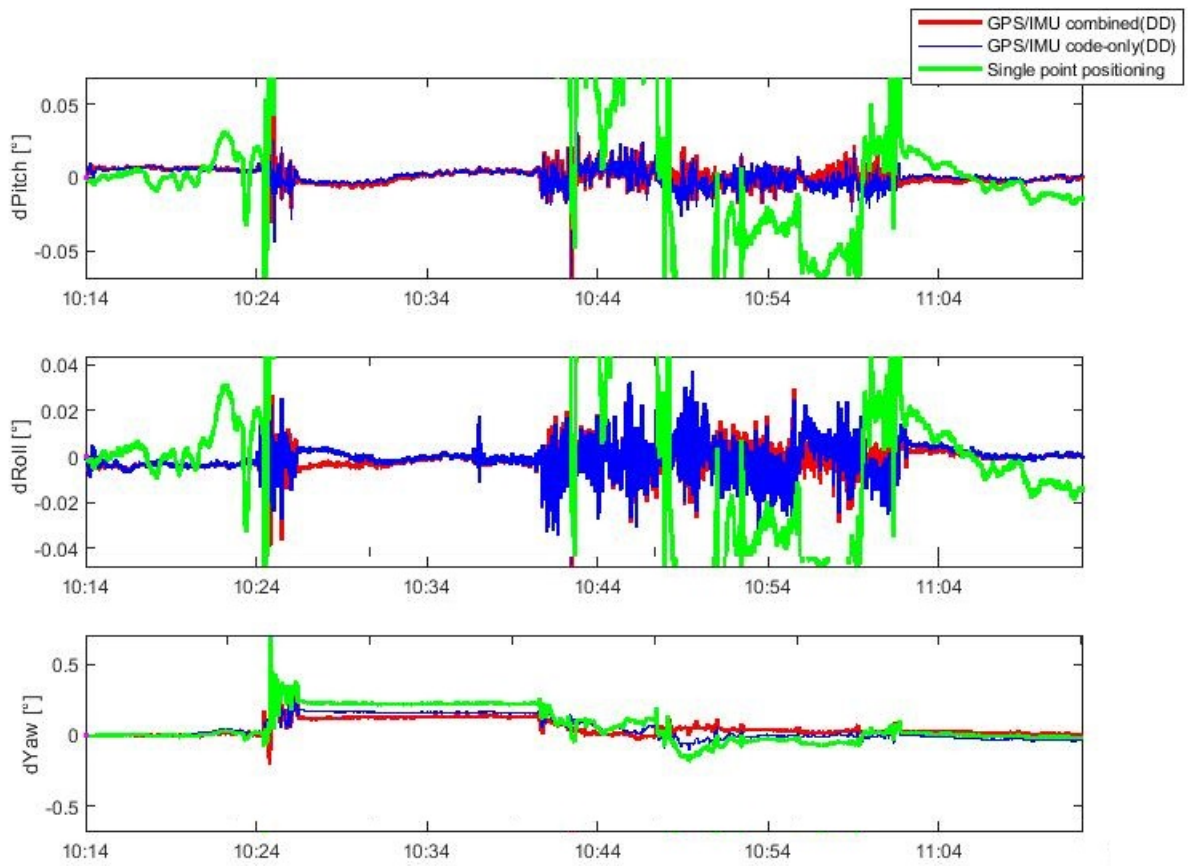


Figure 61: Differences in roll, pitch and yaw angles with respect to Waypoint reference solution (red: DD code+phase; blue: DD code only; green: code SPP)

As expected, the differences in the attitude angles are quite comparable for the DD code+phase and DD code-only solutions and are within 0.02 deg for roll and pitch and within 0.2 deg for the yaw angle. The SPP solution clearly performs worst as displayed in figure 61.

Table 9 shows the root mean square (RMS) and mean values for all 9 parameters of all processing schemes with respect to the reference solution.

Positioning	Mean(m)			RMS(m)		
	ΔN	ΔE	ΔH	ΔN	ΔE	ΔH
DD combined code + carrier phase	0.0074	0.0868	0.3963	0.3185	0.2199	0.5585
DD code-only	-0.0716	-0.0855	0.3993	0.2903	0.3259	0.7492
Single point positioning	0.0504	0.1030	0.4838	0.3675	0.3451	0.9161
Velocity	Mean(m/s)			RMS(m/s)		
	Δv_N	Δv_E	Δv_D	Δv_N	Δv_E	Δv_D
DD combined code + carrier phase	-0.0007	0.0001	0.0057	0.0065	0.0067	0.0108
DD code-only	-0.0022	-0.0022	-0.0165	0.0135	0.0232	0.0327
Single point positioning	0.0020	-0.0011	0.4838	0.0197	0.0272	0.0145
Roll/Pitch/Yaw	Mean(deg)			RMS(deg)		
	ΔR	ΔP	ΔY	ΔR	ΔP	ΔY
DD combined code + carrier phase	-0.0014	0.0012	0.0535	0.0057	0.0054	0.0769
DD code-only	-0.0009	0.0007	0.0481	0.0064	0.0057	0.0950
Single point positioning	-0.0012	0.0007	0.0627	0.0070	0.0063	0.1304

Table 9: Mean value and RMS of all three implemented GPS approaches

From the numbers it can be concluded that the TU developed tightly coupled GPS/IMU integration algorithm based on the DD code + carrier phase processing performs best with respect to reference solution. The RMS values of almost all state parameters are the smallest. Thus, the integrated IMU/GPS DD code + carrier phase approach provides the most stable navigation solution.

7.2.11. The carrier smoothed code

It has to be noted that the previous results were based on already filtered GPS code ranges. The code observations were carrier-smoothed before entering the filter as the original raw code measurement is very noisy and unstable (see chapter 3.6). This subsection describes the improvement in accuracy of the estimated navigation solution calculated by TU algorithm when the PSC method is used. To show the effect of this smoothing the DD code-only solution was processed once with the raw code observations and once with the carrier smoothed code ranges. The differences in the positioning solutions are plotted in figure 62. The blue line shows the difference of the software solution with the original code measurement when compared to the reference solution. The reference solution is based on the tightly coupled GPS/GLONASS

combined DD processing, The red line shows the difference between both solutions after using the PSC.

After reducing the code noise, the position noise decreases from meter to dm level. Figure 63 shows two differences in velocity solutions.

