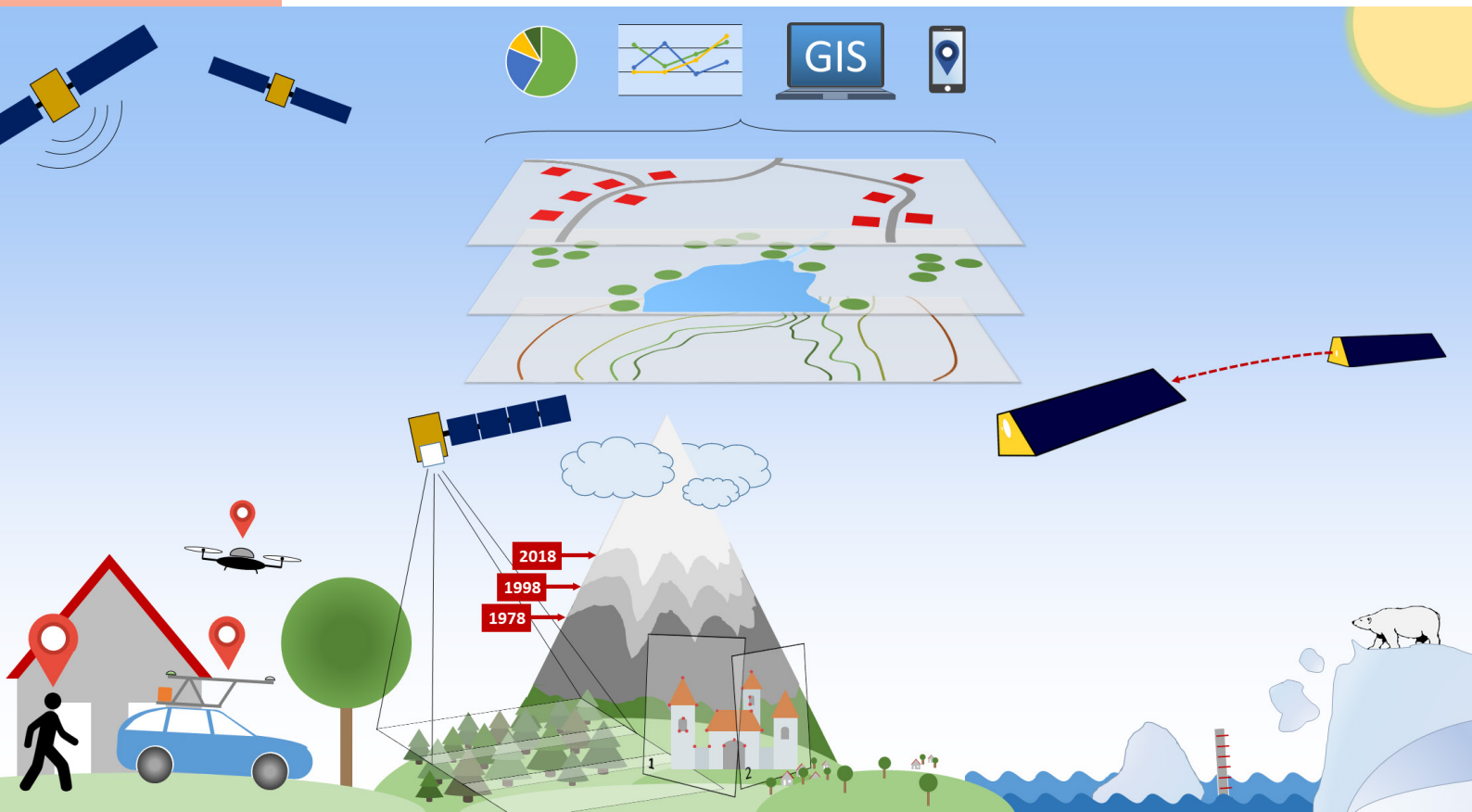
VERLAG DER TECHNISCHEN UNIVERSITÄT GRAZ

Eva Reitbauer

Multi-Sensor Positioning for the Automatic Steering of Tracked Agricultural Vehicles

GEOD 4

MONOGRAPHIC SERIES TU GRAZ
GEODESY



Eva Reitbauer

Multi-Sensor Positioning for the Automatic Steering of Tracked Agricultural Vehicles

Monographic Series TU Graz

Geodesy

Series Editor:

Monographic Series TU Graz

Geodesy Volume 4

Eva Reitbauer

Multi-Sensor Positioning for the Automatic Steering of Tracked Agricultural Vehicles

---

This work is based on the dissertation *"Multi-Sensor Positioning for the Automatic Steering of Tracked Agricultural Vehicles"*, presented at Graz University of Technology; submitted to the Faculty of Mathematics, Physics and Geodesy in 2022.

Supervision / Assessment:
Manfred Wieser (Graz University of Technology)
Terry Moore (University of Nottingham)

Cover design Verlag der Technischen Universität Graz
Cover photo ifG, TU Graz

www.tugraz-verlag.at

E-Book
ISBN 978-3-85125-919-3
DOI 10.3217/978-3-85125-919-3

Acknowledgements

First and foremost, I would like to thank my supervisor Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Manfred Wieser for sparking my interest in integrated navigation and encouraging me to write this thesis. Thank you for your support, trust, and guidance. I could not have wished for a better supervisor.

I would like to express my sincere gratitude to my external appraiser Prof. Terry Moore for taking the time to thoroughly review my thesis and for providing valuable feedback.

Thank you, Dipl.-Ing. Christoph Schmied, for all your support and help in the past years. This project would not have been possible without your expertise and hard work. Thank you for your endless patience and help with all the programming, for co-authoring so many papers, and for always encouraging me not to give up.

I want to thank the whole team of the Working Group Navigation at the Institute of Geodesy. Thank you, Sandra Seybold-Berghold, for always taking care of the administrative issues. Thank you, Bernd Mölg, for constructing the sensor frame and for your help in mounting it on the compost turner. Thank you, Prof. Philipp Berglez, for agreeing to chair the doctoral examination committee. Thank you, Dipl.-Ing. Fabian Theurl, for proofreading Part II of this thesis. Thank you, Dipl.-Ing. Karin Mascher, Dipl.-Ing. Markus Watzko, Dipl.-Ing. Stefan Laller, and Prof. Norbert Kühtreiber, for all the discussions we have had and the advice you have given me. Thank you all for creating such a positive work environment.

Moreover, I want to thank all partners involved in the research project ANTON. Thank you, Thomas Pusch and Richard Schinnerl, for approaching us with your vision to develop an autonomous compost turner. Thank you, Dipl.-Ing. Michael Schedler, for writing the proposal with me and co-authoring two papers. Thank you, Dipl.-Ing. Max

Cichocki, for joining the project team with so much energy and enthusiasm, and for developing the machine control module. Thank you, Prof. Christian Landschützer, for disseminating the results of our research project. I also want to thank the team at Sonnenerde GmbH, especially Wilhelm Wukitsch for steering the compost turner when we collected data for the first tests, and Dominik Dunst for letting us conduct the tests at his composting site. Furthermore, I want to thank the Austrian Research Promotion Agency for funding the project.

I also want to thank Prof. Viktor Kaufmann for creating the 3D point cloud of the composting site and for lending us his measurement equipment whenever we needed it. Furthermore, I want to thank Prof. Werner Lienhart for letting us borrow geodetic equipment from his Institute. I want to thank Dipl.-Ing. Samuel Jost for sharing his expertise on generating the reference trajectory from total station measurements, and Dipl.-Ing. Peter Bauer for his help with the laser scanner.

I would like to thank my family for their love and support. Thank you, dad, for taking the time to proofread my thesis. Thank you, mom, for your emotional support and encouragement. Thank you, Nina, for always being there for me.

Thank you, Fabian, for your unconditional love. Thank you for being there for me when I need someone to talk to, for cheering me up whenever I feel down, and for always supporting me.

Abstract

Automatic steering systems for agricultural machinery are among the main revenue-generating applications of the global GNSS downstream market and their cumulative revenue is expected to grow even further in the next ten years. While automatic steering systems are already commercially available and widely used for tractors or harvesters, no similar system has yet been developed for tracked compost turners.

Compost turners are large agricultural machines used in windrow composting, the most common method of commercial composting. In windrow composting, the organic material is stacked into long, triangle-shaped heaps called windrows. These windrows need to be turned regularly by compost turners to ensure aeration. Since operating compost turners is a repetitive and monotonous job where the driver is exposed to noise, high temperatures, water vapour, and gases, it is often difficult for large composting plants to find workers. It would therefore be a great benefit if compost turners were steered automatically.

The need to develop a tailored automatic steering system for compost turners stems from the different properties that compost turners have compared to tractors or harvesters. Compost turners operate at much slower speeds of only 50-300 metres per hour, and they are subject to strong vibrations, which makes the use of IMUs difficult. Moreover, compost turners are tracked vehicles and are therefore steered differently than tractors or harvesters.

This thesis aims to develop a robust and accurate multi-sensor positioning system that is used in an automatic steering system for tracked compost turners. In the first step, a conventional compost turner was equipped with navigation sensors. The positioning accuracy achievable with these sensors was evaluated by comparing the trajectory estimated by the navigation sensors to a reference trajectory. Based on the results of

this evaluation, a dual-antenna GNSS receiver, a MEMS IMU, a stereo camera and two rotary encoders were selected to be used as navigation sensors of the multi-sensor positioning system.

Next, two integration architectures were developed to fuse information from the sensors selected to estimate the position and attitude of the compost turner: a federated filter architecture consisting of two local filters, and a cascaded filter architecture. Both filter architectures are based on an error-state extended Kalman filter. To take specific motion characteristics of tracked vehicles into account, a tailored odometry model was developed.

To evaluate which filter architecture is most suitable for the automatic steering system, tests were carried out with a prototype of an automatically steered compost turner. For the first tests, the compost turner was steered manually with a remote control along two windrows while data from the navigation sensors were recorded. Then both filter architectures proposed were tested by replaying the data from the navigation sensors, and the trajectories computed were compared to a reference trajectory. Results from the first tests showed that both filter architectures yield a horizontal positioning accuracy of less than 10 cm, but the cascaded filter architecture is better suited for the automatic steering system as it allows for better outlier detection. Moreover, the results showed that the tailored odometry model can be used to bridge GNSS outages of up to 30 seconds while still maintaining a horizontal positioning accuracy of less than 10 cm.

For the final tests, a cascaded filter was used in real time as part of the automatic steering system. The filter fused data from GNSS, the IMU, and the rotary encoders of the machine with the tailored odometry model. The automatic steering system managed to successfully steer the compost turner along the pre-computed routes and achieved a result comparable to that of a human operator.

Kurzfassung

Automatische Lenksysteme für landwirtschaftliche Maschinen gehören global zu den umsatzstärksten Anwendungen des den GNSS nachgelagerten Marktes, und ihr kumulierter Umsatz soll in den nächsten zehn Jahren noch weiter steigen. Während automatische Lenksysteme für Traktoren oder Erntemaschinen bereits im Handel erhältlich und weit verbreitet sind, wurde für kettengetriebene Kompostwender noch kein ähnliches System entwickelt.

Kompostwender sind große landwirtschaftliche Maschinen, die in der Mietenkompostierung eingesetzt werden, die die gängigste Methode der kommerziellen Kompostierung darstellt. Bei der Mietenkompostierung wird das organische Material in langen, dreieckigen Haufen, sogenannten Mieten, aufgeschüttet. Um die Belüftung des Materials zu gewährleisten, müssen diese Mieten regelmäßig von Kompostwendern gewendet werden. Da die Bedienung der Kompostwender eine sich wiederholende und monotone Arbeit ist, bei der der Fahrer Lärm, hohen Temperaturen, Wasserdampf und Gasen ausgesetzt ist, ist es oft schwierig, Arbeitskräfte für große Kompostierungsanlagen zu finden. Es wäre daher von großem Vorteil, wenn Kompostwender automatisch gesteuert würden.

Die Notwendigkeit, ein maßgeschneidertes automatisches Lenksystem für Kompostwender zu entwickeln, ergibt sich aus den unterschiedlichen Eigenschaften, die Kompostwender im Vergleich zu Traktoren oder Erntemaschinen aufweisen. Kompostwender arbeiten mit viel geringeren Geschwindigkeiten von nur 50-300 Metern pro Stunde und sind starken Vibrationen ausgesetzt, was den Einsatz von IMUs erschwert. Außerdem sind Kompostwender kettengetriebene Fahrzeuge und werden daher anders gelenkt als Traktoren oder Erntemaschinen.

Ziel dieser Arbeit ist es, ein robustes und genaues Multisensor-Positionierungssystem zu entwickeln, das in einem automatischen Lenksystem für kettengetriebene Kompost-

wender eingesetzt wird. In einem ersten Schritt wurde ein konventioneller Kompostwender mit Navigationssensoren ausgestattet. Um die erreichbare Positionierungsgenauigkeit mit diesen Sensoren zu bewerten, wurde die von den Navigationssensoren geschätzte Trajektorie mit einer Referenztrajektorie verglichen. Basierend auf diesen Ergebnissen wurden ein GNSS-Empfänger mit zwei Antennen, eine MEMS IMU, eine Stereokamera und zwei Inkrementalgeber als Navigationssensoren für das Multisensor-Positionierungssystem ausgewählt.

Anschließend wurden zwei Integrationsarchitekturen entwickelt, um die Informationen der ausgewählten Sensoren zu fusionieren und so die Position und Lage des Kompostwenders zu schätzen: eine Federated-Filterarchitektur, die aus zwei lokalen Filtern besteht, und eine Cascaded-Filterarchitektur. Beide Filterarchitekturen basieren auf einem Error-State Extended Kalman-Filter. Um die spezifischen Bewegungsmerkmale von Raupenfahrzeugen zu berücksichtigen, wurde ein maßgeschneidertes Odometriemodell entwickelt.

Um herauszufinden, welche Filterarchitektur für das automatische Lenksystem am besten geeignet ist, wurden Tests mit dem Prototyp eines automatisch gelenkten Kompostwender durchgeführt. Bei den ersten Tests wurde der Kompostwender manuell mit einer Fernbedienung entlang zweier Mieten gelenkt, während die Daten der Navigationssensoren aufgezeichnet wurden. Dann wurden die beiden Filterarchitekturen getestet, indem die Daten der Navigationssensoren erneut abgespielt und die berechneten Trajektorien mit einer Referenztrajektorie verglichen wurden. Die Ergebnisse der ersten Tests zeigten, dass beide Filterarchitekturen eine horizontale Positionierungsgenauigkeit von weniger als 10 cm liefern, aber die Cascaded-Filterarchitektur für das automatische Lenksystem besser geeignet ist, da sie Ausreißer besser erkennt. Darüber hinaus zeigten die Ergebnisse, dass das maßgeschneiderte Odometriemodell zur Überbrückung von GNSS-Ausfällen von bis zu 30 Sekunden verwendet werden kann, wobei die horizontale Positionierungsgenauigkeit von weniger als 10 cm erhalten bleibt.

Bei den abschließenden Tests wurde ein Cascaded-Filter in Echtzeit als Teil des automatischen Lenksystems eingesetzt. Der Filter fusionierte Daten von GNSS, der IMU und den Inkrementalgebern der Maschine mit dem maßgeschneiderten Odometriemodell. Dem automatischen Lenksystem gelang es, den Kompostwender erfolgreich entlang

der vorberechneten Routen zu lenken und ein Ergebnis zu erzielen, das mit dem eines menschlichen Bedieners vergleichbar ist.

Contents

List of Figures

List of Tables

Abbreviations

ANTON	Autonomous Navigation for Tracked cOmpost turNers
AOA	Angle Of Arrival
APOS	Austrian Positioning Service
CAN	Controller Area Network
CDMA	Code Division Multiple Access
DBN	Dynamic Bayes Network
ECEF	Earth-Centered Earth Fixed
ECSF	Earth-Centered Space Fixed
EKF	Extended Kalman Filter
EO	Earth Observation
EPOSA	Echtzeit-Positionierung-Austria
EUSPA	European Union Agency for the Space Programme
FDMA	Frequency Division Multiple Access
FFG	Austrian Research Promotion Agency
FMDB	Farm Management Database
GCS	Ground Control Segment
GIS	Geographic Information Systems
GMS	Ground Mission Segment
GNSS	Global Navigation Satellite Systems
GPST	GPS Time
GSD	Ground Sampling Distance
ICP	Iterative Closest Point
IFOG	Interferometric Fiber-Optic Gyroscope
IGS	International GNSS Service
IMU	Inertial Measurement Unit

INS	Inertial Navigation Systems
ITRF	International Terrestrial Reference Frame
LAMBDA	Least-squares AMBiguity Decorrelation Adjustment
LAN	Local Area Network
LIDAR	LIght Detection And Ranging
LOP	Line of Position
MEMS	Micro-Electro-Mechanical System
NDT	Normal Distributions Transform
NTRIP	Network Transportation of RTCM Internet Protocol
PPP	Precise Point Positioning
PRN	Pseudorandom Noise
RANSAC	RANdom SAmpling Consensus
RINEX	Receiver INdependent EXchange Format
RLG	Ring Laser Gyroscope
RMS	Root Mean Square
ROS	Robot Operating System
RTK	Real Time Kinematic
RTT	Round Trip Time
SBAS	Satellite Based Augmentation Systems
SOP	Surface of Position
SPP	Single Point Positioning
SSR	State-Space Representation
TDOA	Time Difference of Arrival
TOA	Time Of Arrival
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter
USB	Universal Serial Bus
UWB	Ultra Wide Band
VIMU	Virtual IMU
VO	Visual Odometry
VRS	Virtual Reference Station

Chapter 1

Introduction

In the last decade, the agricultural sector has experienced a digital transformation (Perez-Ruiz et al., 2021) and high-accuracy Global Navigation Satellite Systems (GNSS) have played a key role in it (GPS World, 05.03.2020).

According to the latest Earth Observation (EO) and GNSS Market Report by the European Union Agency for the Space Programme (EUSPA), automatic steering systems for agricultural machinery are among the main revenue-generating applications of the global GNSS downstream market. While farm machinery guidance dominated shipments of GNSS devices in the first half of the last decade, automatic steering became the leading application in 2019 and 2020 (European Union Agency for the Space Programme, 2022). Furthermore, the cumulative revenue of agricultural GNSS applications is expected to grow to 146.68 bn EUR from 2021-2031. The main revenues will come from automatic steering equipment and commercial augmentation services, which are both inherently linked to automatic steering.

While automatic steering systems are already commercially available and widely used for tractors or harvesters, no similar system has yet been developed for tracked agricultural vehicles. This is exactly where this thesis comes in: it will discuss the development of a multi-sensor positioning system for the automatic steering of tracked agricultural vehicles, focusing on compost turners.

The introductory chapter is structured as follows: Section 1.1 presents the current state of the art of machine steering in precision agriculture. Section 1.2 gives an overview

of essential aspects of windrow composting and describes the need to develop an automatic steering module for tracked compost turners. Section 1.3 introduces the research questions and Section 1.4 addresses the overall structure of the thesis.

1.1 Machine Steering in Precision Agriculture

Modern agriculture relies on Geographic Information Systems (GIS) and GNSS (Lange and Peake, 2021). In a GIS or Farm Management Database (FMDB), different map layers can be used to represent soil types, field topography, and crop varieties. GNSS positions are used to georeference the data that is stored in the FMDB.

The accuracy requirements in precision agriculture depend on the application considered. In general, accuracy can be defined as the difference between the true position and the position computed by the positioning system used. However, in precision agriculture, users are more concerned about repeatability or precision, i.e. the position shift when returning to the same spot in the field in the future.

Field operations in precision agriculture include soil sampling, variable rate application, harvest yield mapping, planting, and automatic steering. For planting or harvesting, the accuracy depends on the crop type. For row crops like soybeans, corn or potatoes, a higher accuracy (5 cm) is needed than for broad acre crops like rye or wheat (submetre). While the accuracy required for soil sampling is about 1-2 metres, automatic steering typically requires an accuracy of less than 5 cm, though this value is also crop-dependent.

Automatic steering systems have many benefits. They allow operating continuously during time-critical operations such as planting or harvesting, reduce crop damage, and minimise soil compaction.

The role of GNSS is fundamental in the absolute positioning and navigation of automous terrestrial equipment (Rovira-Más et al., 2015). To achieve the high accuracies needed for automatic steering, real-time differential correction services are needed (Perez-Ruiz et al., 2021). These services may either be Satellite Based Augmentation Systems (SBAS), such

as the commercial services OmniSTAR by Fugro or Starfire Service by Deere, or ground-based systems such as Real Time Kinematic (RTK) base stations or RTK networks that provide Virtual Reference Station (VRS) data. Since February 2021, Austria has been providing free access to the RTK correction service Austrian Positioning Service (APOS) for agriculture and forestry (Hirt, 2021).

However, when large machines are steered autonomously, we cannot rely on a single sensor. For a robust positioning system, redundancy is crucial. A common approach is to fuse GNSS with Inertial Navigation Systems (INS) (see Noureldin et al., 2013; Han et al., 2017; Ali et al., 2019).

Han et al. (2018) analyse recent developments for automatic steering systems in precision agriculture. They find that in addition to GNSS/INS fusion, vision-based sensors, scanners and tactile sensors are often used.

Automatic steering systems that are already available commercially also rely on a multi-sensor approach. Trimble, for instance, advertises its Autopilot™as a system containing a terrain compensation. The terrain compensation comes from an Inertial Measurement Unit (IMU) (Trimble Inc., 2022). Hexagon's AgrOn AutoSteering system includes wheel angle sensors, and multi-axis gyro sensors. It can also indentify slight slope variations, which indicates that it uses an IMU (Hexagon Agriculture, 2022).

1.2 The Need for Automation in Commercial Composting

To outline why automatic steering systems would be beneficial for the composting sector, compost turners and the process of windrow composting will be introduced in this section.

In the European Union alone, 118 to 130 million tonnes of bio-waste are produced every year (European Commission, 2010), of which about one quarter is recycled (Siebert, 2017). The most common recycling techniques are composting and anaerobic digestion (van der Linden and Reichel, 2020). According to Razza et al. (2018), more than 90% of the bio-waste separately collected in Europe is composted.

(a) **(b)**

Figure 1.1: Windrows at a composting site in Riedlingsdorf, Austria. Seen from (a) above as an orthophoto obtained from UAV photogrammetry and (b) from the front.

In the composting process, bio-waste or organic materials are turned into compost, a biochemically stable product that contains micro-organisms and can store carbon (Sweeten, 2008). When compost is added to soil, the properties of soil are improved, i.e. the organic matter content, water retention capacity and pH buffer capacity are increased (van der Linden and Reichel, 2020).

The most common method of commercial composting is windrow composting (Sweeten, 2008), where the organic material is stacked into long, triangle-shaped heaps, so-called windrows. Figure 1.1 shows how windrows are piled up at a medium-size composting site in Austria. The windrows at this specific site are approximately 3 metres wide and 35 metres long. At larger composting sites, windrows can have a width of 4-7.5 metres.

To support aerobic microbial activity, release moisture, remove excess heat, and to ensure aeration, compost turners turn windrows mechanically. Compost turners are tracked machines (see Figure 1.2) with a spiked drum that is rotated when the machine travels the length of the windrow. With its rotating drum, the compost turner mixes the material and piles it up behind itself.

The windrow turning frequency depends on environmental conditions (Kuhlman, 1990) and may range from daily to weekly. As a general rule, windrows need to be turned more often in the beginning of the composting process. Sweeten (2008) suggests a turning

Figure 1.2: A compost turner (Backhus A55) at a composting site in Alberndorf in der Riedmark, Austria. The organic material is turned with the machine's spiked drum.

schedule where windrows are turned up to three times in the first two weeks, twice in the third week and once each in the fourth and fifth week.

Compost turners operate at low speeds of only 50-300 metres per hour. When an operator drives a compost turner to turn the windrows, he or she is exposed to strong vibrations, noise, high temperatures, water vapour, and gases. Since the working conditions are not a very pleasant and the job is repetitive and monotonous, it is often difficult for large composting plants to find workers. It would therefore be a great benefit if compost turners were steered automatically.

The arguments mentioned above led to the idea of the research project "Autonomous Navigation for Tracked cOmpost turNers (ANTON)". Within this research project, which is funded by the Austrian Research Promotion Agency (FFG) in the course of the Austrian Space Applications Programme, an automatic steering system for an electrically driven compost turner is developed.

1.3 The Research Questions

The aim of this thesis is to develop a robust and highly accurate multi-sensor positioning system for tracked compost turners. The positioning solution is to be used in an auto-

matic steering system developed within the project ANTON. Even though automatic steering systems are already available for tractors or harvesters, no similar system has yet been developed for tracked compost turners.

The need to develop a tailored automatic steering system stems from the different properties that compost turners have compared to tractors or other wheeled vehicles used in agriculture. The first difference that can be noted is the driving speed. Harvesters operate at speeds ranging from 3.0 to 6.5 km/h (American Society of Agricultural and Biological Engineers, 2011) and the optimal field travel speed for tractors is between 6.4 and 9.7 km/h (Grisso et al., 2019). Compost turners, by contrast, operate at much lower speeds of less than 1 km/h (Reitbauer and Schmied, 2021a). Additionally, compost turners are subject to strong vibrations as they turn the windrows with their rotating spiked drums. These strong vibrations and the low operating speed make the use of IMUs difficult, as the vibrations cause accelerometer measurements to be noisy and the low speed makes it difficult to distinguish the actual signal from noise. Moreover, compost turners are tracked vehicles, while tractors or harvesters are wheeled. Tracked vehicles have different motion characteristics and are steered differently than wheeled vehicles.

The accuracy required for the automatic steering of compost turners is less critical than for tractors or harvesters. While an accuracy of 5 cm is needed for crop rows, compost turners work well with a lower accuracy as windrows are not as narrow as crop rows. In discussions with operators of a composting plant, an accuracy of 10 cm was deemed sufficient.

With the established need to develop a tailored multi-sensor positioning system for the automatic steering of tracked compost turners, the following research questions arise:

- Which sensors can be used for the precise positioning of compost turners?
- How can these sensors be combined to optimally estimate the position and attitude of the machine in real-time?
- Which special considerations need to be taken into account for tracked vehicles?
- How can the positioning performance be evaluated?
- What are the advantages of a multi-sensor approach?

The present thesis aims to answer these research questions.

1.4 Structure of the Thesis

This thesis is divided into three parts. **Part I** covers the theoretical foundations in Chapters 2 to 5.

Chapter 2 introduces the fundamentals of navigation, distinguishing between position fixing and dead reckoning, discussing the concepts of redundancy and optimal filtering, and addressing coordinate frames and transformations.

Chapter 3 presents different navigation sensor types and navigation techniques that are commonly used in outdoor scenarios. It discusses high-accuracy GNSS, Inertial Navigation Systems, odometry and image-based navigation.

Chapter 4 introduces Bayes filters, the mathematical tools to estimate an object's state vector from navigation sensor data. It covers Gaussian filters, such as the Kalman filter and its linearized versions, the Extended and Unscented Kalman filter. It also describes non-parametric filters, such as the particle filter, and compares all filters presented with each other.

Chapter 5 explores different concepts and filter architectures for multi-sensor fusion, explaining the terms *loosely, tightly*, and *deeply* coupled that are commonly used in GNSS/INS integration. Then the concept of dual sensor integration is extended to multi-sensor integration, introducing three multi-sensor fusion architectures: centralized, cascaded, and federated integration.

Part II deals with the development of a multi-sensor positioning system for tracked compost turners and covers Chapters 6 and 7.

Chapter 6 discusses the selection of navigation sensors. It covers both the theoretical considerations for this selection, as well as preliminary tests that were carried out with the sensors selected. It concludes with a list of sensors that were selected for the practical investigations.

Chapter 7 deals with the design of a suitable navigation filter for sensor fusion with the selected set of sensors and presents two different integration architectures for the selected sensors: federated and cascaded integration. Moreover, a tailored dynamic model which takes into account specific motion characteristics of tracked vehicles is developed.

Part III describes the practical investigations and results and covers Chapters 8 to 10.

Chapter 8 covers the data collection and describes the tests with a prototype vehicle at a composting site.

In **Chapter 9** the accuracy that can be achieved with the sensors selected and the navigation filters presented is analysed and evaluated. The results and the advantages of a multi-sensor approach are discussed and compared to conventional GNSS/INS integration.

Chapter 10 concludes the thesis and gives an outlook for further research.

Part I

Theoretical Foundations

Chapter 2

Fundamentals of Navigation

The Cambridge Dictionary (2022) defines navigation as *"the act of directing a ship, aircraft, etc. from one place to another, or the science of finding a way from one place to another"*. Hofmann-Wellenhof et al., 2003 divide the tasks of navigation into positioning, routing, and guidance. *Positioning* refers to determining the position or coordinates of a moving object with respect to a reference frame. *Routing* deals with the task of computing an optimal (e.g. shortest or fastest) route from point A to point B and *guidance* aims to guide the moving object along the optimal route.

When navigating, we want to determine the *state vector* of a moving object. The state vector $\mathbf{x}_t$ describes the state of an object at epoch t and consists of the coordinates, velocity, and attitude of the moving object. The path that a moving object follows is referred to as *trajectory*; it consists of a series of state vectors at subsequent epochs.

As this thesis focuses on multi-sensor positioning, this chapter presents basic navigation concepts related to determining the position of a moving object. In the following, first the concepts of position fixing and dead reckoning are introduced in Sections 2.1 and 2.2. Then, the concepts of redundancy and sensor fusion are discussed in Sections 2.3 and 2.4. The chapter ends with Section 2.5, which deals with commonly used reference frames and transformations.

2.1 Position Fixing

All types of positioning techniques can be divided into two categories: methods of position fixing and methods of dead reckoning.

Position fixing, or absolute positioning, refers to positioning techniques that determine the position of an object by measuring to or from points with known coordinates. These measurements can be directions, angles, (pseudo-)ranges, range differences, and range rates (Hofmann-Wellenhof et al., 2003). Each measurement leads to a Line of Position (LOP) in the two-dimensional case or a Surface of Position (SOP) in the three-dimensional case, which is the geometric location of where the object that takes the measurement could be located. By intersecting LOPs or SOPs, the position of an object can be computed. Table 2.1 gives an overview of the LOPs and SOPs for different types of measurements.

Table 2.1: Lines and surfaces of position for different measurements, taken from Hofmann-Wellenhof et al. (2003)

Measurement	**Line of Position (2D)**	**Surface of Position (3D)**
Direction	Straight line	Plane
Angle	Straight line	–
Zenith angle	–	Cone
Range	Circle	Spherical shell
Pseudorange	Circle with biased radius	Spherical shell with biased radius
Range difference	Hyperbolic branch	Hyperbolic shell
Range rate	Straight line	Cone

In the following, some commonly used combinations for position fixing are presented. For the sake of simplicity, they are explained for the two-dimensional case. However, they can also be extended to the third dimension.

Bensky (2016) describes four geometric arrangements for computing the 2D position based on distance (rho) and direction (theta) measurements:

1. Rho-Theta: When an oriented direction (θ_{FT}) and a distance (ρ_{FT}) are measured between a fixed point F with known coordinates and a target T, we can compute the 2D coordinates of the target from

$$\mathbf{x}_T = \mathbf{x}_F + \rho_{FT} \begin{pmatrix} \cos\theta_{FT} \\ \sin\theta_{FT} \end{pmatrix}. \tag{2.1}$$

2. Theta-Theta or Angle Of Arrival (AOA): To determine the 2D coordinates of an unknown point from oriented directions θ_{F_iT}, at least two direction measurements to known points are needed. The LOP of one oriented direction measurement is an oriented line starting at the reference point F_i. The coordinates of the unknown point T can be computed from the intersection of the LOPs.

3. Rho-Rho or Time Of Arrival (TOA): When several distances to known points are measured, the LOPs are circles with the radius ρ_{F_iT} centered at the known points F_i. The coordinates of the unknown point are determined by intersecting these circles. Note that in the two-dimensional case, when two circles intersect, we get two solutions. This ambiguity can be eliminated by either knowing approximate coordinates of the point we want to estimate, or by adding a third distance measurement.

4. Time Difference of Arrival (TDOA) and Hyperbolic Curves: When the clocks between the emitter F_i and receiver T of a radio signal are not perfectly synchronised, we measure a so-called pseudorange R_{F_iT} instead of a range ρ_{F_iT}:

$$R_{F_iT} = \rho_{F_iT} + \Delta\rho, \tag{2.2}$$

   where $\Delta\rho$ is the range error due to imperfect clock synchronisation. One approach to determine the position from measured pseudoranges is hyperbolic positioning. For 2D positioning, three pseudoranges are needed (R_{F_1T}, R_{F_2T}, R_{F_3T}). By computing the range differences for $i = 2, 3$ as follows

$$R_{F_iT} - R_{F_1T} = \rho_{F_iT} - \rho_{F_1T}, \tag{2.3}$$

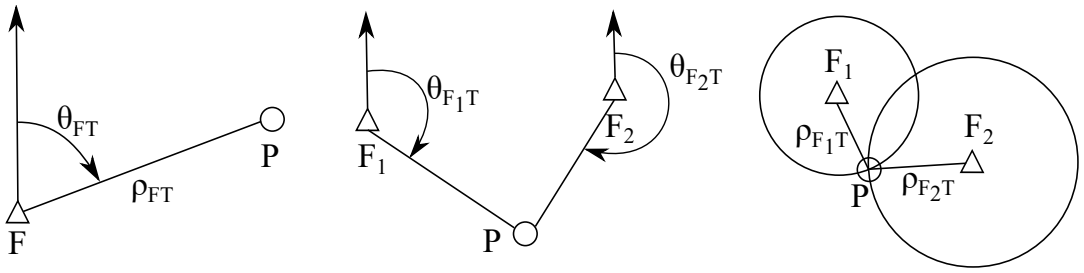


Figure 2.1: Absolute positioning: Rho-Theta, Theta-Theta and Rho-Rho.

the range error $\Delta\rho$ cancels. The LOP of a range difference as in Equation 2.3 is a hyperbola. The coordinates of the receiver T is computed from the intersection of the two hyperbolas.

Groves (2013) and Hofmann-Wellenhof et al. (2003) list another method of absolute positioning: range rate or Doppler positioning. When there is a relative motion between the emitter and the receiver of a signal, the frequency that is measured by the receiver differs from the emitted frequency (Doppler shift). The Doppler shift is proportional to the radial velocity or range rate. For 2D positioning, the LOP of a range rate is an oriented line. For 3D positioning, the SOP of a range rate is a cone, where the position of the emitter is at the apex.

Furthermore, Bensky (2016) lists two methods of absolute positioning that do not belong to the classic geometric categories of absolute positioning, but are nonetheless position fixing methods: proximity and fingerprinting. For proximity, when a radio signal is received, the receiver's position is taken to be within a region around the transmitter's position. Fingerprinting computes the receiver's position by comparing characteristics of the received signals (such as the received signal strength) to a database which contains the same characteristics at known reference points (Reitbauer, 2017).

2.2 Dead Reckoning

Methods of dead reckoning, or relative positioning, compute the position at epoch t from the previous position at epoch $t-1$ and the position change that occurred between these two epochs. The position change can either be obtained from inertial navigation, i.e.

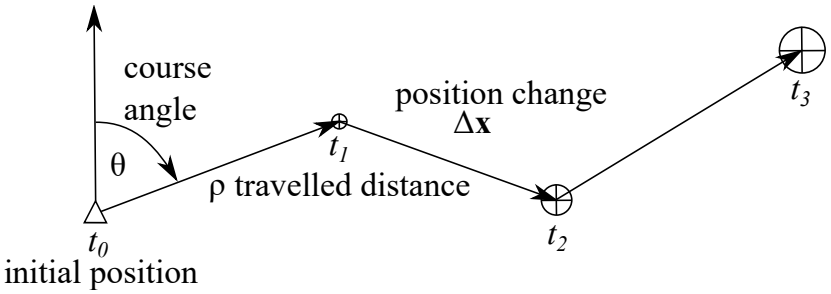


Figure 2.2: Principle of dead reckoning: the new positions are computed from an initial position and either an angle and distance, or a change in position. Adapted from Groves (2013).

from integrating accelerations twice to obtain position changes $\delta\mathbf{x}$ (for the 2D and 3D case), or from a rho-theta technique (for the 2D case) (Hofmann-Wellenhof et al., 2003). The angle ϑ is the course angle, which can be obtained from sensors like a magnetometer or gyrocompass, and ρ is the distance travelled between the two epochs, which can be obtained from odometry.