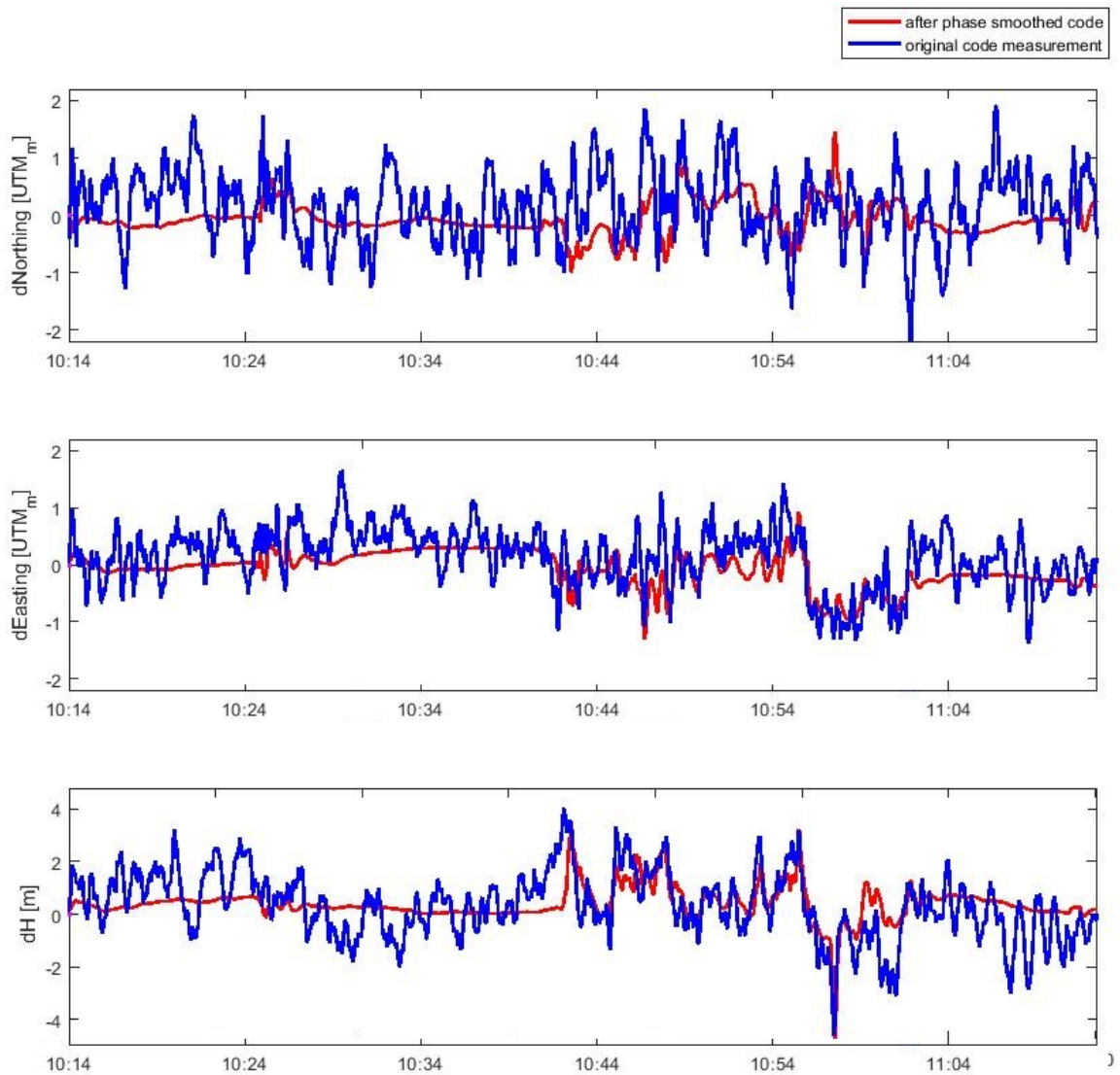


Figure 62: Position differences between the solutions processed by Waypoint software and the TU-Wien software. The blue line represents the solution differences calculated with the original GPS code measurements, and the red line shows the solution differences after using phase smoothed code.

Also in the figures 63 and 64 displaying the velocity and attitude states the smoothing

effect is clearly visible. The velocity differences of the smoothed-code solution are clearly smaller and more stable.