A characteristic of dead reckoning systems is that errors accumulate and that the position error grows with time or distance travelled. This is illustrated in Figure 2.2, where the error bounds are depicted as circles.

2.3 Redundancy

When more than one navigation sensor is used and more information than needed is available to compute the position of an object, we speak of *redundant* information. Hofmann-Wellenhof et al. (2003) distinguish between four types of redundancy: parallel, complementary, dissimilar and analytical redundancy.

Parallel redundancy occurs when several identical sensors are used. An example would be the use of multiple IMUs (see Bancroft and Lachapelle, 2011).

We speak of *complementary* redundancy when sensors with different operation principles and characteristics are combined, for example one position fixing and one dead reckoning sensor. A commonly used example of this type of redundancy is the combination of GNSS and IMU (see e.g. Noureldin et al., 2013; Wendel, 2011; Jekeli, 2001).

Dissimilar redundancy refers to a situation where several non-identical sensors that do not fully complement each other are combined. This could, for example, be a combination of two different types of position fixing sensors, such as the combination of 5G and GNSS (Destino et al., 2018), or the combination of WiFi Round Trip Time (RTT) and Ultra Wide Band (UWB) (Kia et al., 2021).

Analytical redundancy is based on prior knowledge of the system models. An example would be pose estimation using a stereo camera as the navigation sensor and an already existing point cloud of the environment (Theurl, 2022).

2.4 Sensor Fusion

When two or more navigation sensors are used to estimate the position of a moving object, we speak of an integrated navigation system. The goal of an integrated navigation system is to produce a navigation solution that is superior to that of its individual sensors (Ward, 2009). To achieve this goal, redundant information from multiple sensors can be fused in different ways.

Signal blending can be used in the case of parallel redundancy. When several identical sensors are present, a weighted average of the signals can be computed. An example would be the creation of a Virtual IMU (VIMU) from several individual IMUs (Bancroft and Lachapelle, 2011).

Optimal filters, such as Kalman or particle filters, use dynamic motion models and time-variant stochastic covariance models (Hofmann-Wellenhof et al., 2003). They are commonly used in integrated navigation systems and will be discussed in detail in Chapter 4. Optimal filters can be used to fuse position fixing and dead reckoning sensors, but also to fuse multiple position fixing or multiple dead reckoning sensors.

2.5 Reference Systems and Transformations

A point in space can be defined as a set of three coordinates. The coordinates define where the point lies with respect to a certain reference system. Reference systems

are defined by their origin and orientation. The orientation is usually defined by two orthogonal axes, where the third axis is obtained from the cross product of the first two axes, forming either a right-handed or a left-handed reference system (Hofmann-Wellenhof et al., 2003). Note that there is a difference between reference systems and reference frames: the theoretical definition is called reference or coordinate system, whereas the realization is called reference or coordinate frame. A reference frame consists of a set of physical points with determined coordinates within the reference system.

For terrestrial navigation, the following reference frames are commonly used: equatorial frames, such as celestial and terrestrial equatorial frames, local-level frames, body frames and sensor frames.

Celestial Equatorial System

A celestial equatorial system, in the literature sometimes also called Earth-Centered Space Fixed (ECSF) or Earth-Centered Inertial System (Groves, 2013), is a quasi-inertial reference system and will in the following be denoted with the superscript i. Its origin is the geocenter, its $\mathbf{x}_1^i$-axis points towards the vernal equinox, and its $\mathbf{x}_3^i$-axis is the angular momentum axis of the Earth at a standard epoch (J2000.0). The $\mathbf{x}_2^i$ axis is orthogonal to the other two axes, forming a right-handed coordinate system.

In an inertial system, Newton's laws of motion apply. Inertial systems or frames neither accelerate nor rotate. A celestial equatorial system is only quasi-inertial because its origin, the Earth's geocenter, orbits the sun and is therefore subject to accelerations. Inertial frames are needed in navigation because IMUs measure accelerations and angular rates with respect to an inertial frame.

Terrestrial Equatorial System

A terrestrial equatorial system, also called Earth-Centered Earth Fixed (ECEF), also has its origin in the geocenter and the $\mathbf{x}_3^e$-axis is the Earth's mean rotation axis. The $\mathbf{x}_1^e$-axis points towards the Greenwich meridian in the equatorial plane, which means that the frame rotates with the Earth around its rotation axis. The $\mathbf{x}_2^e$-axis is orthogonal to

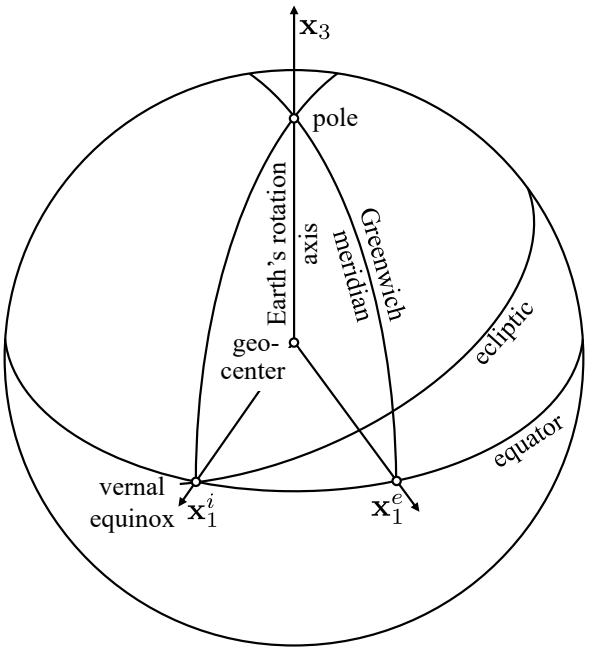


Figure 2.3: Celestial and terrestrial equatorial systems. Adapted from Hofmann-Wellenhof et al. (2008).

the other two axes, forming a right-handed coordinate system. Figure 2.3 shows the $\mathbf{x}_1$- and $\mathbf{x}_3$-axes of terrestrial and celestial equatorial systems.

Terrestrial equatorial frames are important in navigation because GNSS-positions are computed in this frame. GNSS receivers often output positions in geographic or ellipsoidal coordinates φ, λ, and h. The Cartesian coordinates in the terrestrial equatorial frame can be obtained from (Hofmann-Wellenhof et al., 2003)

$$\mathbf{x}^e = \begin{pmatrix} x_1^e \\ x_2^e \\ x_3^e \end{pmatrix} = \begin{pmatrix} (N+h)\cos\varphi\cos\lambda \\ (N+h)\cos\varphi\sin\lambda \\ \left(\frac{b^2}{a^2}N+h\right)\sin\varphi \end{pmatrix}, \tag{2.4}$$

with

$$N = \frac{a^2}{\sqrt{a^2\cos^2\varphi + b^2\sin^2\varphi}}, \tag{2.5}$$

where a and b are the semi-major and semi-minor axes of the ellipsoid and N is the radius of curvature of the prime vertical.

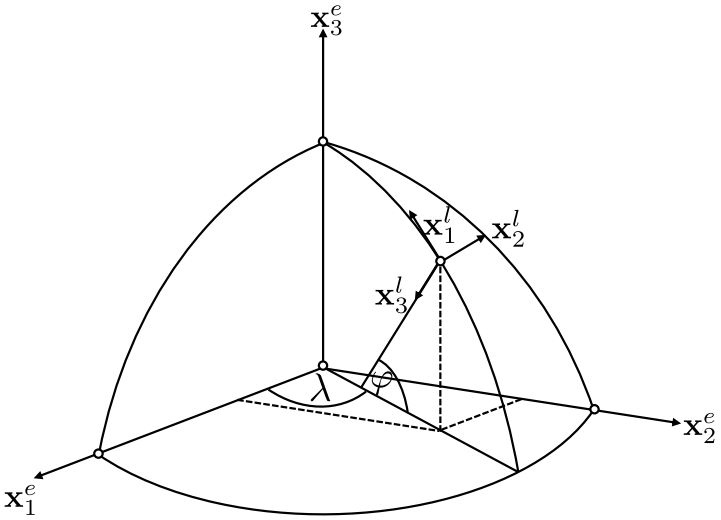


Figure 2.4: Right-handed local-level system and terrestrial equatorial system. Adapted from Hofmann-Wellenhof et al. (2003).

Local-Level System

A local-level system has its origin close to or on the Earth's surface. The $\mathbf{x}_1^l$-axis points towards the North, the $\mathbf{x}_2^l$-axis towards the East, and the $\mathbf{x}_3^l$-axis points either towards the nadir, forming a right-handed frame, or towards the zenith, forming a left-handed system. Figure 2.4 shows the origin and axes of a right-handed local-level system and its relation to a terrestrial equatorial system.

Note that we can distinguish between two types of local-level systems: local tangent-plane systems and local navigation systems (Groves, 2013). A local tangent-plane frame has a fixed origin with respect to the Earth's surface. In a local navigation frame, the origin is tied to the object that we compute the navigation solution for.

Body Frame

A body frame is a frame that is fixed to a moving object. As with a local navigation frame, the origin is situated within the object, usually the center of mass. The $\mathbf{x}_1^b$-axis points forward, the $\mathbf{x}_3^b$-axis down, and the $\mathbf{x}_2^b$-axis to the right.

Sensor Frame

A sensor frame is a frame that is tied to a navigation sensor, e.g. a camera, LiDAR or IMU. When several navigation sensors are used, the lever arms between the origins of the individual sensor frames and the body frame, as well as the orientation of the axes of the sensor frame with respect to the body frame, must be considered.

2.5.1 Transformations

To transform between two coordinate frames with different origins and orientations of axes, a Helmert or seven-parameter transformation can be used (Hofmann-Wellenhof et al., 2008). To transform from a p-frame to a q-frame, where $\mathbf{x}^p$ are the three-dimensional Cartesian coordinates in the p-frame and $\mathbf{x}^q$ the three-dimensional coordinates in the q-frame, the transformation is described by

$$\mathbf{x}^q = \mathbf{c} + \mu \mathbf{R}_p^q \mathbf{x}^p, \tag{2.6}$$

where $\mathbf{c}$ stands for the translation (or the coordinates of the origin of the p-frame expressed in the q-frame), $\mathbf{R}_p^q$ is the rotation matrix from the p-frame to the q-frame, and μ is the scale factor. Note that the scale factor is needed when transferring between geodetic datums, e.g. a datum with a local and a datum with a global reference ellipsoid.

When transforming between coordinate frames with the same origin or when only difference vectors are transformed, the orthogonal rotation matrix $\mathbf{R}_p^q$ (with a determinant of 1) is sufficient. Note that since rotation matrices are orthogonal, the inverse rotation from the q-frame to the p-frame is expressed by the transpose of the matrix $\mathbf{R}_p^q$, i.e.:

$$\mathbf{R}_q^p = \mathbf{R}_p^{q\ \mathrm{T}}. \tag{2.7}$$

A three-dimensional rotation between two frames can be described by three sequential rotations about the axes of the first frame. A commonly used transformation is the rotation from the local-level to the body frame:

$$\mathbf{R}_l^b = \mathbf{R}_1(\theta)\mathbf{R}_2(\phi)\mathbf{R}_3(\psi), \tag{2.8}$$

where θ, ϕ, ψ stand for the Euler angles roll, pitch, and yaw. The individual rotation matrices $\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3$ about the $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$- axes of the l-frame are defined as:

$$\begin{aligned} \mathbf{R}_1(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{pmatrix}, \mathbf{R}_2(\phi) = \begin{pmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{pmatrix}, \\ \mathbf{R}_3(\psi) = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}. \end{aligned} \tag{2.9}$$

The rotation between two frames is also called attitude and the Euler angles roll (θ), pitch (ϕ), and yaw (ψ) are often referred to as the attitude parameters.

Quaternion Notation

According to Euler's rotation theorem, the rotation may also be expressed as a single rotation by the angle α about the axis $\boldsymbol{n}$. This is used when we represent the attitude using quaternions. A quaternion q consists of a scalar q_0 and a vector $\mathbf{q} = (q_1, q_2, q_3)^T$. Quaternions are multiplied as follows (Giorgi, 2017):

$$\mathrm{qq}' = q_0 \begin{pmatrix} q_0' \\ \mathbf{q}' \end{pmatrix} + \begin{pmatrix} 0 & -\mathbf{q}^T \\ \mathbf{q} & [\mathbf{q}^+] \end{pmatrix} \begin{pmatrix} q_0' \\ \mathbf{q}' \end{pmatrix}, \tag{2.10}$$

with

$$\left[\boldsymbol{q}^+\right] = \begin{pmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{pmatrix}. \tag{2.11}$$

Quaternions with a norm of $||\mathrm{q}|| = 1$ represent a rotation. The scalar q_0 depends on the magnitude of the rotation and the vectorial part $\mathbf{q}$ is linked to both the magnitude and

the rotation axis (Groves, 2013). A rotation by the angle α about the axis $\boldsymbol{n}$ can be expressed as (Giorgi, 2017):

$$q(\alpha, \boldsymbol{n}) = \begin{pmatrix} \cos\left(\frac{\alpha}{2}\right) \\ \sin\left(\frac{\alpha}{2}\right) \boldsymbol{n} \end{pmatrix}. \tag{2.12}$$

The quaternion representation of the matrix $\mathbf{R}_l^b$ in Equation 2.8 is:

$$\begin{aligned} &\mathbf{R}_l^b(\mathrm{q}) = \left(q_0^2 - \boldsymbol{q}^\top \boldsymbol{q}\right) \mathbf{I}_3 + 2\boldsymbol{q}\boldsymbol{q}^\top - 2q_0 \left[\boldsymbol{q}^+\right] = \\ &\begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2\left(q_1 q_2 + q_0 q_3\right) & 2\left(q_1 q_3 - q_0 q_2\right) \\ 2\left(q_1 q_2 - q_0 q_3\right) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2\left(q_2 q_3 + q_0 q_1\right) \\ 2\left(q_1 q_3 + q_0 q_2\right) & 2\left(q_2 q_3 - q_0 q_1\right) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix}. \end{aligned} \tag{2.13}$$

The quaternion notation in Equation 2.13 is sometimes preferred to the Euler angle notation in Equation 2.8, as it is not affected by a phenomenon known as gimbal lock. Gimbal lock is a geometric problem that occurs for Euler angles when the axis of the first and of the third rotation align. In this case, the sum of the first and third rotation angle perform a rotation around a common axis, and a degree of freedom is lost.

Another commonly used rotation matrix in navigation describes the rotation from the local-level frame to the ECEF-frame. It can be expressed as a function of the parameters φ and λ, which are the ellipsoidal coordinates of the origin of the local-level frame expressed in the ECEF-frame:

$$\mathbf{R}_l^e = \begin{pmatrix} -\sin\varphi\cos\lambda & -\sin\lambda & -\cos\varphi\cos\lambda \\ -\sin\varphi\sin\lambda & \cos\lambda & -\cos\varphi\sin\lambda \\ \cos\varphi & 0 & -\sin\varphi \end{pmatrix}. \tag{2.14}$$

Homogeneous Transformation Matrix

In robotics and for image-based navigation, homogeneous coordinates and homogeneous transformation matrices are commonly used. The homogeneous coordinates of a point $\mathbf{x}^p$ in the p-frame are represented by the 4×1 vector

$$\mathbf{p}^p = \begin{pmatrix} x_1^p \\ x_2^p \\ x_3^p \\ 1 \end{pmatrix}, \tag{2.15}$$

where x_1^p, x_2^p, x_3^p are the Cartesian coordinates of the point $\mathbf{x}^p$ and 1 in the fourth component is a scaling factor (Briot and Khalil, 2015). A direction or difference vector also consists of four components, where the entry in the fourth component is equal to zero.

To transform the vector $\mathbf{p}^p$ from the p-frame to the q-frame, we introduce the homogeneous transformation matrix $\mathbf{T}_p^q$:

$$\mathbf{p}^q = \mathbf{T}_p^q \mathbf{p}^p. \tag{2.16}$$

The 4×4-matrix $\mathbf{T}_p^q$ has the form

$$\mathbf{T}_p^q = \begin{pmatrix} r_{11} & r_{12} & r_{13} & c_1 \\ r_{21} & r_{22} & r_{23} & c_2 \\ r_{31} & r_{32} & r_{33} & c_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} & \mathbf{R}_p^q & & \mathbf{c} \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{2.17}$$

where $\mathbf{R}_p^q$ is the 3×3 rotation matrix from the $p-$ to the $q-$frame and $\mathbf{c}$ is the translation vector. An important property of homogeneous transformation matrices is that the total transformation can be described as the multiplication of consecutive transformations:

$$\mathbf{T}_k^0 = \prod_{i=1}^{k} \mathbf{T}_i^{i-1} = \mathbf{T}_1^0 \cdot \mathbf{T}_2^1 \cdot \mathbf{T}_3^2 \cdots \mathbf{T}_k^{k-1}. \tag{2.18}$$

This representation is often used when dead reckoning is performed with image-based sensors, e.g. when Visual Odometry (VO) is used. For more details on homogeneous coordinates and transformations the reader is referred to Briot and Khalil (2015).

Chapter 3

Navigation Sensors and Techniques

While the previous chapter introduced basic concepts of navigation, this chapter now focuses on commonly used navigation sensors and navigation techniques. Section 3.1 starts with the basic principles of GNSS and then explains the system architecture, the signal structure and observables, as well as high-precision positioning techniques. Here, a focus is placed on differential carrier phase positioning and also attitude determination using GNSS baselines, as these techniques are applied later in this thesis. Section 3.2 introduces the concept of inertial navigation. It explains the measurement principle of inertial sensors, explains sensor errors and the strapdown navigation technique. Section 3.3 presents odometry, a navigation technique for land vehicles. It discusses different types of wheel speed sensors and presents a model for linear and differential odometry for tracked vehicles. Section 3.4 focuses on image-based navigation. It presents different imaging sensors and discusses two techniques for image-based navigation, namely stereo visual odometry and Normal Distributions Transform (NDT).