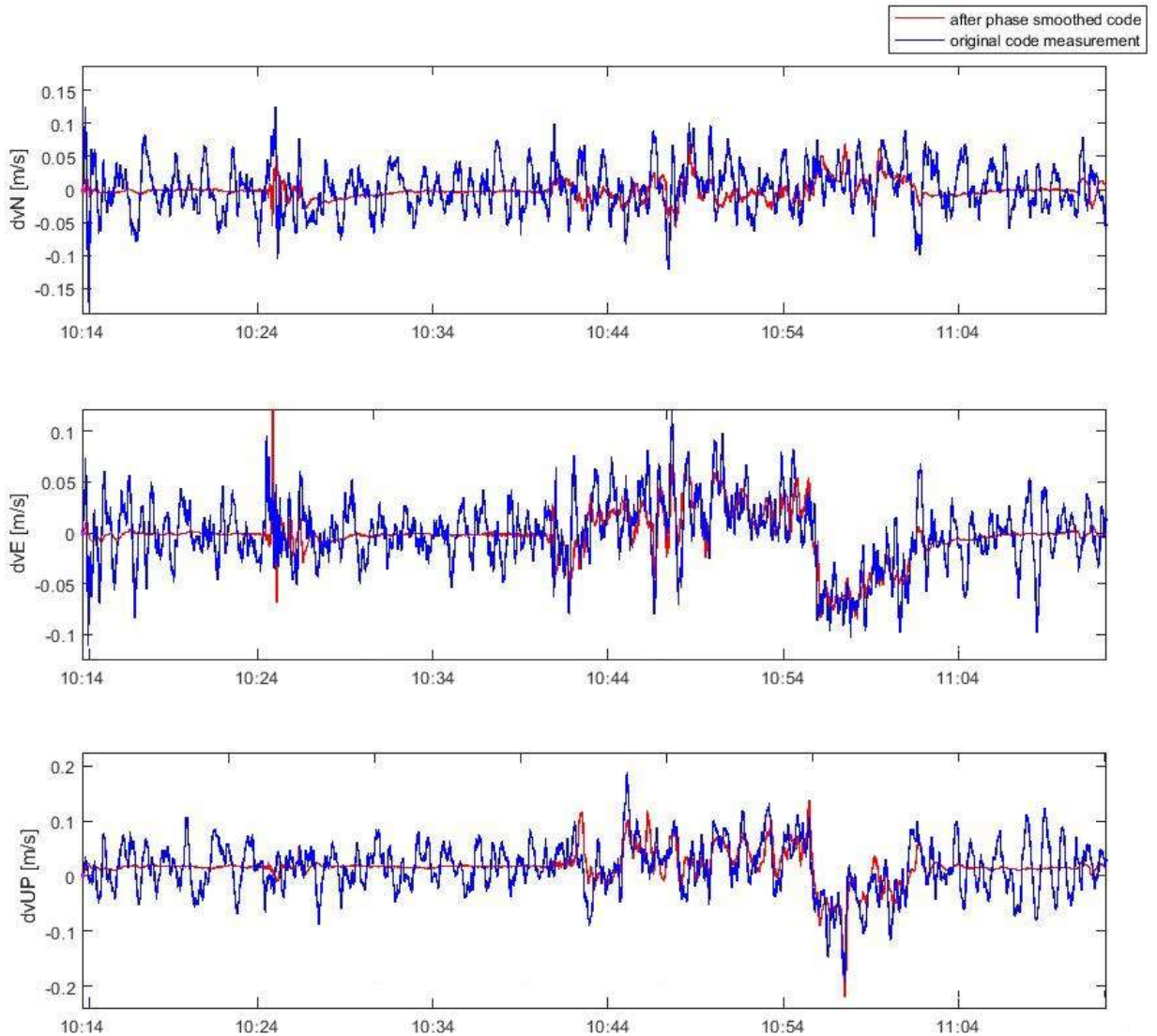


Figure 63: Velocity differences between the Waypoint reference solution and the DD code only solution based on raw code ranges (blue) and phase-smoothed-code-ranges (red)

Figure 64 shows the comparisons in roll, pitch and yaw angles of the vehicle. Here the differences are not that visible than four position and velocity. The PSC solution still outperforms the raw code solution, but for roll and pitch the differences are small. In yaw the PSC solution is clearly more stable but at 10:24h a large difference of up to 0.2° remains. The raw code measurement solution provides a large difference up to -0.5° .

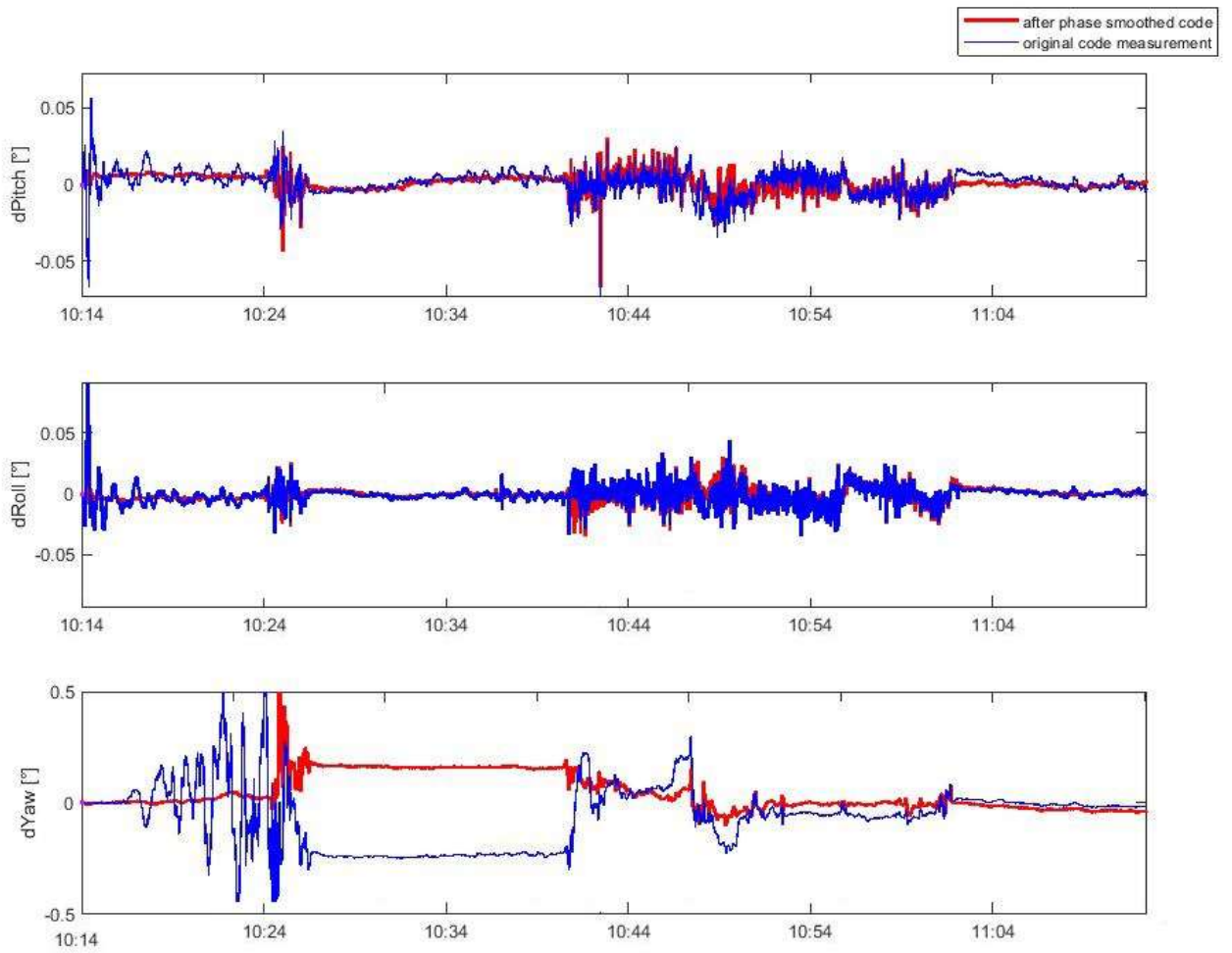


Figure 64: Differences in roll, pitch and yaw angles between the Waypoint reference solution and the DD code only solution based on raw code ranges (blue) and phase smoothed code ranges (red)

The root mean square (RMS) and mean values with respect to the Waypoint reference software solutions are represented in table 10.

In general, the code measurements are noisy, the carrier-phase measurements are more accurate. Therefore, smoothing GPS pseudo-range using both code and carrier phase observations has also been considered. The differences between the original code measurements and after using PSC on the GPS positioning solution are represented. It can be concluded that an a priori code smoothing improves the accuracy of positioning and stability. The PSC algorithm is already adopted in the TU-Wien software, which can improve the GPS DD performance, and the positioning solutions are more accurate and stable.

Positioning	Mean(m)			RMS(m)		
	ΔN	ΔE	ΔH	ΔN	ΔE	ΔH
Original code measurement	0.1608	0.1238	0.3446	0.6465	0.5304	1.3020
After PSC	-0.0716	-0.0855	0.3993	0.2903	0.3259	0.7492
Velocity	Mean(m/s)			RMS(m/s)		
	Δv_N	Δv_E	Δv_D	Δv_N	Δv_E	Δv_D
Original code measurement	0.0015	-0.0011	-0.0160	0.0356	0.0330	0.0477
After PSC	-0.0022	-0.0022	-0.0165	0.0135	0.0232	0.0327
Roll/Pitch/Yaw	Mean(deg)			RMS(deg)		
	ΔR	ΔP	ΔY	ΔR	ΔP	ΔY
Original code measurement	-0.0010	0.0009	-0.0649	0.0079	0.0080	0.1662
After PSC	-0.0009	0.0007	0.0481	0.0064	0.0057	0.0950

Table 10: Mean value and RMS after using PSC and the original code measurements

8. Conclusions and outlooks

The contribution of this doctoral thesis can be summarized as follows. At first, a software solution for a loosely coupled GPS/IMU/ODO sensor integration has been successfully designed and implemented. In addition, a fall-back scenario just based on an IMU/ODO data integration has also been tested during the GNSS outage. At second, an algorithm of the tightly couple GPS/IMU integration has been successfully implemented. The developed Kalman filtering is realized by using the GPS pseudo-range, the carrier-phase and the Doppler shift measurements simultaneously from a rover and a base station, which merges with the solution from the IMU measurement data.

In the first test scenario, the trajectory of a moving train was processed by the in-house software and the results were compared with the reference solution calculated with the commercial iMar software. The differences of the navigation solution are quite small, when the GNSS signal processing with the satisfying quality is available. When the train passes a bridge, the position differences increase up to 2 meter during 40-second GNSS signal gaps, the velocity differences raise up to 0.2 m/s, and the difference in pitch, roll and yaw angles up to 1 degree. Due to the noise in the angular rate data in the z-direction, the estimated yaw angle is unstable especially, when the train is at rest.

Adding the odometer data, the proposed IMU and odometer integration drifts with time, which is still not good enough for positioning in the urban areas where the GNSS signal is blocked. In the future work, the use of an omnidirectional vision sensor (OV) as additional sensor to the integrated system can be considered to improve the accuracy of the positioning solution in the urban area. An OV sensor can compute accurate attitude solution during a long


period of time. This can improve the quality of the attitude solution, which comes from the IMU standalone drifts during GNSS outage [Kim+11]. The attitude information from OV Sensor as well as odometer measurement data can be combined with IMU measurement to provide stable position and attitude solution without the error accumulation.

In the second step, a software solution for a tightly coupled GPS/IMU sensor integration, which has been designed and implemented. The algorithm supports different GPS observation processing strategies based on the DD phase and code observations but can handle also code-only scenarios. The second test scenario concerns the trajectory of a moving car, which was processed by the in-house software and the results were compared with the reference solution calculated with the commercial Waypoint software. The position differences for the DD code+phase approach in the plane coordinates are quite small at the few dm level while in height differences up to 1m show up. The differences in the velocity components and attitude angles are at the 0.05m/s and 0.1 deg level. Also DD code and SPP scenarios were processed to evaluate the potential of cheap sensor data just offering phase-smoothed code ranges in combination with high quality IMU data.

Lastly, it can hereby be confirmed that the combination of multi-sensor observations can offer improved reliability, availability, and accuracy of positioning. In future, the tightly coupled filter software solution shall be capable to process also odometer data. This augmentation would of course stabilize the state vector in periods of GPS signal blocking and shall be used for processing sensor data tracked by fast and slowly moving trains. From the present perspective, multi-sensor navigation system application becomes more and more popular and is expanding rapidly. Adapting multi GNSS (e.g. GPS and Galileo) and multi-frequency processing into the presented integrated system can also be proposed. It can be expected that GNSS signals from the more satellite constellations can be tracked and the quality of the positioning processing will be improved. PPP is an absolute GNSS positioning approach, which differs from traditional DD approach. Kinematic PPP approach becomes more and more popular, which can be used in smartphone [Gla22]. Hence, the future work can also focus on the fusion of the GNSS PPP observation data and the IMU data to determine the navigation solution of a moving object.

Appendices

Appendix A Data sheet of iMAR iNAV-RQH-003




GESELLSCHAFT FÜR INERTIALE MESS-,
AUTOMATISIERUNGS- UND REGELSYSTEME MBH
WWW.IMAR-NAVIGATION.DE

Technical Data of iNAV-RQH-003

Data Output:	Heading, Roll, Pitch, Angular Velocity, Velocity (Body and World), Position, Raw Data, Internal Status Information, Odo. and GPS inf.
True Heading:	< 0.04° sec(lat) free inertial; 0.01° with DGPS, 0.008° postproc RTK
Altitude Accuracy:	< 0.01° (< 0.005° with DGPS, 0.003° postproc with RTK aiding)
Position Accuracy:	0.8 nm/hr free inertial; < 1 m GPS (S/A off) and < 10 cm RTK online, < 30 cm DGPS and 2 cm RTK/INS postproc, < 0.1 % distance travelled (with odometer and GPS, applic. depend.) < 0.2 % dist.trav. on underwater vehicles (incl. RDI DVL interface)
Velocity Accuracy:	10 mm/s (aided with L1/L2 RTK GPS receiver, < 3 mm/s postproc RTK)
Alignment Time:	< 12 minutes on-shore, < 30 minutes off-shore
Range:	± 450 deg/s (no angle limitation) ±5 g (option 2/5/10/12/25 g)
Drift (unaided) / Offset:	< 0.003 °/hr < 100 µg (at ±5 g range)
Random Walk / Q:	< 0.0025 °/√h < 8 µg/sqrt(Hz)
Resolution:	0.0003° (1,13"), 0.001 deg/s < 1 µg
Scale/Linearity Error:	< 10 ppm / < 5 ppm < 100 ppm / < 20 µg/g²
Axis Misalignment:	< 100 µrad
Data Output Rate:	1...2'000 Hz, bandwidth 400 Hz
Data Latency:	< 2 ms
Data storage:	up to 16 Gbyte on internal flash drive (option)
Output (options):	RS232/422, Ethernet TCP/IP / UDP, CAN, PPT (Pulse Per Time)
Inputs (options):	internal/external (RTK)GPS (option: GPS/GLONASS integrated) , marker event trigger, 3 x odometer (RS422 level)
Synchronization:	Input for pulse-per-second [PPS / SYNC]
Connectors:	MIL-C-38999 III
Power:	11...34 V DC, < 35 W
Temperature (case):	-40...+65°C operating, (+71°C opt.), -46...+85°C not operating
Rel. Humidity:	8...100%, IP67
Magnetic. insensitivity:	< 500 µTesla (5 Gauss)
MTBF / MTTR:	> 25.000 hrs (estimated for surveying applications) / < 30 minutes
Shock, Vibration:	25 g, 11 ms ;60 g, 5 ms (operating); 20...2000 Hz, 3 g rms
Weight / Size:	9.3 kg / approx. 299 x 213 x 179 mm (without connectors)
Software:	internal online Kalman filter, Nav_Command, INS/RTK-GPS postproc. (option)