3.1 Global Navigation Satellite Systems

Positioning with GNSS belongs to the category of position fixing techniques. The basic principle is that a receiver measures the run-time of an electromagnetic signal (one-way ranging) as it travels from the satellite s to the receiver r. The signal run-time is

multiplied by the propagation speed c to obtain a pseudorange measurement R_r^s :

$$R_r^s = \rho_r^s + \Delta\rho = \sqrt{(x^s - x_r)^2 + (y^s - y_r)^2 + (z^s - z_r)^2} + c \cdot \delta t, \tag{3.1}$$

where ρ_r^s is the true geometric distance between receiver and satellite and δt is the receiver clock error. x^s, y^s, and z^s are the coordinates of the satellite, and x_r, y_r and z_r are the coordinates of the receiver. Since the satellite also broadcasts its ephemeris data, we can compute the position of the satellite. Equation 3.1 therefore has four unknowns: the three receiver coordinates and the receiver clock error. When at least four pseudorange measurements are available, we can compute our receiver's position with a Rho-Rho technique. In the three-dimensional case, the SOP of a pseudorange measurement is a sphere. Note that the spheres of the four pseudorange measurements do not perfectly intersect in one point. Geometrically, the solution lies at the centre of a sphere that is tangent to the four SOPs of the pseudoranges (Hofmann-Wellenhof et al., 2008). The radius of this sphere is the range error.

3.1.1 System Architecture

Nowadays, there are four GNSS: USA's GPS, Russia's GLONASS, the European Galileo and China's BeiDou. Each GNSS has two segments: a space segment, which comprises the satellite constellation, and a ground segment, which monitors and controls the satellites. In the following, the two segments will be briefly described, using Galileo as an example.

Galileo's nominal constellation consists of 30 satellites at an altitude of 23 222 km. The satellites are placed in 3 orbital planes with an inclination of 56°. Of these 30 satellites, 24 are planned to be active and 6 spare. Each satellite carries four atomic clocks: two rubidium atomic frequency standards and two passive hydrogen masers (Falcone et al., 2017).

Galileo's ground segment consists of a Ground Control Segment (GCS) and a Ground Mission Segment (GMS). They are controlled via two control centers in Oberpfaffenhofen, Germany and Fucino, Italy. The GCS consists of worldwide network of six telemetry, tracking and control stations. The GMS monitors the navigation signals, computes

the navigation message and sends it to the satellites. For this purpose, continuous measurements are conducted at the Galileo sensor stations. The navigation messages are sent to the satellites via the five mission uplink stations.

3.1.2 Signal Structure and Observables

GNSS satellites broadcast electromagnetic signals in a frequency spectrum between 1.2 and 1.6 GHz (Langley et al., 2017). All satellites send signals on at least two different frequencies so that ionospheric delays can be compensated for when the signals are processed.

For ranging measurements, a so-called Pseudorandom Noise (PRN) code is modulated onto the carrier waves. It is a binary sequence transmitted at a rate of 1-10 MHz and is repeated every few milliseconds to seconds, depending on the length of the sequence. Most GNSS also have a unique PRN for each satellite. When satellites are distinguished depending on the PRN, we speak of Code Division Multiple Access (CDMA). GLONASS uses different carrier frequencies to distinguish between satellites, a principle which is called Frequency Division Multiple Access (FDMA) (Hofmann-Wellenhof et al., 2008).

In addition to the PRN, a navigation message is modulated onto the carrier wave. This navigation message contains the broadcast ephemerides, i.e. the information needed to compute the orbit of the satellite, as well as information on the satellite clock error.

A GNSS receiver internally produces a replica of the signal by modulating the known PRN onto the carrier wave. The replica generated is correlated with the incoming signal in a tracking loop that aligns the replicated signal with the incoming signal. Through this process, the transmission time of the incoming signal is determined. The transmission time is compared to the receiver time to obtain the signal propagation time, which is multiplied by the speed of light c to obtain the pseudorange.

There are three basic types of GNSS observables: pseudorange, Doppler, and carrier phase. The pseudorange is obtained as described in the paragraph above and has decimeter-precision. The Doppler shift is the change in the frequency received due to the Doppler effect. It is proportional to the radial velocity or range rate and therefore often used in navigation to determine the velocity of a moving object. The carrier phase

is a measurement of the instantaneous beat phase and the number of integer cycles since the initial epoch. Carrier phase measurements are about two orders of magnitude more precise than pseudorange measurements (Langley et al., 2017). However, since only the fractional beat phase can be measured when the receiver is switched on, the integer number of cycles from the satellite to the receiver is unknown. This so-called integer ambiguity remains constant when the receiver continuously tracks the signal and counts the accumulated cycles without interruption.

3.1.3 High-Precision GNSS

When electromagnetic signals travel from the satellites to the receivers on Earth, they are influenced by several additional error sources. The basic observation equation of the code pseudorange (Equation 3.1) can therefore be extended to (Hauschild, 2017)

$$\begin{aligned} R_r^s(t) =& \rho_r^s(t) + c\left(\delta t_r(t) - \delta t^s(t) + \delta t^{\text{rel}}(t)\right) + I_r^s(t) + T_r^s(t) \\ &+ \xi_r^s(t) + c\,(d_r - d^s) + e_r^s(t), \end{aligned} \tag{3.2}$$

where δt_r is the receiver clock error, δt^s is the satellite clock error, δt^{rel} is the relativistic correction, I_r^s the ionospheric propagation delay, T_r^s is the tropospheric propagation delay, ξ_r^s is a correction term due to phase-center offsets of the satellite's and receiver's antenna, and d_r and d_s are instrumental delays of receiver and satellite. The term e_r^s contains additional errors like multipath and receiver noise. For details on these errors and propagation delays, the reader is referred to Hauschild (2017) and Sanz Subirana et al. (2013).

For a very simple Single Point Positioning (SPP), the satellite coordinates, satellite clock error, ionospheric propagation delay can be computed from the navigation message and the tropospheric propagation delay can be modelled. For high-precision GNSS, these terms need to be modelled with higher accuracy (or eliminated through differencing) and also the other terms in Equation 3.2 have to be taken into consideration.

The observation equation for the carrier phase (expressed in metres) reads as follows (Hauschild, 2017):

$$\begin{aligned}\varphi_r^s(t) =& \rho_r^s(t) + c\left(\delta t_r(t) - \delta t^s(t) + \delta t^{\text{rel}}(t)\right) - I_r^s(t) + T_r^s(t) \\ &+ \lambda\left(\omega_r^s(t) + N_r^s\right) + \zeta_r^s(t) + c\left(\delta_r - \delta^s\right) + \epsilon_r^s(t),\end{aligned} \tag{3.3}$$

where λ is the wavelength of the carrier wave, ω_r^s is the phase wind-up, and N_r^s are the integer ambiguities. Note that in this notation, the phase wind-up and the ambiguities are negative values, as they are multiplied by $+\lambda$. ζ_r^s is the phase center offset, δ_r and δ^s are the instrumental delays of the satellite and receiver; since these three terms are different for code pseudorange and phase, they are denoted with different symbols. Note that the magnitude of the ionospheric correction term is the same for code pseudorange and phase, but the signs are different (the code is delayed as it travels through the ionosphere while the carrier wave is advanced). The residual error term ϵ_r^s contains effects such as multipath and tracking noise.

Precise Point Positioning

One method to obtain highly precise GNSS positions is the so-called Precise Point Positioning (PPP), which is a more sophisticated version of the classical SPP. In contrast to SPP which uses code pseudoranges and simple error models, PPP mainly relies on the more precise carrier phase measurements and highly precise models. Highly precise satellite orbits and clock errors can be obtained from the International GNSS Service (IGS). Through a linear combination of dual-frequency measurements, first-order ionospheric effects can be removed. The tropospheric effect is modelled, and the residual tropospheric delay as well as the integer ambiguities are estimated. Satellite and receiver antenna offsets, phase wind-up, as well as Earth tides and ocean loading are also modelled (Langley et al., 2017). PPP can achieve decimeter accuaracies for the positioning of moving platforms. A drawback is the convergence time: it takes about 20-30 minutes for the positioning solution to converge and achieve decimeter-accuracy.

Differential Carrier Phase Positioning

Another precise positioning technique using GNSS is differential carrier phase positioning. This technique combines carrier phase observations at a reference station with known coordinates with those taken at a rover. The basic concept of differential carrier phase positioning is to precisely determine the baseline between the reference station and the rover. When the coordinates $\mathbf{X_A}$ of the reference station are known and the baseline vector $\mathbf{b_{AB}}$ is determined, the coordinates of the rover $\mathbf{X_B}$ can be computed from (Hofmann-Wellenhof et al., 2008):

$$\mathbf{X_B} = \mathbf{X_A} + \mathbf{b_{AB}}. \tag{3.4}$$

The baseline is determined through differencing the carrier phase measurements taken at the positions of the rover and the reference station. Differencing the observations has the advantage that error sources are eliminated.

For single differences, simultaneous carrier phase observations on the same satellite are differenced between the reference station and the rover. We use a simplified model (Langley et al., 2017) for the carrier phase observations taken at reference station A and rover B on the satellite s:

$$\varphi^{\mathrm{s}}_{\mathrm{A}} = \rho^{\mathrm{s}}_{\mathrm{A}} + c\,(\delta t_{\mathrm{A}} - \delta t^{\mathrm{s}}) + T^{\mathrm{s}}_{\mathrm{A}} - I^{\mathrm{s}}_{\mathrm{A}} + \lambda M^{\mathrm{s}}_{\mathrm{A}} + \epsilon^{\mathrm{s}}_{\mathrm{A}}, \tag{3.5}$$

$$\varphi^{\mathrm{s}}_{\mathrm{B}} = \rho^{\mathrm{s}}_{\mathrm{B}} + c\,(\delta t_{\mathrm{B}} - \delta t^{\mathrm{s}}) + T^{\mathrm{s}}_{\mathrm{B}} - I^{\mathrm{s}}_{\mathrm{B}} + \lambda M^{\mathrm{s}}_{\mathrm{B}} + \epsilon^{\mathrm{s}}_{\mathrm{B}}, \tag{3.6}$$

where $\rho^{\mathrm{s}}_{(\cdot)}$ is the geometric distance between the receiver and satellite s, $\delta t_{(\cdot)}$ is the receiver clock error and δt^s the satellite clock error, $T^{\mathrm{s}}_{(\cdot)}$ is the tropospheric and $I^{\mathrm{s}}_{(\cdot)}$ the ionospheric error. λ is the wavelength of the carrier wave and the term $M^{\mathrm{s}}_{(\cdot)} = N^{s}_{(\cdot)} + \delta_{(\cdot)} - \delta^s$ contains the integer ambiguities $N^{s}_{(\cdot)}$, as well as the instrumental receiver and satellite phase delays expressed in cycles. Note that here, the term $M^{\mathrm{s}}_{(\cdot)}$ is a negative value as it is multiplied by $+\lambda$. The term $\epsilon^{\mathrm{s}}_{(\cdot)}$ contains errors like receiver noise and multipath. Subtracting Equation 3.5 from Equation 3.6 yields the single difference:

$$\varphi^{s}_{AB} = \rho^{s}_{AB} + c\delta t_{AB} + T^{s}_{AB} - I^{s}_{AB} + \lambda M^{s}_{AB} + \epsilon^{s}_{AB}, \tag{3.7}$$

where $(\cdot)_{AB} = (\cdot)_B - (\cdot)_A$. Note that the satellite clock error was eliminated through differencing as it was the same for both phase observations. Furthermore, the term $M^s_{AB} = N^s_{AB} + \delta_{AB}$ no longer contains satellite phase delays. For short baselines of less than 50 km and when the receivers are at the same altitude, also the ionospheric and tropospheric delays cancel out (Kaplan and Hegarty, 2006) and Equation 3.7 is further reduced to:

$$\varphi^s_{AB} = \rho^s_{AB} + c\delta t_{AB} + \lambda M^s_{AB} + \epsilon^s_{AB}. \tag{3.8}$$

The geometric interpretation of a single difference is illustrated in Figure 3.1. The vector $\mathbf{e}_s$ is the unit vector from the receivers to the satellite. Since, in the case of Galileo, the satellite s is 23 222 km away from our receivers A and B (20 200 km in the case of GPS), we can assume that the signal propagation paths to A and B are parallel. Differencing the phase observations yields the single difference φ^s_{AB}. Geometrically, this can be interpreted as the projection of the baseline vector onto the line of sight vector to the satellite (Kaplan and Hegarty, 2006). The single difference can therefore also be written as the dot product of the baseline $\mathbf{b_{AB}}$ and the unit vector to the satellite:

$$\varphi^s_{AB} = (\mathbf{b_{AB}} \cdot \mathbf{e}^s). \tag{3.9}$$

To form a double difference, two single differences from two different satellites s and t are differenced. We therefore introduce the single difference to satellite t

$$\varphi^t_{AB} = \rho^t_{AB} + c\delta t_{AB} + \lambda M^t_{AB} + \epsilon^t_{AB} \tag{3.10}$$

and subtract Equation 3.8 from Equation 3.10 to obtain the double difference

$$\varphi^{st}_{AB} = \rho^{st}_{AB} + \lambda N^{st}_{AB} + \epsilon^{st}_{AB}. \tag{3.11}$$

Note that now the receiver-dependent biases such as the receiver clock error and the receiver phase bias have cancelled. This has the advantage that we no longer need a real-valued parameter M, but now have integer ambiguities λN^{st}_{AB}.

Coming back to the geometric interpretation, the two single differences to satellites s and t can be written as $\varphi^s_{AB} = (\mathbf{b_{AB}} \cdot \mathbf{e}^s)$ and $\varphi^t_{AB} = (\mathbf{b_{AB}} \cdot \mathbf{e}^t)$. The double difference

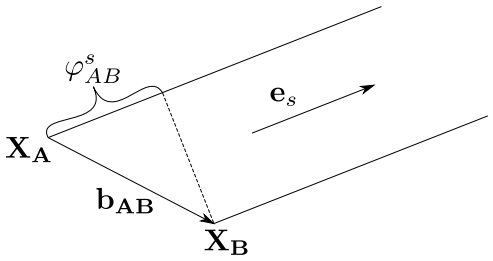


Figure 3.1: The single difference measurement is the projection of the baseline vector onto the line of sight vector to the satellite.

φ^{st}_{AB} can therefore also be written as

$$\varphi^{st}_{AB} = (\mathbf{b_{AB}} \cdot \mathbf{e}^{st}), \tag{3.12}$$

where $\mathbf{e}^{st}$ is the difference of the unit vectors to the satellites s and t.

When double differences are processed, both the combined integer ambiguities and the baseline vector are estimated. The observation equation for four double differences can be written as (Kaplan and Hegarty, 2006)

$$\begin{pmatrix} \varphi^{12}_{AB} \\ \varphi^{13}_{AB} \\ \varphi^{14}_{AB} \\ \varphi^{15}_{AB} \end{pmatrix} = \begin{pmatrix} e^{12}_x & e^{12}_y & e^{12}_z \\ e^{13}_x & e^{13}_y & e^{13}_z \\ e^{14}_x & e^{14}_y & e^{14}_z \\ e^{15}_x & e^{15}_y & e^{15}_z \end{pmatrix} \begin{pmatrix} b_{AB,x} \\ b_{AB,y} \\ b_{AB,z} \end{pmatrix} + \lambda \begin{pmatrix} N^{12}_{AB} \\ N^{13}_{AB} \\ N^{14}_{AB} \\ N^{15}_{AB} \end{pmatrix}, \tag{3.13}$$

where φ^{12}_{AB} is the double difference including the satellites 1 and 2, e^{12}_x the x-component of the differenced unit vectors to the satellites (and the subscripts y and z refer to the corresponding y- and z-component), $b_{AB,x}$ is the x-component of the baseline vector, λ is the wavelength, and N^{12}_{AB} the combined integer ambiguity term. The superscripts $13, 14, 15$ refer to the double differences including satellites 1 and 3, 1 and 4, 1 and 5, respectively.

In matrix notation, Equation 3.13 can be written as

$$\boldsymbol{\varphi}_{AB} = \mathbf{H}\mathbf{b}_{AB} + \lambda \mathbf{N}_{AB}. \tag{3.14}$$

Equations 3.13 and 3.14 can be easily extended to more than four double differences. In the case of n double differenced observations, the vector $\boldsymbol{\varphi}_{AB}$ is a $n \times 1$ column vector, the matrix $\mathbf{H}$ has the dimensions $n \times 3$, the baseline vector $\mathbf{b}_{AB}$ is a 3×1 column vector, λ is a scalar and $\mathbf{N}_{AB}$ is a $n \times 1$ vector containing the integer ambiguities.

Equation 3.14 has more unknowns (the baseline coordinates and the integer ambiguities) than observations. For a number of $n + 1$ satellites, n double differences can be formed (left side of the equation). On the right side of the equation we have three unknown baseline coordinates plus n unknown ambiguities. We can therefore never resolve the baseline and ambiguities using only a single epoch of observations. However, as long as no cycle slips occur, the ambiguities remain constant and can be resolved with several epochs of observations.

Initially, the ambiguities are estimated as floating-point numbers before they are fixed to integer numbers, e.g. with the Least-squares AMBiguity Decorrelation Adjustment (LAMBDA) method by Teunissen (1995). When the ambiguities are fixed, we can determine the baseline coordinates with high accuracy (a few centimetres while the rover is moving). Modern receivers need a few tens of seconds of observations to fix ambiguites (Langley et al., 2017).

It is interesting to note that to determine the baseline between two antennas accurately, it is not necessary to know the exact position of the reference antenna with high accuracy. According to Petovello (2011), the error that imperfect reference coordinates cause to the baseline can be approximated with

$$\varepsilon = |\delta \mathbf{X_A}| \cdot |\mathbf{b_{AB}}| \cdot 10^{-8}, \tag{3.15}$$

where $\delta \mathbf{X_A}$ is the coordinate error of the reference antenna and $\mathbf{b_{AB}}$ is the baseline vector between the two antennas. This means that if the position of a reference antenna is only roughly known with an error of 100 metres and the baseline between the two antennas is 10 km long, only 1 cm of error is introduced. This also allows determining attitude using GNSS (see subsection 3.1.4). When GNSS antennas are placed on a moving platform, i.e. both the reference and the rover are moving, then the baseline between the antennas can be determined with a high accuracy even if we only have a rough estimate of the position of the reference (e.g. from SPP).

Real-Time Kinematic Positioning

When differential carrier-phase positioning is carried out in real-time, we speak of Real Time Kinematic (RTK) positioning. For RTK, the reference station transmits data via the Internet, e.g. with the Network Transportation of RTCM Internet Protocol (NTRIP). Often not only a single reference station but a whole reference station network is used (network RTK). This has the advantage that the rover can be farther away from the reference station. Instead of 10-20 km baseline distance for a single reference station, the stations of a reference network can be placed up to 100-200 km apart (Odijk and Wanninger, 2017).