iMAR is manufacturing and development of inertial navigation and guidance systems for all application areas. All systems manufactured by iMAR are maintained at iMAR in Europe / Germany.

iMAR use latest and high reliable ring laser gyro technology in its advanced inertial navigation and guidance systems for industrial and defence applications.



iMAR GmbH • Im Reihersbruch 3 • D-66386 St. Ingbert / Germany
 Phone: +49-(0)-6894-9657-0 • Fax: +49-(0)-6894-9657-22
www.imar-navigation.de • sales@imar-navigation.de

© iMAR® / 10 (Technical modifications reserved)

Appendix B Data sheet of iMAR RailLoc-C



Technical Data iRailLoc-C (typical, rms):

	Rate	Acceleration	Attit./Heading	Position	Velocity	Height
Range:	± 500 °/s	± 8 g	unlimited	unlimited	unlimited	unlimited
Bias Stability (AV) ¹ :	< 2.5 °/h	< 0.1 mg				
Bias (filtered) ¹ :	< 5 °/h	< 1 mg				
Bias day-to-day ² :	< 0.2 °/s	< 2 mg				
Angles (Altitude, Hdg.):			0.1° / 0.2° 0.15° / 0.25°	0.05° / 0.1° RP/Y 0.1° / 0.15° RP/Y	(INS / GNSS, w/o with RTK) (after 10 s GNSS outage, w/o with RTK)	
Position (horizontal plane) ³ :					±0.1 m CEP (INS/GNSS RTK real-time) ±0.6 m CEP (INS/GNSS with SBAS) ±1.8 m CEP (INS/GNSS) 0.1 % of DT CEP (with odometer, during GNSS RTK outage) ⁴	0.02 m/s (INS / RTK GNSS)
Velocity:						
Noise:	0.15 °/sqrt(hr)	23 µg/√Hz	0.02 °	< 0.01 m	< 0.01 m/s	
Resolution:	< 0.0001 °/s	< 20 µg	0.001 °	< 0.001 m	< 0.001 m/s	
Linearity error:	< 0.2 %	< 0.5 %	< 0.2 %			
Scale factor error:	< 0.3 %	< 0.1 %				
Scale factor (filtered)	< 0.1 %	< 0.07 %	< 0.1 %			
INS / GNSS / ODO proc.:	advanced 42+ state INS/GNSS/+ extended Kalman filter data fusion					
Internal GNSS Engine:	multi-frequency / multi-constellation GPS / GLONASS / GALILEO / BeiDou, SBAS, QZSS, RTK					
processing Power:	2 separate powerful ARM CPUs, one for sensor data fusion and one for other tasks (e.g. map matching; incl. 128 GB SD)					
Data Processing Rate:	up to 500 Hz; PPS timing accuracy better 10 ns					
Data Output Rate:	1...500 Hz; all data available in real time, latency < 3 ms, jitter < 1 ms					
Synchronization:	2 x PPS_OUT (RS422 level, latency < 1 µs); NTP output (GPS time)					
Radio Communication:	integrated GSM-R modem, integrated GSM/LTE modem with 3G/2G fall-back; 2 integrated SIM slots; optional one of both replaceable by integrated 5G modem; recommended antenna (GNSS multi-frequency, GSM-R, LTE): Huber & Suhner 1399.99.0152 (Sencity Rail rooftop MIMO Antenna)					
Interfaces:	<ul style="list-style-type: none"> - 2 x isolated UART RS422 with common 24 V DC (max. 6 W) output, - 2 x isolated UART RS422 with additional PPS trigger output, - 2 x Ethernet 100 Mbit/s (TCP/IP, UDP), NTP Time Server, - up to 3 odometers supported (A or A/B opto-coupler; 3.5...30 V, 5 mA) with isolated supply output (15 V DC, 6 W max.), types of Knorr-Bremse, Baumer, Sick etc. supported - improved odometer output provided (A/B, RS422) to provide velocity with high accuracy (from data fusion, typically better 0.1 %) to ETCS-OBU to allow enlargement distances of ETCS balises (as option) 					
GNSS Correction Data:	DGPS/RTK correction data from base station, if available (via radio modem)					
Data Latency:	< 1 ms (sampling accuracy better 1 µs, time-stamped according to PPS; jitter < 1 ms) in RTK mode					
Connectors:	M12 (D coded, 4 pin) for Ethernet; M12 (x-coded, 8 pin) for UARTs; N sockets for LTE / GSM-R / GNSS antennas; Harting HAN-16A, male as Main Connector (power, odometers)					
Integrated Data Storage:	32 GByte (option: 128 GByte); lasts for several days continuous data sampling as "black-box"					
Graphical User Interface:	MS Windows or LINUX or MacOS based GUI / HMI software iXCOM-CMD for configuration, visualization, data recording, data converting and playback operation					
Power Supply:	nom. 24...110 V DC (extended voltage range 14.4...154 V DC according to EN 50155); < 35 W typ.; reverse and overvoltage protection up to 160 V DC					
Temperature; MTBF:	-40...+71 °C (outer case temperature) operating, -40...85 °C storage; 50'000 hrs (surveying applications)					
Shock, Vibration:	90 g, 11 ms, 10...2'000 Hz 5 g rms (endurance); 10...2'000 Hz 2 g rms (operational)					
Applied Standards:	EN 45545, EN 50121, EN 50155-2017 (certification in progress)					
Mass, size; IP:	3'500 grams, ≈ 202 x 232 x 111 mm ³ (without connectors); IP67					
Operational Support:	railway environment / train applications / ATO					
Part Number:	00230-00005-xxxx					
Deliverables:	<ul style="list-style-type: none"> - iRailLoc-C MEMS based INS with integrated GNSS, radio modem, odometer interface, lab or customized cable set - iXCOM-CMD MS Windows or LINUX or MacOS based GUI software - SW drivers / toolkits under C++ and Python (with SDK under Qt / C) 					
Options:	<ul style="list-style-type: none"> - railway certified GNSS antenna (including support up to GSM-R / 2G / 3G / 4G / 5G mobile communication; FRMCS, when available / required) - support in architecture design for integration into applications requiring SIL 3 / SIL 4 on system level (incl. diversity / redundancy, ETCS integration, CENELEC requirements, RAMS aspects etc.) 					



IMAR Navigation GmbH • Im Reihersbruch 3 • D-66386 St. Ingbert / Germany
 Phone: +49-(0)-6894-9657-0 • Fax: +49-(0)-6894-9657-22
www.imar-navigation.de • sales@imar-navigation.de

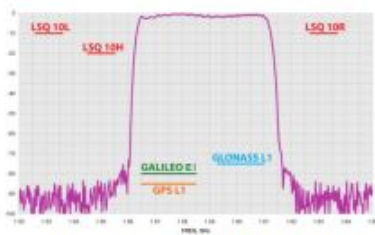
¹ after algorithm converging under sufficient motion excitation with sufficient GPS aiding conditions
² values without sufficient INS/GNSS data fusion conditions; the bias are estimated / compensated during GNSS aiding under motion automatically (Kalman filter)
³ GNSS based altitude deviation is about 1.5 times of GNSS based horizontal error
⁴ position error in relation to distance travelled (DT) during GNSS outages (requires a vehicle motion sensor / wheel sensor) – after suffic. GNSS



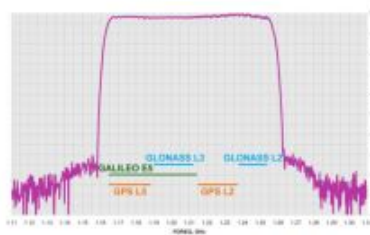
Appendix C GNSS antennen GrAnt G3T



	G3	G3T	G5T
Signal			
Capability	GPS L1, GLONASS L1, GALILEO E1/E2, BEIDOU B1, WAAS L1, EGNOS, MSAS, GAGAN, QZSS L1	GPS L1/L2/L2C/L5, GLONASS L1/L2/L3, GALILEO E1/E2/E5ab, BEIDOU B1/B2, WAAS L1/L5, EGNOS, MSAS, GAGAN, QZSS L1/L2/L2C/L5	GPS L1/L2/L2C/L5, GLONASS L1/L2/L3, GALILEO E1/E2/E5ab/E6, BEIDOU B1/B2/B3, WAAS L1/L5, EGNOS, MSAS, GAGAN, QZSS L1/L2/L2C/L5/LEX
Frequency, MHz	1551–1614	1551–1614 1164–1254	1551–1614 1164–1300
Electrical			
Antenna Gain, dB (typ.)	5.0	5.0 (1551–1614 MHz) 4.0 (1164–1254 MHz)	5.0 (1551–1614 MHz) 4.0 (1164–1300 MHz)
Axial Ratio, dB (max.)		3.0	
Output Impedance, Ohm		50	
LNA gain, dB		32±2 40±2 (optional)	
Noise Figure, dB (typ.)		1.7	
DC voltage, VDC		3.0–15.0	
Current, mA (typ.) @ 5 V	25 32 (40 dB LNA gain)	45 52 (40 dB LNA gain)	45. 52 (40 dB LNA gain)
Power consumption, W (max)	0.38 0.48 (40 dB LNA gain)	0.68 0.78 (40 dB LNA gain)	0.68 0.78 (40 dB LNA gain)
Environmental			
Operating Temperature, °C		-45 – +85	
Storage Temperature, °C		-50 – +85	
Humidity		Waterproof, 100% non-condensing, IP68	
Mechanical			
Antenna type		Microstrip	
Connector		TNC; N-type (optional)	
Weight, g	450	515	
Dimensions, mm		140 x 140 x 62	
Enclosure		Radome: ABS, Base: Aluminum	
Color		Green	
Mounting		5/8-11 or 1-14 inches mount, or 4 holes M5	



Frequency response of J-Shield filter for GPS L1, Galileo E1, and GLONASS L1 bands.



Frequency response of J-Shield filter for GPS L2, GPS L5, GLONASS L2, GLONASS L3, and Galileo E5

References

- [Adm] Federal Aviation Administration. *"Satellite Navigation - GPS - Control Segment"*. DOI: https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/gps/controlsegments.
- [BAE17] Saad Bolbol, Amr H. Ali, and Mona S. El-Sayed. "Performance Evaluation of Precise Point Positioning (PPP) Using CSRS-PPP Online Service". In: *American Journal of Geographic Information System* 6(4) (2017), pp. 156–167. DOI: doi:10.5923/j.ajgis.20170604.03.
- [BAL11] Jared B. Bancroft, M. Haris Afzal, and Gérard Lachapelle. "High Performance GNSS Augmented Pedestrian Navigation in Signal Degraded Environments". In: *International Global Navigation Satellite Systems Association, IGNSS Conference, UNSW Australia* (2011).
- [Bar+22] Sascha Bartl, Manuel Kadletz, Philipp Berglez, and Tomáš Duša. "Mitigating the Threat of Jamming and Spoofing to Aeronautics". In: *Inside GNSS* (2022).
- [Bar96] Yoaz E. Bar-Sever. "A new model for GPS yaw attitude". In: *Journal of Geodesy* 70 (1996), pp. 714–723.
- [Ber+10] Philipp Berglez, Jürgen Seybold, Bernhard C. Geiger, Michael Soudan, Christian Vogel, Christoph Abart, Ajay Pratap Singh, and Bernhard HofmannWellenhof. "Development of a Dual Frequency Software based GNSS Receiver". In: *23rd International Technical Meeting of the Satellite Division of The Institute of Navigation, Portland, OR* (2010).
- [BG09] Sunil Bisnath and Yang Gao. "Precise Point Positioning: A Powerful Technique with a Promising Future". In: *GPS World* 20(4) (2009), pp. 43–49.
- [Bla08] Franz Blauensteiner. "GPS/IMU Integration für die präzise Bestimmung von hoch kinematischen Flugtrajektor". MA thesis. Vienna University of Technology, 2008.
- [Ble97] Geoffrey Blewitt. "Basics of the GPS Technique: Observation Equations". In: Department of Geomatics, University of Newcastle, United Kingdom, 1997.
- [Cam14] Alan Cameron. "GNSS Backup Delivers 5-Meter Accuracy". In: *GPS World* (2014).
- [Che16] Si Rui Chen. "Seamless Train Localization Based on BDS/INS/Odometer/MM Multi-Sensor Navigation System". In: *International Global Navigation Satellite Systems Association, IGNSS Conference, UNSW Australia* (2016).
- [Chi04] Kai Wei Chiang. "INS/GPS Integration Using Neural Networks for Land Vehicular Navigation Applications". PhD thesis. University of Calgary, 2004. DOI: <http://www.geomatics.ucalgary.ca/links/GradTheses.html>.