Network-RTK requires several processing steps. In a first processing step, the ambiguities within the reference station network are fixed. Then, correction coefficients are estimated within the network to model distance-dependent biases. A commonly used technique for this is using area correction parameters (also known as FKP, the abbreviation for the German term *Flächenkorrekturparameter*). Here, the biases are split into dispersive (ionospheric) and non-dispersive errors. For every satellite, both a dispersive and non-dispersive correction parameter in North-South as well as East-West direction is estimated. Dispersive errors have larger short-term variations and are therefore updated more frequently (e.g. every 10 seconds), while non-dispersive errors (tropospheric or orbital errors) are updated approximately every 60 seconds (Odijk and Wanninger, 2017). When the rover sends its approximate position to the reference station network, it receives correction data from a so-called Virtual Reference Station (VRS) close to the rover. The VRS data are computed from the real data observed at the closest reference station and the distance-dependent biases. The rover receiver can then process the VRS data in the same way as it would process data from a single reference station.

PPP-RTK

Another method of high-precision GNSS is PPP-RTK, a combination of PPP and network RTK. Here, the individual error components are determined using the RTK network and sent to the receiver using a State-Space Representation (SSR) (Teunissen and Montenbruck, 2017). This differs from network RTK, where combined corrections or area

control parameters are determined and used to compute VRS observations. PPP-RTK allows for faster ambiguity fixing and higher accuracies than conventional PPP.

3.1.4 Attitude Determination with GNSS

GNSS can also be used to determine the attitude of a moving object. When GNSS antennas are mounted on a rigid body (e.g. a vehicle or airplane), the coordinates of the antennas and the baselines between them are known in the body frame. Through differential carrier phase positioning, the baseline vectors between the antennas can be determined in an ECEF-frame and brought to a local-level frame (Kaplan and Hegarty, 2006) with the rotation matrix $\mathbf{R}_e^l = \mathbf{R}_l^{e\ \mathrm{T}}$ (for $\mathbf{R}_l^e$, see Equation 2.14). To determine the full attitude of a moving object, at least three GNSS antennas or two non-parallel baselines are needed.

The known baseline coordinates in the body frame are written in the columns of matrix $\mathbf{B}$:

$$\mathbf{B} = \begin{pmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \cdots & \mathbf{b}_m \end{pmatrix}, \tag{3.16}$$

where each baseline $\mathbf{b}_i$ is a p-component vector and m is the number of baselines. The parameter $p = 1$ for aligned baselines, $p = 2$ for noncollinear but coplanar baselines and $p = 3$ for noncoplanar baselines (Giorgi, 2017). p defines the rank of matrix $\mathbf{B}$.

The coordinates of the baselines in the local-level frame can also be written in a matrix $\mathbf{L}$:

$$\mathbf{L} = \begin{pmatrix} \mathbf{l}_1 & \mathbf{l}_2 & \cdots & \mathbf{l}_m \end{pmatrix}, \tag{3.17}$$

where each baseline $\mathbf{l}_i$ is a 3×1 column vector.

Furthermore, we introduce the generalized, orthonormal rotation matrix $\mathbf{R}$ with the dimension $3 \times p$. The set $S\mathcal{O}(3, p) \subset \mathbb{R}^{3\times p}$ contains the proper rotation matrices with positive determinants. This generalization is needed to also estimate attitude information from only a single baseline. The rotation matrix $\mathbf{R}$ links the estimated baseline coordinates in the local-level frame to the baseline coordinates in the body frame:

$$\hat{\mathbf{L}} = \mathbf{R}\mathbf{B} + \mathbf{\Theta}, \tag{3.18}$$

where $\mathbf{\Theta}$ is the baseline estimation error (Giorgi, 2017).

To estimate the rotation matrix $\mathbf{R}$, different methods exist. The simplest method, three-axis attitude determination (Wertz, 1978), uses three GNSS antennas or two non-parallel baselines. The estimated baselines in the local-level frame are put in a matrix

$$\hat{\mathbf{L}} = \begin{pmatrix} \hat{\mathbf{l}}_1 & \hat{\mathbf{l}}_2 \end{pmatrix}. \tag{3.19}$$

The same goes for the known baselines in the body frame:

$$\mathbf{B} = \begin{pmatrix} \mathbf{b}_1 & \mathbf{b}_2 \end{pmatrix}. \tag{3.20}$$

Using the above baselines, two orthonormal bases are formed as follows (Giorgi, 2017):

$$\mathbf{v}_1 = \frac{\hat{\mathbf{l}}_1}{\left\|\hat{\mathbf{l}}_1\right\|}, \quad \mathbf{u}_1 = \frac{\mathbf{b}_1}{\|\mathbf{b}_1\|}, \tag{3.21}$$

$$\mathbf{v}_2 = \frac{\hat{\mathbf{l}}_1 \times \hat{\mathbf{l}}_2}{\left\|\hat{\mathbf{l}}_1 \times \hat{\mathbf{l}}_2\right\|}, \quad \mathbf{u}_2 = \frac{\mathbf{b}_1 \times \mathbf{b}_2}{\|\mathbf{b}_1 \times \mathbf{b}_2\|}, \tag{3.22}$$

$$\mathbf{v}_3 = \mathbf{v}_1 \times \mathbf{v}_2, \quad \mathbf{u}_3 = \mathbf{u}_1 \times \mathbf{u}_2. \tag{3.23}$$

The rotation matrix can then be computed from

$$\hat{\mathbf{R}} = \sum_{i=1}^{3} \mathbf{v}_i \mathbf{u}_i^\top. \tag{3.24}$$

The three-axis attitude determination method only works for two baselines. To extend the problem to any number of baselines, we need to formulate the estimation of $\mathbf{R}$ as a least-squares problem. We introduce the estimation error

$$\Xi = \hat{\mathbf{L}} - \mathbf{R}\mathbf{B}, \tag{3.25}$$

and aim to minimize its squared weighted norm, i.e. to find the rotation matrix $\mathbf{R}$ in the set $S\mathcal{O}(3,p)$ that minimizes (argmin) the following condition:

$$\hat{\mathbf{R}} = \underset{\mathbf{R}\in S\mathcal{O}(3,p)}{\operatorname{argmin}} \| \operatorname{vec}(\boldsymbol{\Xi})\|^2_{\mathbf{Q}_{\hat{\mathbf{L}}\hat{\mathbf{L}}}} = \underset{\mathbf{R}\in S\mathcal{O}(3,p)}{\operatorname{argmin}} \left(\operatorname{vec}(\boldsymbol{\Xi})^\top \mathbf{Q}^{-1}_{\hat{\mathbf{L}}\hat{\mathbf{L}}} \operatorname{vec}(\boldsymbol{\Xi}) \right), \tag{3.26}$$

where the operator vec stacks the columns of a matrix to a vector, and $\mathbf{Q}_{\hat{\mathbf{L}}\hat{\mathbf{L}}}$ is the variance-covariance matrix of the estimated baselines $\hat{\mathbf{L}}$ in the local-level frame. There are several approaches to minimize the residuals in Equation 3.26. For detailed derivations, the reader is referred to Giorgi (2017) .

3.2 Inertial Navigation

Inertial navigation belongs to the category of dead reckoning navigation techniques. The navigation sensor used in this technique, an Inertial Measurement Unit (IMU), is actually a sensor assembly that consists of three accelerometers and three gyroscopes in three orthogonal axes. The accelerometers measure specific forces that can be integrated to obtain velocity and position changes. The gyroscopes measure angular rates which can provide attitude information. An Inertial Navigation Systems (INS) uses measurements of an IMU, a gravity model and initial position, velocity and attitude values to compute a navigation solution.

3.2.1 Inertial Measurement Unit

As mentioned above, an IMU consists of three accelerometers and three gyroscopes mounted in orthogonal axes. In the following, the measurement principles of these sensors will be described briefly.

An accelerometer contains a proof mass that is connected to a case through a pair of springs. When acceleration acts on the sensor, the proof mass is displaced from its equilibrium position. This displacement is measured and scaled to obtain the acceleration. As an accelerometer measures acceleration in one sensitive axis, the proof mass needs to be supported in the other axes. Accelerometers used in navigation are either pendulous or vibrating-beam accelerometers (Groves, 2013). Both types can be built mechanically or using Micro-Electro-Mechanical System (MEMS) technology.

In pendulous accelerometers, the proof mass is connected to the case using a hinge and a pendulous arm. This arrangement constrains the movement of the proof mass and allows it only to move in the sensitive axis. Precise pendulous accelerometers use a

force-feedback or closed-loop configuration: There, the proof mass is kept in its equilibrium position and the force needed to keep it there is measured and proportional to the acceleration. In vibrating-beam accelerometers, the proof mass is also connected to the case using a pendulous arm, but supported along the sensitive axis using a vibrating beam. When the sensor is accelerated, the beam is compressed or stretched, which changes the resonant frequency. This frequency change is measured and used to determine specific force.

Gyroscopes measure the angular rate of the sensor with respect to an inertial frame. There are various types of gyroscopes, but the most common ones used in navigation are optical and vibratory gyroscopes. Ring Laser Gyroscope (RLG) and Interferometric Fiber-Optic Gyroscope (IFOG) belong to the category of optical gyroscopes, while MEMS gyroscopes are based on the vibratory principle.

Optical gyroscopes are based on the principle that when light travels through a medium, its speed is constant seen from an inertial frame. When light is sent into a closed-loop path (e.g. into an optical fiber or a path with mirrors) in both directions and there is no rotation, the path length for both beams of light is the same. When the closed-loop path is rotated in the same direction as the travelling direction of one beam, then the path length (seen from the inertial frame) increases for this beam and decreases for the beam in opposite direction (Sagnac effect). By measuring how the path length changes, the angular rate of the sensor with respect to the inertial frame can be determined. Vibratory gyroscopes contain a vibrating element and detect the Coriolis acceleration of this element when the gyroscope is rotated. By measuring the amplitude of the motion caused by the Coriolis acceleration, the angular rate can be determined.

Note that the price and the quality of an IMU largely depend on the type of gyroscopes that are used. An IMU with RLG is designed for high-performance applications and might cost a few 100 000 EUR. IFOG are a lower-cost alternative and available in different price and performance ranges. MEMS IMUs are cheapest and also available in different price ranges, from several tens to more than 1 000 EUR.

3.2.2 Inertial Sensor Errors

As every other navigation sensor, IMUs have measurement errors. These errors may be systematic or random (stochastic). Among the common systematic errors are a bias offset, scale factor error, non-linearity, scale factor sign asymmetry, dead zone, quantization error, non-orthogonality error, and misalignment error (Grewal et al., 2007). To reduce the effect of systematic errors, an IMU can be calibrated. The category of random or stochastic errors includes run-to-run bias offset, bias drift, scale factor instability and white noise (Noureldin et al., 2013). Random errors can be modelled stochastically, e.g. through a random walk or a first order Gauss-Markov process.

Noureldin et al. (2013) proposes the following mathematical model to describe gyroscope measurements:

$$\tilde{\boldsymbol{\omega}}_{ib}^{b} = \boldsymbol{\omega}_{ib}^{b} + \mathbf{b}_g + \mathbf{S}_g\boldsymbol{\omega}_{ib}^{b} + \mathbf{N}_g\boldsymbol{\omega}_{ib}^{b} + \boldsymbol{\varepsilon}_g. \tag{3.27}$$

In this model, $\boldsymbol{\omega}_{ib}^{b}$ is the true angular rate vector and $\tilde{\boldsymbol{\omega}}_{ib}^{b}$ is the measured angular rate vector; the subscript ib indicates that these are the angular rates from the body frame relative to the inertial frame, and the superscript b indicates that they are expressed in the body frame. $\mathbf{b}_g$ is the vector containing the gyroscope bias, $\mathbf{S}_g$ is a scale factor matrix, $\mathbf{N}_g$ the non-orthogonality matrix and $\boldsymbol{\varepsilon}_g$ is the sensor noise. The scale factor matrix is a diagonal matrix, where the diagonal elements $s_{g,(\cdot)}$ are the gyroscope scale factors for the three axes:

$$\mathbf{S}_g = \begin{pmatrix} s_{g,x} & 0 & 0 \\ 0 & s_{g,y} & 0 \\ 0 & 0 & s_{g,z} \end{pmatrix}. \tag{3.28}$$

The non-orthogonality matrix $\mathbf{N}_g$ contains the misalignment angles $\theta_{g,(\cdot)(\cdot)}$ as off-diagonal elements:

$$\mathbf{N}_g = \begin{pmatrix} 1 & \theta_{g,xy} & \theta_{g,xz} \\ \theta_{g,yx} & 1 & \theta_{g,yz} \\ \theta_{g,zx} & \theta_{g,zy} & 1 \end{pmatrix}. \tag{3.29}$$

The measurement model for the accelerometer measurement is defined as

$$\tilde{\mathbf{f}}^{b} = \mathbf{f}^{b} + \mathbf{b}_a + \mathbf{S}_1\mathbf{f}^{b} + \mathbf{S}_2\mathbf{f}^{b^2} + \mathbf{N}_a\mathbf{f} + \delta\mathbf{g} + \boldsymbol{\varepsilon}_a, \tag{3.30}$$

where $\mathbf{f}^b$ is the true specific force vector and $\tilde{\mathbf{f}}^b$ the measured specific force vector. $\mathbf{b}_a$ is the accelerometer bias vector, $\mathbf{S}_1$ and $\mathbf{S}_2$ are matrices for the scale factors, $\mathbf{N}_a$ is the non-orthogonality matrix, $\delta\mathbf{g}$ the vector describing how the true gravity deviates from the theoretical gravity and $\boldsymbol{\varepsilon}_a$ the sensor noise. Both the linear scale factor matrix $\mathbf{S}_1$ and the non-linear scale factor matrix $\mathbf{S}_2$ contain the linear scale factors for all three axes on their main diagonal:

$$\mathbf{S}_1 = \begin{pmatrix} s_{1,x} & 0 & 0 \\ 0 & s_{1,y} & 0 \\ 0 & 0 & s_{1,z} \end{pmatrix}, \tag{3.31}$$

$$\mathbf{S}_2 = \begin{pmatrix} s_{2,x} & 0 & 0 \\ 0 & s_{2,y} & 0 \\ 0 & 0 & s_{2,z} \end{pmatrix}. \tag{3.32}$$

In the non-orthogonality matrix of the accelerometer triad, the misalignment angles are the off-diagonal elements:

$$\mathbf{N}_a = \begin{pmatrix} 1 & \theta_{a,xy} & \theta_{a,xz} \\ \theta_{a,yx} & 1 & \theta_{a,yz} \\ \theta_{a,zx} & \theta_{a,zy} & 1 \end{pmatrix}. \tag{3.33}$$

3.2.3 Inertial Navigation Systems

There are two types of INS: gimbaled and strapdown systems. Gimbaled systems are aligned with the navigation frame so that the output of the accelerometers can be directly integrated to obtain velocity and position in the navigation frame. They are realized by mounting an IMU on a gimbaled platform that is rotated by torque motors in response to the rotations sensed by the gyroscopes (Noureldin et al., 2013). Gimbaled systems are expensive and mechanically complex and therefore rarely used.

In a strapdown system, the IMU is rigidly mounted onto the body of the moving platform. This makes the navigation equations more complex. An overview of the algorithm for processing IMU data in a strapdown system is given in Figure 3.2. Like every dead

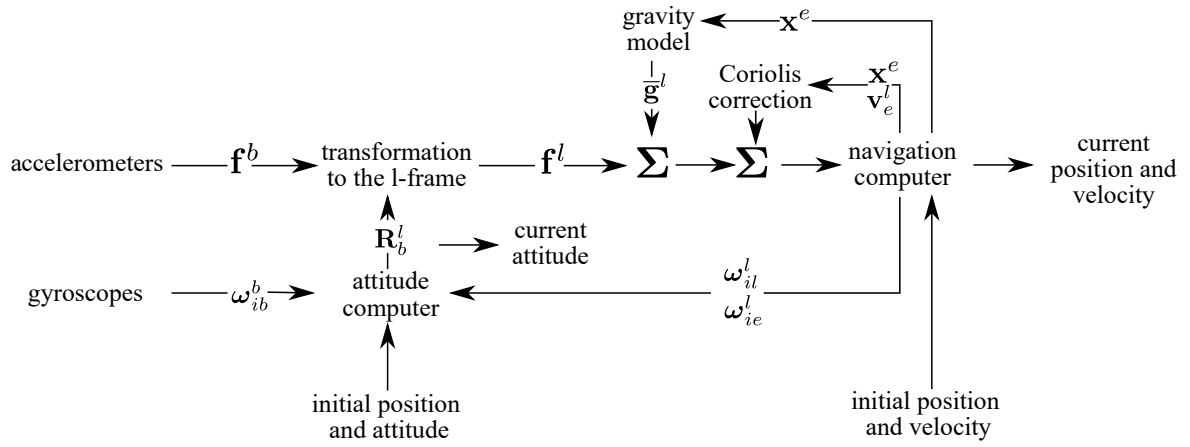


Figure 3.2: Diagram of strapdown algorithm. Adapted from Skaloud and Legat (2006).

reckoning system, it has to be initialized. The initial position $\mathbf{x}^e(t_0)$ in a terrestrial equatorial frame may be obtained from GNSS. The initial velocity $\mathbf{v}_e^l(t_0)$ is often assumed to be zero, or may be initialized with GNSS. The initial attitude matrix $\mathbf{R}_b^l(t_0)$ comes from an alignment phase.

The IMU measures specific forces and angular rates in the sensor frame. For the sake of simplicity, the sensor frame here is assumed to be aligned with the body frame, which is indicated by superscript b. The vector $\mathbf{f}^b$ contains the specific forces in the body frame, and the vector $\boldsymbol{\omega}_{ib}^b$ contains the angular rates of the body frame relative to the quasi-inertial i-frame expressed in the body frame.