- [CP14] Peter Cederholm and Darius Plausinaitis. "Cycle slip detection in single frequency GPS carrier phase observations using expected Doppler shift". In: *Nordic Journal of Surveying and Real Estate Research 2014* 10(1) (2014), pp. 63–79.
- [CS14] Robert D. Christ and Robert L. Wernli Sr. "*The ROV Manual, A User Guide for Remotely Operated Vehicles, 2nd Ed*". Elsevier Science, 2014, pp. 453–475. ISBN: 9780080982885. DOI: "<https://doi.org/10.1016/C2011-0-07796-7>".
- [Dac15] Rolf Dach. "*User Manual of the Bernese GNSS Software, Version 5.2*". 2015. DOI: <http://www.bernese.unibe.ch>.
- [Daw10] Karol Dawidowicz. "Antenna Phase Center Variations Corrections in Processing of GPS Observation with Use of Commercial Software". In: *Technical Sciences* 13(1) (2010), pp. 120–132. DOI: 10.2478/v10022-010-0012-9.
- [El-04] Naser El-Sheimy. "Mobile Mapping Systems – State of the Art and Future Trends". In: *The session TS SS 3 "Mobile multi-sensor mapping systems" at the "XXth ISPRS Congress", Istanbul, Turkey* (2004).
- [Exp20] NovAtel Inertial Explorer. "*A NovAtel Precise Positioning Product- Waypoint Software 8.90 User Manual*". 2020. DOI: <https://novatel.com/products/waypoint-post-processing-software/inertial-explorer-xpress>.
- [EY20] Naser El-Sheimy and Ahmed Youssef. "Inertial sensors technologies for navigation applications: state of the art and future trends". In: *Satellite Navigation* 1(2) (2020).
- [Fot18] Aristeidis Fotiou. "A Discussion on Least Squares Adjustment with Worked Examples". In: *University course materials, School of Rural and Surveying Engineering, Aristotle University of Thessaloniki* (2018).
- [Fu+18] Qiangwen Fu, Sihai Li, Yang Liu, Qi Zhou, and Feng Wu. "Automatic Estimation of Dynamic Lever Arms for a Position and Orientation System". In: *Sensors* 18(12) (2018). DOI: <https://doi.org/10.3390/s18124230>.
- [Gla22] Franz Markus Glaner. "Towards instantaneous PPP convergence using multiple GNSS signals". PhD thesis. Vienna University of Technology, 2022. DOI: <https://doi.org/10.34726/hss.2022.73610>.
- [Glo] GlobalPOS. "*JAVAD*". DOI: <https://globalpos.com.au/collections/javad>.
- [GMV11a] GMV. "*GPS Ground Segment*". 2011. URL: https://gssc.esa.int/navipedia/index.php?title=GPS_Ground_Segment.
- [GMV11b] GMV. "*Precise Point Positioning*". 2011. URL: https://gssc.esa.int/navipedia/index.php/Precise_Point_Positioning.
- [GMV11c] GMV. "*RTK Fundamentals*". 2011. URL: https://gssc.esa.int/navipedia/index.php/RTK_Fundamentals.

- [Gra07] Jesper Grandin. "Aided inertial navigation field tests using an RLG IMU". MA thesis. Royal Institute of Technology (KTH), 2007.
- [Gro13] Paul D. Groves. *"Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, 2nd Ed"*. Artech House, 2013. ISBN: 9781608070053.
- [GWA06] Mohinder S. Grewal, Lawrence R. Weill, and Angus P. Andrews. *"Global Positioning Systems, Inertial Navigation, and Integration, 2nd Ed"*. A John Wiley & Sons, Inc., Publication, 2006. ISBN: 9780470041901.
- [Has20] Hayo Hase. "Die Bedeutung Geodätischer Observatorien und die Rolle des BKG für das Globale Geodätische Bezugssystem". In: *VDV-magazin* 5(20) (2020), pp. 378–388.
- [He15] Kaifei He. "GNSS Kinematic Position and Velocity Determination for Airborne Gravimetry". PhD thesis. Technische Universität Berlin, 2015.
- [Hem08] Elder M. Hemerly. "Implementation of a GPS/INS/Odometer Navigation System". In: *ABCM Symposium Series in Mechatronics* 3 (2008), pp. 519–524.
- [Hin16] Edgar L. V. Hinueber. "INS/GNSS/Odometer Data Fusion in Railway Applications". In: *2016 DGON Inertial Sensors and Systems, Karlsruhe, Germany* (2016).
- [HLW03] Bernhard Hofmann-Wellenhof, Klaus Christian Legat, and Manfred Wieser. *"Navigation - Principles of Positioning and Guidance"*. Springer Verlag Wien New York, 2003. ISBN: 3-211-00828-4.
- [HS15] Yao Hao and Feng Shen. "A Low-cost IMU/GNSS Cooperative Positioning Method for VANETs in the Urban Environments". In: *Internation Journal of smart home* 9(12) (2015), pp. 255–266.
- [HWD17] H. Han, Jian Wang, and M. Du. "GPS/BDS/INS Tightly Coupled Integration Accuracy Improvement using an Improved Adaptive Interacting Multiple Model with Classified Measurement update". In: *Chinese Journal of Aeronautics* 31(3) (2017).
- [iMa21] iMar. *"Datasheet of iRailLoc-C"*. 2021. URL: <https://www.imar-navigation.de/de/produkte-uebersicht/product-overview-by-product/item/iirailloc-c-advanced-train-localization-and-communication-system>.
- [iMa22] iMar. *"Short Introduction into Inertial Measuring Technology"*. 2022. URL: https://www.imar-navigation.de/downloads/Documentations/iNAT/MAN_Introduction-into-Inertial-Measuring-Technology.pdf.
- [Jek01] Christopher Jekeli. *"Inertial Navigation Systems with Geodetic Application"*. Walter de Gruyter, Berlin New York, 2001.
- [Kap06] Elliott D. Kaplan. *"Understanding GPS Principles and Application: Second Edition"*. Artech House, 2006. ISBN: -58053-894-0.

- [Kim+11] S. B. Kim, J. C. Bazin, Hyung Keun Lee, K. H. Choi, and S. Y. Park. "Ground vehicle navigation in harsh urban conditions by integrating inertial navigation system, global positioning system, odometer and vision data". In: *Radar, Sonar Navigation* 5(8) (2011), pp. 814–823.
- [KM15] Zdzislaw Kowalczyk and Tomasz Merta. "Evaluation of position estimation based on accelerometer data". In: *10th International Workshop on Robot Motion and Control (RoMoCo), Poland* (2015). DOI: 10.1109/RoMoCo.2015.7219743.
- [Lae10] Klemens Laengauer. "Trajektorienbestimmung mittels Integration von GNSS und IMS". MA thesis. Technische Universität Graz, 2010.
- [Lan14] Richard B. Langley. "Innovation: Cycle Slips". In: *GPS World* 12(5) (2014), pp. 46–51. DOI: <https://www.gpsworld.com/innovation-cycle-slips/>.
- [Lee06] Hyung Keun Lee. "Lever Arm Compensation for GPS/INS/Odometer Integrated System". In: *International Journal of Control Automation and Systems* 4(2) (2006), pp. 247–254.
- [Liu+22] Weiping Liu, Bo jiao, jinming Hao, Zhiwei Lv, jiao Xie, and Jing Liu. "Signal-in-Space Range Error and Positioning Accuracy of BDS-3". In: *Scientific Reports* 12(8181) (2022).
- [LW21] Qing Li and Robert Weber. "GNSS/IMU based vehicle Positioning". In: *AHORN 2021, virtual conference* (2021).
- [LW22] Qing Li and Robert Weber. "Tightly coupled GPS/IMU data integration for the estimation of vehicle trajectories". In: *European Journal of Navigation* 1 (2022), pp. 32–45.
- [MMA11] Haroldo Antonio Marques, Joao Francisco Galera Monico, and Marcio Aquino. "RINEX_HO: Second- and Third-order Ionospheric Corrections for RINEX Observation Files". In: *GPS solutions* 15(3) (2011), pp. 305–314. DOI: :10.1007/s10291-011-0220-1.
- [Nie08] Michael Nielsen. "Development and Flight Test of a Robust Optical-Inertial Navigation System Using Low-Cost Sensors". MA thesis. Air force institute of technology, Ohio, 2008.
- [Nil+13] Tobias Nilsson, Johannes Böhm, Dudy D. Wijaya, Ana Tresch, Vahab Nafisi, and Harald Schuh. "Path Delays in the Neutral Atmosphere. in Johannes Böhm Harald Schuh, *Atmospheric effects in space geodesy*". Springer Berlin Heidelberg, 2013, pp. 73–146.
- [Pet15] Mark Petovello. "How Does a GNSS Receiver Estimate Velocity". In: *InsideGNSS* 10 (2015), pp. 38–41.

- [Qia16] Chengyue Qian. "Tropospheric Correction Modeling in SAPOS Reference Network under Large Height Difference Condition". MA thesis. Technische Universität München, 2016.
- [Rea09] Eugenio Realini. "GoGPS-Free and Constrained Relative Kinematic Positioning with Low Cost Receivers". PhD thesis. Politecnico di Milano, 2009.
- [See03] Günter Seeber. *"Satellite Geodesy 2nd Completely Revised and Extended Edition"*. Walter de Gruyter GmbH & Co. KG,10785 Berlin, 2003. ISBN: 3-11-017549-5.
- [Shi01] Eun-Hwan Shin. "Accuracy Improvement of Low-Cost INS/GPS for Land Applications". MA thesis. University of Calgary,Phd thesis, 2001.
- [Shi05] Eun-Hwan Shin. "Estimation Techniques for Low-Cost Inertial Navigation". PhD thesis. University of Calgary,Phd thesis, 2005. DOI: <https://geomatics.ucalgary.ca/links/GradTheses.html>.
- [Sic20a] Jan Van Sickle. *"Differencing"*. 2020. URL: <https://www.e-education.psu.edu/geog862/node/1727>.
- [Sic20b] Jan Van Sickle. *"Multipath"*. 2020. URL: <https://www.e-education.psu.edu/geog862/node/1849>.
- [SZ11] Sanz Subirana and Juan Zornoza. *"Ionosphere-Free Combination for Dual Frequency Receivers"*. 2011. URL: https://gssc.esa.int/navipedia/index.php/Ionosphere-free_Combination_for_Dual_Frequency_Receivers.
- [SZG11] Sanz Subiran, Juan Zornoza, and Adrià Rovira García. *Relativistic Path Range Effect*. 2011. URL: https://gssc.esa.int/navipedia/index.php/Relativistic_Path_Range_Effect.
- [SZG13] Sanz Subirana, Juan Zornoza, and Adrià Rovira García. "Overview of GNSS Positioning Techniques and Code Pseudo-Range Modelling". In: *University course materials, Technical University of Catalonia (UPC)* (2013).
- [SZP11] Sanz Subirana, Juan Zornoza, and M.Hernández Pajares. *"Carrier-Smoothing of Code Pseudoranges"*. 2011. URL: https://gssc.esa.int/navipedia/index.php/Carrier-smoothing_of_code_pseudoranges.
- [TW96] David H. Titterton and John L. Weston. *"Strapdown Inertial Navigation Technology, 2nd Ed"*. Titterton. D. H., 1996. ISBN: 0863413587.
- [VNB20] Sudha Vana, Nacer Naciri, and Sunil Bismath. "Dual-frequency PPP GNSS and MEMS-IMU Integration For Continuous Navigation in Obstructed Environments". In: *Inside GNSS, January 8, 2020* (2020).
- [Web21] Robert Weber. "Navigation". In: *University course materials, Technical University of Vienna* (2021).

- [Wie] Manfred Wieser. "*Reference Trajectory - Exact Knowledge of Your Path*". URL: <https://www.tugraz.at/institute/ifg/research/navigation/reference-trajectory>.
- [Zha+16] Shoujian Zhang, Lei Zhao, Xiantan Li, and Bing Cheng. "A Sequential and Partial Ambiguity Resolution Strategy for Improving the Initialization Performance of Medium-baseline Relative Positioning". In: *Earth, Planets and Space* 68(29) (2016), p. 10.