In principle, the strapdown algorithm can be broken down to six computation steps (Skaloud and Legat, 2006). In the first step, the current attitude matrix $\mathbf{R}_b^l$ is obtained by integrating the angular rate measurements numerically. The following differential equation is integrated:

$$\dot{\mathbf{R}}_b^l\left(t_{k+1}\right) = \mathbf{R}_b^l\left(t_k\right)\boldsymbol{\Omega}_{ib}^b\left(t_{k+1}\right) - \boldsymbol{\Omega}_{il}^l\left(t_k\right)\mathbf{R}_b^l\left(t_k\right), \tag{3.34}$$

where the term $\boldsymbol{\Omega}_{ib}^b$ contains the angular rate measurements in the form of a skew-symmetric axiator matrix. $\mathbf{R}_b^l\left(t_k\right)$ contains the attitude information of the previous epoch and $\boldsymbol{\Omega}_{il}^l\left(t_k\right)$, also in the form of a skew-symmetric axiator matrix, contains the angular rates between the quasi-inertial frame and the local-level frame expressed in the local-level frame and is obtained from the solution of the the previous epoch.

In the second step, the measured specific forces at the current epoch are brought to the local-level frame with

$$\mathbf{f}^l\left(t_{k+1}\right) = \mathbf{R}_b^l\left(t_{k+1}\right)\mathbf{f}^b\left(t_{k+1}\right). \tag{3.35}$$

In the third step, the normal gravity vector is determined in the local-level frame:

$$\overline{\mathbf{g}}^l\left(t_k\right) = \begin{pmatrix} 0 \\ 0 \\ \gamma\left(\varphi\left(t_k\right), h\left(t_k\right)\right) \end{pmatrix}, \tag{3.36}$$

where $\gamma\left(\varphi\left(t_k\right), h\left(t_k\right)\right)$ is computed from the latitude and height of the previous epoch and from parameters of a reference ellipsoid.

In the fourth step, the Coriolis force is computed from

$$\mathbf{f}_{\text{Coriolis}}^l\left(t_k\right) = -\left(\Omega_{il}^l\left(t_k\right) + \Omega_{ie}^l\left(t_k\right)\right)\mathbf{v}_e^l\left(t_k\right), \tag{3.37}$$

where all terms come from the navigation solution of the previous epoch. $\mathbf{v}_e^l\left(t_k\right)$ is the velocity vector in the ECEF e-frame expressed in the local-level l-frame. Ω_{ie}^l is a skew-symmetric axiator matrix containing the angular rates between the quasi-inertial $i-$frame and the ECEF $e-$frame expressed in the local-level frame. The Coriolis force has to be considered because the reference frame that we navigate in rotates with respect to the (quasi-) inertial frame.

In the fifth step, the specific forces, normal gravity vector and Coriolis force (all expressed in the local-level frame) are integrated numerically to obtain the velocity vector at the current epoch. The differential form of the first navigation equation looks as follows:

$$\dot{\mathbf{v}}_e^l\left(t_{k+1}\right) = \mathbf{f}^l\left(t_{k+1}\right) + \overline{\mathbf{g}}^l\left(t_k\right) + \mathbf{f}_{\text{Coriolis}}^l\left(t_k\right). \tag{3.38}$$

In the sixth and last step, the second navigation equation is integrated to obtain the current position in the ECEF-frame. The differential equation reads as follows:

$$\dot{\mathbf{x}}^e\left(t_{k+1}\right) = \mathbf{R}_l^e\left(t_k\right)\mathbf{v}_e^l\left(t_{k+1}\right), \tag{3.39}$$

where $\mathbf{R}_l^e$ is the rotation matrix from the local-level frame to the ECEF-frame and can be computed with Equation 2.14.

3.3 Odometry

A navigation technique that can be used for ground vehicles or robots is odometry. Odometry refers to determining the speed of a vehicle or the distance travelled by measuring the wheel rotations (Groves, 2013). The term linear odometry refers to computing coordinate changes or the velocity of the vehicle, while differential odometry refers to computing heading information from differencing left and right wheel speeds.

3.3.1 Wheel speed sensors

The rotation of a wheel or track can be measured with rotary encoders. There are different types of encoders, such as magnetic, optical, and capacitive encoders.

Magnetic encoders detect changes in the magnetic flux density. They consist of a rotating magnetized disk or ring with several magnetized pole pairs, and magnetic sensors (e.g. Hall-effect sensors) located around the disk. When the disk rotates and a pole passes by a sensor, a voltage spike is generated. This allows counting the wheel revolutions. Magnetic encoders are robust against dirt and vibration (Smoot, 2022).

Optical encoders typically have a higher resolution than magnetic encoders, but are easily affected by dust or dirt. They consist of a light source (LED), a coded disk made of glass or plastic with opaque lines, and photodetectors on the opposite side of the disk. The LED emits a beam of light that passes through the coded disk to the photodetector. When the disk rotates, the photodetector either detects a light when the beam passes through a transparent part of the disk, or no light when the beam hits an opaque line.

Capacitive encoders work by detecting changes in capacitance (Collins, 2015). They consist of a rotor with a sinusoidal pattern, as well as a stationary transmitter and a stationary receiver. The transmitter emits a reference signal. When the rotor moves, the

pattern of the reference signal is modulated. From the modulated signal, information on the rotary motion can be derived.

3.3.2 Skid steering

There are different ways in which robots or ground vehicles can be steered and the steering configuration influences how we compute odometry information from the measurements of the wheel speed sensors. The most common method is Ackermann steering, which is used in most cars. The two front wheels are fixed to a common axle and are used for steering, while the two rear wheels (also fixed to a common axle) are used for driving.

Tracked vehicles, however, use skid steering. Skid steering is based on the concept of a differential drive. When both tracks are steered forward with the same speed, the vehicle follows a straight line. When one track is steered faster than the other, the vehicle takes a curved path. When one track is steered forward and the other backward, the vehicle rotates around its center.

Tracked vehicles have the advantage of being very stable on uneven terrain, but slippage occurs when they drive a curved path (Yamauchi et al., 2017). We can define the slip ratios a_r and a_l for the left and right tracks (Reitbauer and Schmied, 2021a)

$$a_r = \frac{v_r - v_r'}{v_r} \tag{3.40}$$

$$a_l = \frac{v_l - v_l'}{v_l}, \tag{3.41}$$

where v_r and v_l are the speeds measured by the rotary encoders of the right and left tracks, and v_r' and v_l' are the true ground speeds. Slip is affected by the resistances that occur when the tracks interact with the ground (e.g. a compaction resistance or bulldozing resistance). When slip is modelled using resistances, both soil and track parameters have to be known (Yamauchi et al., 2016).

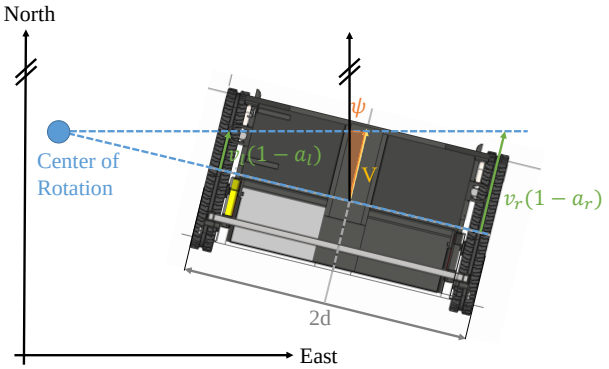


Figure 3.3: Odometry model for tracked vehicles. The true ground speed of the tracks ($v_i\prime$) is equal to the measured speed (v_i) multiplied by $(1 - a_i)$, where a_i is the slip ratio.

When the slip ratios and the current heading (ψ) of the vehicle are known, the velocity (linear odometry) in a local-level frame can be computed from:

$$\dot{N} = \frac{v_r\,(1 - a_r) + v_l\,(1 - a_l)}{2} \cos\psi, \tag{3.42}$$

$$\dot{E} = \frac{v_r\,(1 - a_r) + v_l\,(1 - a_l)}{2} \sin\psi, \tag{3.43}$$

where $\dot{N}$ is the velocity in North direction and $\dot{E}$ is the velocity in East direction. Furthermore, the heading rate $\dot{\psi}$ (differential odometry) can be obtained from:

$$\dot{\psi} = \frac{v_r\,(1 - a_r) - v_l\,(1 - a_l)}{2d}, \tag{3.44}$$

where $2d$ is the track width (see Figure 3.3).

3.4 Image-Based Navigation

Image-based navigation uses a sensor (e.g. a camera or stereo camera) to capture a 2-D or 3-D image of the environment (Groves, 2013). Image-based navigation systems can be used for both dead reckoning or position fixing. When subsequent images are compared to derive translational and rotational motion, we speak of dead reckoning. When the

images or features of the images are compared to information stored in a database, e.g. when a point cloud of a stereo camera is registered to a target point cloud of the environment to obtain the so-called *pose* (position and attitude) of the sensor, this is a form of position fixing. The following section will first present different imaging sensors, and then discuss one method of image-based dead reckoning (stereo visual odometry) as well as one method of position fixing (Normal Distributions Transform (NDT)) which uses image-based sensors.

3.4.1 Imaging Sensors

Imaging sensors can be either active or passive. Active sensors transmit a signal and measure the time it takes for the signal to be reflected and return to the sensor. The signal might be an electromagnetic radio wave (as used in radars), an acoustic wave (sonar), or light (LIght Detection And Ranging (LIDAR)). Passive sensors do not emit a signal; commonly used passive sensors for image-based navigation are cameras and stereo-cameras.

3.4.2 Stereo Visual Odometry

Visual Odometry (VO) is a technique of dead reckoning that estimates a camera's relative motion from a sequence of images. Since a stereo camera will later be used as a navigation sensor in this thesis, the concept of stereo visual odometry will be briefly discussed here.

According to Yousif et al. (2015), algorithms for stereo VO can be described with six steps:

1. Features in the left frame and right frame at time t are extracted and matched to reconstruct their 3D coordinates through triangulation.
2. The features in (1.) are matched with their corresponding features at time $t + 1$.

3. A homogeneous transformation $\mathbf{T}_t^{t+1}$, which gives the minimum (argmin) sum of squared distances, is estimated:

$$\mathbf{T}_t^{t+1} = \underset{\mathbf{T}_t^{t+1}}{\operatorname{argmin}} \sum_i \left| \mathbf{X}_{t+1} - \mathbf{T}_t^{t+1} \mathbf{X}_t \right|^2, \tag{3.45}$$

   where $\mathbf{X}_{t+1}$ and $\mathbf{X}_t$ are the 3D coordinates of the features at time $t+1$ and t, respectively.

4. The transformation is refined and recalculated without outliers, e.g. using a RANdom SAmpling Consensus (RANSAC) approach.

5. The transformation obtained from step (4.) is multiplied by the transformation from the previous epoch to obtain the global transformation (see Equation 2.18).

6. Steps (1.)-(5.) are repeated for each time step.

3.4.3 Normal Distributions Transform (NDT)

As mentioned before, image-based sensors can also be used for position fixing when a-priori information of the environment that they are observing is available. When a stereo camera is used, the point cloud observed can be matched to a target point cloud that was previously recorded. The process of matching point clouds is known as *registration*. A widely used registration algorithm is Iterative Closest Point (ICP), which estimates the pose by minimizing the sum of squared distances between corresponding points. However, ICP is computationally expensive.

A registration method that is less computationally expensive is Normal Distributions Transform (NDT) method. It was first proposed by Biber and Strasser (2003) as a method to match 2D scans, and was later extended by Magnusson (2009) for the three-dimensional case. As NDT is later used in this thesis, the algorithm is explained in the following.

In a first step, the target point cloud is divided into voxels. For every voxel, the mean vector $\boldsymbol{\mu}$

$$\boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x_i} \tag{3.46}$$

and covariance matrix $\mathbf{Q}$

$$\mathbf{Q} = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x_i} - \boldsymbol{\mu}) (\mathbf{x_i} - \boldsymbol{\mu})^T \tag{3.47}$$

are computed from the points $\mathbf{x}_{i=1\ldots n}$ in the voxel. The parameters $\boldsymbol{\mu}$ and $\mathbf{Q}$ are stored for every voxel.

We introduce the pose $\mathbf{p}_6$:

$$\mathbf{p}_6 = (c_1, c_2, c_3, \theta, \phi, \psi)^{\mathrm{T}}, \tag{3.48}$$

a 6×1 column vector that contains three parameters for translation (c_1, c_2, c_3) and three parameters for rotation, namely the Euler angles θ, ϕ, ψ. The goal of NDT is to optimally estimate the parameters in $\mathbf{p}_6$, i.e. the translation and rotation of the stereo camera with respect to the target scan.

The transformation function T_E that transforms a point $\mathbf{x}$ by the pose $\mathbf{p}_6$ can be written as (Magnusson, 2009)

$$T_E (\mathbf{p}_6, \mathbf{x}) = \mathbf{R}_1(\theta)\mathbf{R}_2(\phi)\mathbf{R}_3(\psi)\mathbf{x} + \mathbf{c}, \tag{3.49}$$

where the vector $\mathbf{c} = (c_1, c_2, c_3)^{\mathrm{T}}$ contains the parameters for translation.

The likelihood that the points $\mathbf{x}_k$ measured by the stereo camera correspond to the points on the target scan when transformed by the pose $\mathbf{p}_6$ can be expressed as:

$$\Psi = \prod_{k=1}^{n} p (T_E (\mathbf{p}_6, \mathbf{x}_k)), \tag{3.50}$$

where $p(\mathbf{x})$ is the probability density function for a scan point. NDT tries to find the best pose, i.e. the pose with maximum likelihood, or the minimum negative logarithmic likelihood:

$$s(\mathbf{p}_6) = -\log \Psi = -\sum_{k=1}^{n} \log \left(p \left(T_E \left(\mathbf{p}_6, \mathbf{x}_k \right) \right) \right). \tag{3.51}$$

Here, $s(\mathbf{p}_6)$ is the NDT score function.

Using a Gaussian approximation, the probability for a single scan point $\mathbf{x}_k$ to lie on the surface of the reference scan when transformed by pose $\mathbf{p}_6$ can be expressed by (Magnusson, 2009)

$$p(\mathbf{x}_k) = -d_1 \exp\left(-\frac{d_2}{2} (\mathbf{x}_k - \boldsymbol{\mu})^{\mathrm{T}} \mathbf{Q}^{-1} (\mathbf{x}_k - \boldsymbol{\mu}) \right), \tag{3.52}$$

where $\boldsymbol{\mu}$ and $\mathbf{Q}$ are the mean vector and covariance matrix of the voxel in which $\mathbf{x}_k$ lies. The parameters d_1 and d_2 are used to bound the influence of outliers.

NDT minimizes the score function (Equation 3.51) to find the optimal pose $\mathbf{p}_6$ using Newton's algorithm (Magnusson, 2009). The equation

$$\mathbf{H} \Delta \mathbf{p}_6 = -\mathbf{g} \tag{3.53}$$

is solved iteratively, where $\mathbf{H}$ is the Hessian matrix containing the second-order partial derivatives of the score function, and $\mathbf{g}$ is the gradient vector of the score function. In each iteration, the increment $\Delta \mathbf{p}_6$ is added to the current pose ($\mathbf{p}_6 \leftarrow \mathbf{p}_6 + \Delta \mathbf{p}_6$). The iteration is carried out until the pose estimate $\mathbf{p}_6$ converges.

Chapter 4

Bayes Filtering

The task of positioning in navigation is to estimate a moving object's *state vector* from sensor data. The state vector may consist of the coordinates of the object with respect to a reference frame, as well as its velocity and attitude, and some additional parameters, such as systematic sensor errors. Usually, these parameters are not directly measurable, but we have to rely on navigation sensors to provide us with partial information on these quantities. Since an ideal sensor does not exist, all measurements are affected by noise. One mathematical tool that helps us to estimate the state vector from noisy data are the so-called *Bayes filters*.

Bayes filters are recursive filters for state estimation. When using Bayes filters, both the sensor measurements $\mathbf{z_t}$ as well as the state vector $\mathbf{x_t}$ are modelled as random variables with probability density functions. Some important concepts of Bayes filters will be discussed in the following.

A state vector $\mathbf{x_t}$ is *complete* if it is the best predictor of the future, i.e. when previous measurements or states do not carry additional information that help to better predict the future (Thrun et al., 2006). When a state is complete, we can express the *state transition probability* as

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t), \tag{4.1}$$

where $\mathbf{x}_{t-1}$ is the complete state of the previous epoch and $\mathbf{u}_t$ are control data that carry information on how the state changes (e.g. odometer measurements).

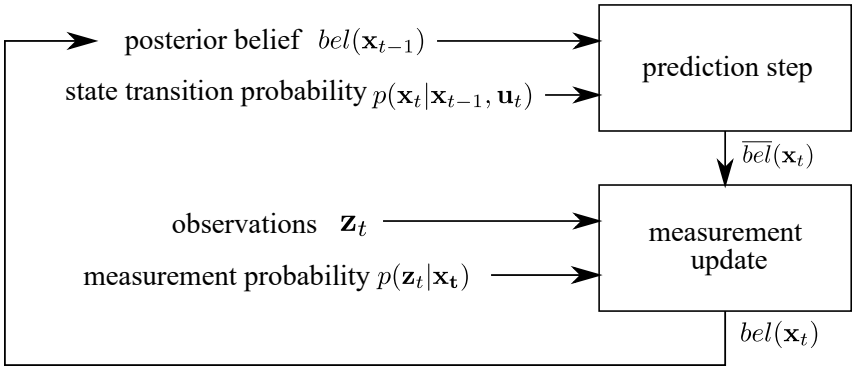


Figure 4.1: Flowchart of Bayes filtering.

Furthermore, when the state $\mathbf{x_t}$ is complete, the *measurement probability* is conditionally independent from past measurements and can be expressed solely by

$$p(\mathbf{z}_t|\mathbf{x_t}). \tag{4.2}$$

The state transition probability (Equation 4.1) and the measurement probability (Equation 4.2) describe the so-called Dynamic Bayes Network (DBN). The assumption that the state is complete is sometimes also referred to as Markov assumption.

Another important concept for Bayes filtering is the concept of *belief*. A belief $bel(\mathbf{x}_t)$ is a conditional probability distribution that represents a moving object's knowledge of its state $\mathbf{x}_t$.

Figure 4.1 gives an overview of how Bayes filters work. All Bayes filters are based on the same two steps (Thrun et al., 2006): In the first step, the so-called *prediction step*, the prior belief of $\mathbf{x}_t$ is computed from the previous posterior belief and information on how the state changes (dynamic model, $\mathbf{u}_t$):

$$\overline{bel}(\mathbf{x}_t) = \int p(\mathbf{x}_t|\mathbf{u}_t, \mathbf{x}_{t-1}) bel(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1}. \tag{4.3}$$

Equation 4.3, which links the prior belief $\overline{bel}(\mathbf{x}_t)$ with the previous posterior belief $bel(\mathbf{x}_{t-1})$, is also known as the Chapman-Kolmogorov equation (Haug, 2012). In the second step, the *measurement update*, Bayes' law is used to obtain the new belief. Here, the predicted belief is multiplied by the measurement probability (and a normalization

constant η) to estimate the final (posterior) belief

$$bel(\mathbf{x}_t) = \eta p(\mathbf{z}_t|\mathbf{x_t})\overline{bel}(\mathbf{x}_t). \tag{4.4}$$

Since Bayes filters are recursive algorithms, the prediction step (Equation 4.3) and measurement update (Equation 4.4) are repeated for every epoch t.

Bayes filters can be divided into two categories: Gaussian and non-parametric filters. In Gaussian filters, the beliefs are modelled as multivariate normal distributions. By contrast, non-parametric filters do not model the belief with a fixed functional relationship, but may represent it using a histogram (histogram filter) or a finite number of particles (particle filter). In the following, an overview of commonly used Gaussian and non-parametric filters will be given. As a Kalman filter will be used later in this thesis, it will be explained in more detail.

4.1 Gaussian Filters

Gaussian filters are recursive Bayes filters which model beliefs using multivariate normal distributions. The probability density function for a multivariate normal distribution for the state vector $\mathbf{x}$ of dimension $n \times 1$ is given by

$$p(\mathbf{x}) = \det(2\pi\mathbf{C})^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\mathbf{C}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right), \tag{4.5}$$

where $\boldsymbol{\mu}$ is a mean vector with the same dimension as the state vector, and $\mathbf{C}$ is the $n \times n$-covariance matrix. The parametrization in Equation 4.5 is also called moments parametrization, as the mean is the first and the covariance the second moment of a probability distribution (all higher moments are zero for normal distributions). This type of parametrization is used in the Kalman filter.

Note that there is also an alternative parametrization, the so-called canonical parametrization, which uses an information matrix $\boldsymbol{\Omega} = \mathbf{C}^{-1}$ and an information vector $\boldsymbol{\zeta} = \mathbf{C}^{-1}\boldsymbol{\mu}$. The canonical parametrization is used in the information filter. For details on the information filter, the reader is referred to Thrun et al. (2006).

4.1.1 Kalman Filter

The Kalman Filter was first formulated by Kalman (1960) for linear Gaussian systems. In the Kalman filter, beliefs are represented by the mean $\boldsymbol{\mu}$ and covariance $\mathbf{C}$.

The following properties must be fulfilled so that the posterior belief is Gaussian (Thrun et al., 2006):

1. The dynamic model (or the state transition probability) which expresses how the state changes from epoch $t-1$ to t has to be linear and the added noise has to be Gaussian. For the discrete Kalman filter, this can be expressed by (Hofmann-Wellenhof et al., 2003)
$$\mathbf{x}_t = \boldsymbol{\Phi}_{t-1}\mathbf{x}_{t-1} + \mathbf{w}_{k-1}, \tag{4.6}$$
where $\mathbf{x}$ is the $n \times 1$ state vector with its subscripts t and $t-1$ referring to the current and the previous epoch, $\boldsymbol{\Phi}_{t-1}$ is the $n \times n$ transition matrix, and $\mathbf{w}_{k-1}$ is the Gaussian system noise with zero mean and $n \times n$ covariance $\mathbf{Q}_{t-1}$ ($\mathbf{w}_{t-1} \sim N(\mathbf{0}, \mathbf{Q}_{t-1})$).

2. The observation model and the measurement probability have to be linear with added Gaussian noise. The linear observation model is given by (Hofmann-Wellenhof et al., 2003):
$$\mathbf{z}_t = \mathbf{H}_t\mathbf{x}_t + \mathbf{v}_t, \tag{4.7}$$
where $\mathbf{z}_t$ is the $l \times 1$ observation vector, and $\mathbf{H}_t$ is the $l \times n$ design matrix describing the linear relation between the measurements and the state vector. $\mathbf{v}_t$ is the Gaussian measurement noise with zero mean and $l \times l$ covariance $\mathbf{R}_t$ ($\mathbf{v}_t \sim N(\mathbf{0}, \mathbf{R}_t)$).

3. The initial belief must follow a Gaussian distribution with mean $\boldsymbol{\mu}_0$ and covariance $\mathbf{C}_0$.

With the above conditions and the Markov assumption fulfilled, we can now formulate the equations for the discrete linear Kalman filter. In the following, the notation by Hofmann-Wellenhof et al. (2003) is used, where a tilde ($\tilde{}$) denotes a prediction and a hat ($\hat{}$) indicates an estimated variable. Note that some authors, like Groves (2013) or Noureldin et al. (2013), use a minus (-) for the prediction and a plus (+) for the estimate that includes the measurements.

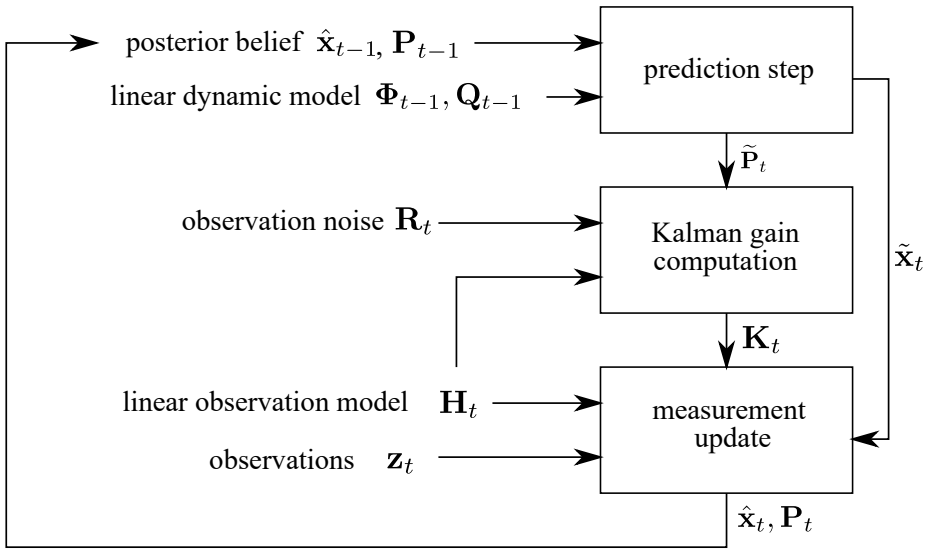


Figure 4.2: Flowchart of linear Kalman filtering.

An overview of the Kalman filter algorithm is given in Figure 4.2. In the time update or prediction step, the estimated state vector $\hat{\mathbf{x}}_{t-1}$ from the previous epoch, as well as its covariance matrix $\mathbf{P}_{t-1}$ are predicted to the current epoch t with:

$$\tilde{\mathbf{x}}_t = \mathbf{\Phi}_{t-1}\hat{\mathbf{x}}_{t-1}, \tag{4.8}$$

$$\widetilde{\mathbf{P}}_t = \mathbf{\Phi}_{t-1}\mathbf{P}_{t-1}\mathbf{\Phi}_{t-1}^T + \mathbf{Q}_{t-1}, \tag{4.9}$$

where $\mathbf{\Phi}_{t-1}$ is the linear transition matrix and $\mathbf{Q}_{t-1}$ the covariance matrix of the system noise $\mathbf{w}_{k-1}$ (see Equation 4.6).

In an intermediate step, the Kalman gain matrix is computed from the design matrix $\mathbf{H}_t$, the predicted covariance matrix $\widetilde{\mathbf{P}}_t$ of the state vector and the covariance matrix of the observations $\mathbf{R}_t$:

$$\mathbf{K}_t = \widetilde{\mathbf{P}}_t\mathbf{H}_t^T\left(\mathbf{H}_t\widetilde{\mathbf{P}}_t\mathbf{H}_t^T + \mathbf{R}_t\right)^{-1}. \tag{4.10}$$

The Kalman gain matrix is needed for the measurement update. It expresses how much weight is given to the observations to update the state in comparison to the prediction, where each column corresponds to a measurement and each row to a state (Groves, 2013). The Kalman gain depends both on the covariance of the predicted state $\widetilde{\mathbf{P}}_t$ and the covariance of the observations $\mathbf{R}_t$. When $\widetilde{\mathbf{P}}_t$ is big (high process noise) and $\mathbf{R}_t$ is small (highly accurate observations), then $\mathbf{K}_t$ becomes larger and more weight is given

to the observations. When the dynamic model is very accurate ($\widetilde{\mathbf{P}}_t$ is small) and the observations are noisy ($\mathbf{R}_t$ is big), then $\mathbf{K}_t$ becomes smaller and more weight is given to the prediction (Noureldin et al., 2013).

In the measurement update, the prediction of the state vector is updated with the new observations $\mathbf{z}_t$ with

$$\hat{\mathbf{x}}_t = \tilde{\mathbf{x}}_t + \mathbf{K}_t \left(\mathbf{z}_t - \mathbf{H}_t \tilde{\mathbf{x}}_t \right) \tag{4.11}$$

and the covariance matrix of the state vector is updated using

$$\mathbf{P}_t = \left(\mathbf{I} - \mathbf{K}_t \mathbf{H}_t \right) \tilde{\mathbf{P}}_t. \tag{4.12}$$

Note that instead of Equation 4.12, the numerically more stable *Joseph* form (Bucy and Joseph, 2005) can also be used:

$$\mathbf{P}_t = \left(\mathbf{I} - \mathbf{K}_t \mathbf{H}_t \right) \tilde{\mathbf{P}}_t \left(\mathbf{I} - \mathbf{K}_t \mathbf{H}_t \right)^T + \mathbf{K}_t \mathbf{R}_t \mathbf{K}_t^T. \tag{4.13}$$

Since the Kalman filter is a recursive algorithm for state estimation, Equations 4.8 to 4.12 are repeated for every epoch. Note that the prediction step (Equation 4.9) increases uncertainty (as the system noise matrix $\mathbf{Q}_{t-1}$ is a covariance matrix and therefore positive semi-definite), while the measurement update (Equation 4.12) decreases uncertainty in the belief.

4.1.2 Extended Kalman Filter

While the Kalman filter is a simple tool for Bayes filtering, it has a major drawback: it requires both a linear observation model, and a linear dynamic model. In reality, however, the functional relationships are often nonlinear (e.g. when GNSS pseudoranges are used as observations to estimate the state vector). This problem can be circumvented through linearisation and the EKF.

In an EKF, the non-linear dynamic model is given by (Hofmann-Wellenhof et al., 2003):

$$\mathbf{x}_t = \boldsymbol{\varphi}_{t-1} \left(\mathbf{x}_{t-1} \right) + \mathbf{w}_{t-1}, \tag{4.14}$$

where $\boldsymbol{\varphi}_{t-1}(\mathbf{x}_{t-1})$ is a vector containing the dynamic functions describing how the state changes from epoch $t-1$ to t, and $\mathbf{w}_{t-1}$ is the Gaussian system noise. The dynamic functions are not always directly available, but sometimes given in the form of differential equations as follows (Groves, 2013):

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t) + \mathbf{w}(t). \tag{4.15}$$

In this case, Equation 4.15 has to be integrated to obtain the dynamic functions $\boldsymbol{\varphi}_{t-1}$.

The non-linear observation model is given by

$$\mathbf{z}_t = \mathbf{h}_t(\mathbf{x}_t) + \mathbf{v}_t, \tag{4.16}$$

where $\mathbf{h}_t(\mathbf{x}_t)$ is a vector containing the observation functions and $\mathbf{v}_t$ is the Gaussian observation noise.

In an EKF, we need to linearize the non-linear dynamic model and the observation model. The dynamic model is linearized as follows to obtain the $n \times n$ dynamic matrix $\boldsymbol{\Phi}$ (with n being the length of the state vector), with its elements $\Phi_{t-1(j,q)}$:

$$\Phi_{t-1(j,q)} = \left.\frac{\partial \varphi_{t-1(j)}(\mathbf{x})}{\partial x_{(q)}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{t-1}}, \tag{4.17}$$

with the indices $j = 1 \dots n$ and $q = 1 \dots n$. Note that for the transition matrix, the estimated state of the previous epoch $\hat{\mathbf{x}}_{t-1}$ serves as the Taylor point. Ideally, one would linearize with respect to the true state, but since the true state is unknown, the estimated state of the previous epoch is the best estimate of the true state.

The $l \times n$ observation matrix $\mathbf{H}_t$, with l being the number of observations at epoch t and n the number of parameters in the state vector, is obtained by the following linearization:

$$H_{t(i,p)} = \left.\frac{\partial h_{t(i)}(\mathbf{x})}{\partial x_p}\right|_{\mathbf{x}=\tilde{\mathbf{x}}_t}, \tag{4.18}$$

where the index $i = 1 \dots l$ and $p = 1 \dots n$. For the linearized design matrix, the predicted state vector $\tilde{\mathbf{x}}_t$ serves as the Taylor point (as it is the best available estimation of the true state at this point).

An EKF executes similar computation steps as the linear Kalman filter (Equations 4.8 to 4.12). In contrast to the linear Kalman filter, however, the EKF uses the linearized transition matrix (Equation 4.17) and the linearized design matrix (Equation 4.18) to compute the updates of the covariance matrix $\mathbf{P}$ and the Kalman weight $\mathbf{K}$. The state vector is updated with the nonlinear vector equations, a process which will be explained in the following.

To compute the time update, the state vector of the previous epoch is propagated to the current epoch by applying the vector equations φ_{t-1}:

$$\tilde{\mathbf{x}}_t = \varphi_{t-1}\left(\hat{\mathbf{x}}_{t-1}\right). \tag{4.19}$$

The formula for the time update of the covariance matrix is identical to Equation 4.9, where now the matrix $\mathbf{\Phi}_{t-1}$ contains the linearized dynamic functions with respect to the estimated state vector of the previous epoch (Equation 4.17).

The Kalman weight is computed with Equation 4.10, where the design matrix $\mathbf{H}_t$ now comes from the linearization in Equation 4.18.

In the measurement update, the new state vector is computed with

$$\hat{\mathbf{x}}_t = \tilde{\mathbf{x}}_t + \mathbf{K}_t\left(\mathbf{z}_t - \mathbf{h}_t\left(\tilde{\mathbf{x}}_t\right)\right). \tag{4.20}$$

The covariance matrix is updated with Equation 4.12.

4.1.3 Error-State Extended Kalman Filter

Another variant for dealing with nonlinear functional relationships is the error-state formulation of the EKF. While the classical EKF estimates the total state vector $\mathbf{x}_t$, the error-state EKF estimates the change of the state vector $\delta\mathbf{x}_t$ (= error state vector). When the error state vector is used to update the state vector, we speak of a closed-loop error state filter (Wendel, 2011). According to Groves (2013), the best stage for updating the state vector with the error states is immediately after the measurement update, since then the error state vector is zero when the time update begins.

With the closed-loop implementation of the error-state EKF, the state vector is propagated with

$$\tilde{\mathbf{x}}_t = \varphi_{t-1}\left(\hat{\mathbf{x}}_{t-1}\right), \tag{4.21}$$

and the corresponding covariance matrix with

$$\widetilde{\mathbf{P}}_t = \mathbf{\Phi}_{t-1}\mathbf{P}_{t-1}\mathbf{\Phi}_{t-1}^T + \mathbf{Q}_{t-1}. \tag{4.22}$$

Note that these steps are the same as for the EKF (Equations 4.19 and 4.9). Also, the Kalman gain is computed identically to the gain computation in the linear Kalman filter and the EKF (Equation 4.10).

In contrast to the classical EKF, an error state EKF does not directly process the measurements $\mathbf{z}_t$, but rather the difference $\delta\mathbf{z}_t$ between the predicted measurements $\mathbf{h}_t(\tilde{\mathbf{x}}_t)$ and the observations $\mathbf{z}_t$:

$$\delta\mathbf{z}_t = \mathbf{h}_t(\tilde{\mathbf{x}}_t) - \mathbf{z}_t. \tag{4.23}$$

It then estimates the error state vector with

$$\delta\hat{\mathbf{x}}_t = \mathbf{K}_t\delta\mathbf{z}_t. \tag{4.24}$$

The covariance matrix is updated in the same way as for the normal Kalman filter and the normal EKF, with Equation 4.12. For the closed-loop implementation, the state vector is updated immediately after the measurement update:

$$\hat{\mathbf{x}}_t = \tilde{\mathbf{x}}_t - \delta\hat{\mathbf{x}}_t. \tag{4.25}$$

Note that the error state formulation of the EKF is often used for GNSS/INS integration.

A major advantage of the EKF (and the error-state EKF) is that it is computationally efficient. Each update step has a computational complexity of

$$O(l^{2.4} + n^2), \tag{4.26}$$

where l is the dimension of the measurement vector and n is the dimension of the state vector (Thrun et al., 2006).

4.1.4 Unscented Kalman Filter

The Unscented Kalman Filter (UKF) was first proposed by Julier and Uhlmann (1997). In contrast to the EKF, which uses Taylor series for linearization, the UKF uses a stochastic linearization (Thrun et al., 2006) in the form of an unscented transformation.

An unscented transformation is a method that can be used to compute the statistics of a random variable which is transformed with a nonlinear function (Wan and van der Merwe, 2000). The UKF transforms the $n \times 1$ state vector $\mathbf{x}_t$ and its covariance matrix $\mathbf{P}$ to $2n$ sigma points $\mathbf{x}_t^{(i)}$. The unscented transformation is reversible, i.e. the state vector can be computed from the mean of the sigma points and the covariance matrix from the variance of the sigma points (Groves, 2013).

There are different types of unscented transformations (see e.g. Simon (2006) or Menegaz et al. (2015)). In the following, we will use the square-root form, as it is numerically more stable (Menegaz et al., 2015).

For the prediction step of the UKF, we first need to compute the square root $\mathbf{S}_{t-1}$ of covariance matrix $\mathbf{P}_{t-1}$. Here, Cholesky factorization can be used to solve

$$\mathbf{P}_{t-1} = \mathbf{S}_{t-1}\mathbf{S}_{t-1}^T. \tag{4.27}$$

In the next step, the sigma points $\hat{\mathbf{x}}_{t-1}^{(i)}$ are computed from (Groves, 2013):

$$\hat{\mathbf{x}}_{t-1}^{(i)} = \hat{\mathbf{x}}_{t-1} + \sqrt{n}\mathbf{S}_{t-1,:,i} \tag{4.28}$$

for $i \leq n$, and

$$\hat{\mathbf{x}}_{t-1}^{(i)} = \hat{\mathbf{x}}_{t-1} - \sqrt{n}\mathbf{S}_{t-1,:,(i-n)}, \tag{4.29}$$

for $i > n$, where n is the length of the state vector. The subscripts $:, i$ indicate the i-th column of the matrix, and the subscripts $:, (i-n)$ indicate the $(i-n)$-th column of the matrix.

Now, each sigma point is propagated to the current epoch with

$$\tilde{\mathbf{x}}_t^{(i)} = \varphi_{t-1}\left(\hat{\mathbf{x}}_{t-1}^{(i)}\right). \tag{4.30}$$

In the last part of the prediction step, the predicted state vector $\tilde{\mathbf{x}}_t$ and its covariance matrix $\tilde{\mathbf{P}}_t$ are obtained from:

$$\tilde{\mathbf{x}}_t = \frac{1}{2n}\sum_{i=1}^{2n}\tilde{\mathbf{x}}_t^{(i)}, \tag{4.31}$$

$$\tilde{\mathbf{P}}_t = \frac{1}{2n}\sum_{i=1}^{2n}\left(\tilde{\mathbf{x}}_t^{(i)} - \tilde{\mathbf{x}}_t\right)\left(\tilde{\mathbf{x}}_t^{(i)} - \tilde{\mathbf{x}}_t\right)^{\mathrm{T}} + \mathbf{Q}_{k-1}. \tag{4.32}$$

For the measurement update, the square root $\tilde{\mathbf{S}}_t$ of the predicted covariance matrix $\tilde{\mathbf{P}}_t = \tilde{\mathbf{S}}_t\tilde{\mathbf{S}}_t^T$ is obtained from Cholesky decompostion. Then, new sigma points are generated using the predicted state vector $\tilde{\mathbf{x}}_t$ and $\tilde{\mathbf{S}}_t$:

$$\tilde{\mathbf{x}}_t^{(i)} = \tilde{\mathbf{x}}_t + \sqrt{n}\tilde{\mathbf{S}}_{t,:,i} \tag{4.33}$$

for $i \leq n$, and

$$\tilde{\mathbf{x}}_t^{(i)} = \tilde{\mathbf{x}}_t - \sqrt{n}\tilde{\mathbf{S}}_{t,:,(i-n)}, \tag{4.34}$$

for $i > n$. Again, the subscripts $:, i$ indicate the i-th column of the matrix, and the subscripts $:, (i-n)$ indicate the $(i-n)$-th column of the matrix.

Next, the reduced observations for each sigma point are computed with

$$\delta\mathbf{z}_t^{(i)} = \mathbf{h}_t(\tilde{\mathbf{x}}_t^{(i)}) - \mathbf{z}_t, \tag{4.35}$$

and the mean observation is received from

$$\delta\mathbf{z}_t = \frac{1}{2n}\sum_{i=1}^{2n}\delta\mathbf{z}_t^{(i)}. \tag{4.36}$$

The covariance matrix of the observations is computed from (Groves, 2013):

$$\mathbf{C}_t = \frac{1}{2n} \sum_{i=1}^{2n} \left(\delta \mathbf{z}_t^{(i)} - \delta \mathbf{z}_t \right) \left(\delta \mathbf{z}_t^{(i)} - \delta \mathbf{z}_t \right)^{\mathrm{T}} + \mathbf{R}_t \tag{4.37}$$

and the Kalman gain matrix from (Groves, 2013):

$$\mathbf{K}_t = \left(\frac{1}{2n} \sum_{i=1}^{2n} \left(\tilde{\mathbf{x}}_t^{(i)} - \tilde{\mathbf{x}}_t \right) \left(\delta \mathbf{z}_t^{(i)} - \delta \mathbf{z}_t \right)^{\mathrm{T}} \right) \mathbf{C}_t^{-1} \tag{4.38}$$

In the last phase of the measurement update of the UKF, the state vector is updated with

$$\hat{\mathbf{x}}_t = \tilde{\mathbf{x}}_t - \mathbf{K}_t \delta \mathbf{z}_t, \tag{4.39}$$

and the covariance matrix is updated using

$$\mathbf{P}_t = \tilde{\mathbf{P}}_t - \mathbf{K}_t \mathbf{C}_t \mathbf{K}_t^{\mathrm{T}}. \tag{4.40}$$

The UKF yields better results than the EKF when the measurement model or the dynamic model are significantly nonlinear. This is because the sigma points, when propagated through a nonlinear system, capture the posterior mean and covariance to the third order of a Taylor series expansion (Wan and van der Merwe, 2000). Note that UKF can also be combined with the EKF. For navigation, the UKF can be used in the time update when there are large attitude uncertainties, or in the measurement update when short ranges are processed (Groves, 2013).

4.2 Non-Parametric Filters

Non-parametric filters do not require the posterior to be Gaussian or follow a fixed functional form. Instead, the posterior can be represented by a histogram, as used in the histogram filter, or by a finite number of samples or particles, as in the particle filter (Thrun et al., 2006).

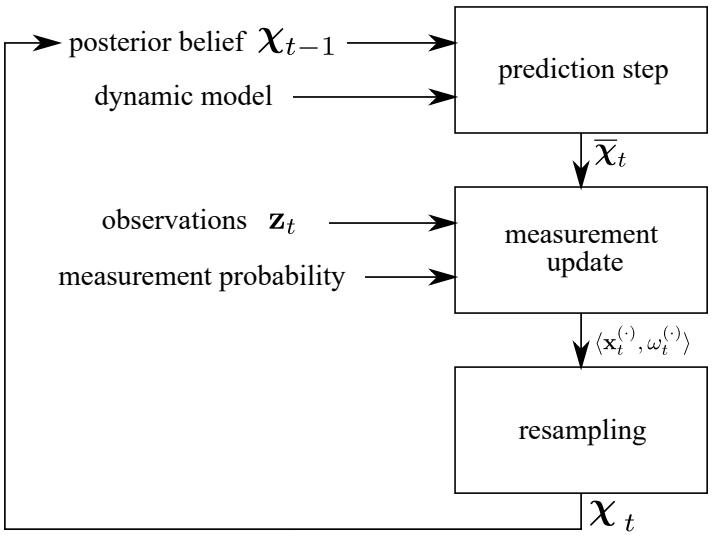


Figure 4.3: Flowchart of particle filtering.

4.2.1 Particle Filter

In the particle filter, the belief is represented by a set of m particles

$$\chi_t = \{\mathbf{x}_t^{(1)}, \mathbf{x}_t^{(2)}, \ldots, \mathbf{x}_t^{(m)}\}. \tag{4.41}$$

Each particle $\mathbf{x}_t^{(\cdot)}$ is an hypothesis of what the state might be at time t. All particles have equal weight $w_t^{(\cdot)} = \frac{1}{m}$. The particles are distributed proportionally according to the posterior belief. This means that when a region of the state space is densely populated by particles, the likelihood that the true state is in this region is high (Thrun et al., 2006). Similarly, a low density of particles in a region represents a low probability that the true state lies in this region.

Like all Bayes filters, the particle filter uses a prediction step and a measurement update. Figure 4.3 gives an overview of the computation steps in particle filtering. In the following, the individual computation steps of the particle filter will be explained.

In the prediction step, the posterior belief, i.e. the set of particles of the previous epoch χ_{t-1}, is propagated to the current epoch. Each particle is propagated individually by sampling from the state transition distribution or, in other words, by applying a dynamic model. Here, the dynamic model does not have to be linear. The resulting set of particles corresponds to the new prior belief $\overline{bel}(\mathbf{x}_t) = \overline{\chi}_t$.

In the measurement update, Bayes' law (Equation 4.4) is applied to obtain the new belief. To obtain the measurement probability, an importance factor $w_t^{(\cdot)}$ is computed for each particle. The importance factor corresponds to the probability that the measurement $\mathbf{z}_t$ is observed at the location of the particle $\mathbf{x}_t^{(\cdot)}$:

$$\omega_t^{(\cdot)} = p(\mathbf{z}_t|\mathbf{x}_t^{(\cdot)}). \tag{4.42}$$

The factor $\omega_t^{(\cdot)}$ can also be interpreted as the weight of a particle $\mathbf{x}_t^{(\cdot)}$. The set of weighted particles corresponds to the posterior belief. However, we want to have a set of particles with equal weight $w_t^{(\cdot)} = \frac{1}{m}$, where the distribution of the particles represents the posterior belief. Therefore, an additional computation step, the so-called *resampling*, is needed in the particle filter.

In the resampling step, a new set of m particles with equal weight is drawn from the set of m weighted particles in a way that particles with a previously low importance weight are eliminated and particles with a high importance weight are duplicated. There are many different types of resampling algorithms, such as multinomial resampling, stratified sampling, systematic resampling and residual resampling. All resampling algorithms generate a set of m ordered random numbers $u_k \sim U[0, 1)$. These random numbers are then used to draw new particles $\mathbf{x}_t^{(k)*}$ using

$$\mathbf{x}_t^{(k)*} = \mathbf{x}_t^{(i)}, \tag{4.43}$$

with i and k such that (Hol et al., 2006)

$$u_k \in \left[\sum_{s=1}^{i-1} \omega_t^{(s)}, \sum_{s=1}^{i} \omega_t^{(s)}\right), \tag{4.44}$$

where $\omega_t^{(s)}$ are the normalized importance factors. This process ensures that particles with a previously high importance weight will be drawn more often (as a lot of random numbers will come to lie in the sampling interval of Equation 4.44) whereas for particles with a low importance weight, it is more unlikely that a random number will fall into the sampling interval, i.e. it is unlikely that this particle will be drawn.

The set of resampled particles $\boldsymbol{\chi}_t$ with equal weights $w_t^{(\cdot)}$ is then used as the new posterior (see Figure 4.3) in the next iteration step of the particle filter.

4.3 Comparison of Bayes Filters

When selecting a Bayes filter to estimate the state vector of a moving object, several aspects need to be taken into consideration. Among these aspects are the (non-)linearity of the dynamic model and the observation model, the distribution of the observation noise, the processing power available, and the type of observations (e.g. whether map information is integrated or not).

For linear observation and dynamic models and Gaussian distributions, the Kalman filter yields optimal results and is computationally efficient. In the case of non-linear models, a linearized version of the Kalman filter, such as the EKF or UKF must be chosen. The EKF has the advantage that it is more computationally efficient than the UKF. While the EKF uses a Taylor series expansion (up to the first term) to linearize, the UKF uses stochastic linearization in the form of sigma points. For strongly non-linear functional relationships, the UKF yields more accurate results than the EKF (e.g. when short ranges are processed). However, for many practical applications, the difference between the UKF and EKF can be neglected (Thrun et al., 2006).

Non-parametric filters, such as the particle filter, result in a higher computational load than Gaussian filters. However, they can be used to incorporate map information. This can be particularly useful for indoor positioning. When a floor plan is available, it can be used to constrain the state space (e.g. a particle cannot be located inside a wall, or particles cannot travel through walls).

Chapter 5

Multi-Sensor Fusion Architectures

In the previous chapter, Bayes filters were introduced as a tool to estimate an object's state vector from noisy navigation sensor data. However, when several navigation sensors are used, not only the type of Bayes filter must be chosen, but a suitable sensor fusion architecture must be found.

In Section 5.1 of this chapter the terminology used in conjunction with GNSS/INS integration will be explained. In Section 5.2 the concept of dual-sensor fusion will be extended to multi-sensor fusion and the three types of of multi-sensor fusion architectures will be explained. Subsection 5.2.1 introduces the concept of centralized integration, Subsection 5.2.2 deals with cascaded integration, and Subsection 5.2.3 explains federated integration architectures. Since an error-state extended Kalman filter will be used later in this thesis, the multi-sensor architectures in Subsections 5.2.1 to 5.2.3 will be illustrated using the example of error-state EKFs. In Section 5.3 the advantages and disadvantages of the three multi-sensor fusion architectures are summarized.

5.1 GNSS/INS Integration: Stages of Coupling

The most commonly combined navigation sensors are GNSS and INS. Since the properties of GNSS and INS are complementary, integrating or fusing observations from both sensors leads to a superior navigation solution.

INS have a high short-term accuracy and a high output rate. However, when used as a stand-alone system, errors quickly accumulate and the navigation solution starts to drift. GNSS, on the other hand, has a low short-term accuracy (for pseudorange positioning), a lower output rate, and is vulnerable to interference. Nevertheless, since it is a method of position fixing, it is stable in the long term and does not drift. By combining GNSS and INS, one ideally gets a navigation solution which has high output rate, a high short-term accuracy, does not drift, and is invulnerable to interference.

The architectures of how GNSS and INS can be integrated vary depending on which type of GNSS observations are used, whether or how corrections are fed back to the INS, and how or if the GNSS equipment is aided by the integration (Groves, 2013).

The type of GNSS observation defines whether we classify an integration system as loosely or tightly coupled. A *loosely coupled* GNSS/INS integration architecture, sometimes also referred to as position-domain integration or cascaded architecture, uses GNSS positions, GNSS velocity, and (if available) GNSS attitude information in the measurement update of the Bayes filter. A *tightly coupled* GNSS/INS integration architecture, also referred to as range-domain integration, uses GNSS pseudoranges, Doppler or carrier phase observations in the measurement update.

The terms *ultra-tightly* or *deeply coupled* refer to integration architectures where the GNSS receiver's carrier-phase or code tracking loops are aided by the integrated navigation solution (Farrell and Wendel, 2017).

The way in which corrections are fed back to the INS defines whether we speak of open-loop or closed-loop filters. In an *open-loop* implementation, corrections are estimated to correct the INS solution within the integration algorithm itself, but these corrections are not fed back to the INS. Consequently, an independent INS solution is maintained. This type of implementation is only suitable for highly precise IMUs. In a *closed-loop* implementation, the filter's estimated position, velocity, and attitude solution is fed back to aid the INS. When the filter is implemented as an error-state EKF, the closed-loop implementation minimizes both the size of the error state vector and the linearization errors. This type of implementation is also suitable for low-cost IMUs (Groves, 2013).

5.2 Multi-Sensor Integration

When two or more navigation sensors are used, there are three different architectures how they can be combined to optimally estimate the state vector: *centralized, cascaded,* and *federated* integration. These architectures will be explained in the following and examples are given for error-state EKFs.

5.2.1 Centralized Integration

In a centralized integration architecture, or a one-stage filter, measurements of navigation sensors are directly used as observations, i.e. they are not preprocessed. Figure 5.1 gives an overview of an error-state centralized integration architecture for integrating dead reckoning and position fixing systems. A dead reckoning system, e.g. an INS or odometry, is used to compute a reference navigation solution, which corresponds to the prediction step of Bayes filters. In the measurement update, measurements of several aiding sensors are used to correct the prediction. The aiding sensors can be either position fixing or dead reckoning sensors, but it is important that the raw measurements (e.g. the code pseudoranges or carrier phase observations in the case of GNSS, or the steering angle and wheel speeds of individual wheels of a robot or car) are processed. Tightly coupled GNSS/INS integration is an example of a centralized integration architecture. The dashed lines in Figure 5.1 indicate information that may be fed back from the integration filter to the individual sensors. In a closed-loop error state EKF, the reference correction is fed back to the reference navigation system. Moreover, estimated parameters of the state vector, such as a GNSS receiver clock bias or clock drift, may be fed back to the individual aiding sensors.

While a centralized filter theoretically leads to an optimal estimate of the state vector (Allerton and Jia, 2005), the filter equations become more and more complex the more sensors are added to the system as all systematic sensor errors have to be modelled in the same integration filter. Moreover, black box navigation systems, such as dual-antenna GNSS receivers that directly output heading information, cannot be used as aiding sensors in centralized integration architectures.

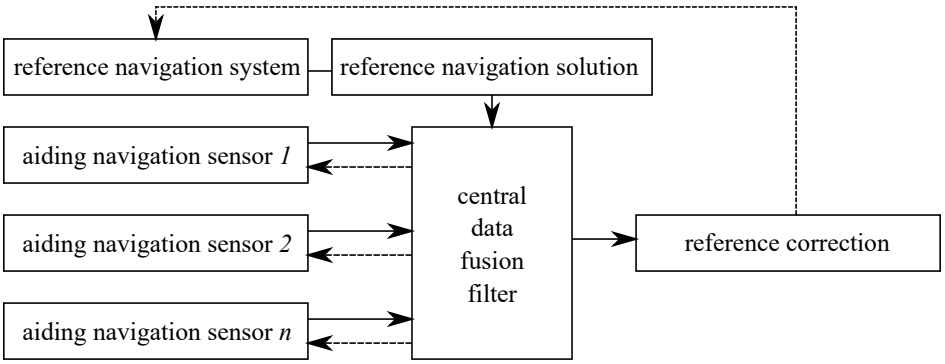


Figure 5.1: Error-state centralized integration architecture, adapted from Groves (2013). The dashed lines indicate information that may, but does not have to be, fed back.

5.2.2 Cascaded Integration

In a cascaded integration architecture, which is also called a two-stage or decentralized filter, the measurements of the individual navigation sensors are pre-processed to yield a position, velocity or attitude solution before they are fused together in a master filter.

Figure 5.2 gives an overview of an error-state cascaded integration architecture. Like in the centralized filter, a dead reckoning system is used to compute a reference navigation solution for the time update step. In contrast to the centralized filter, the raw measurements of the individual aiding sensors are pre-processed and the pre-processed navigation solutions are used as observations in the measurement update. In an error-state closed-loop implementation, a reference correction is estimated and fed back to the reference navigation processor. Loosely coupled GNSS/INS integration is an example of a cascaded filter architecture. There, the raw GNSS observations are processed to yield a position, velocity (and possibly attitude) solution. This navigation solution is then used in the measurement update to correct the reference navigation solution.

When compared to centralized filters, cascaded filters are less complex to implement and easier to extend to a multi-sensor integration scenario. A cascaded filter architecture can integrate new sensors while only requiring minimal modifications to the original filter (Allerton and Jia, 2005). Cascaded filters can process information from GNSS receivers which directly output a navigation solution. However, when the aiding navigation processors are also filters, the observations which are inserted into the integration filter suffer

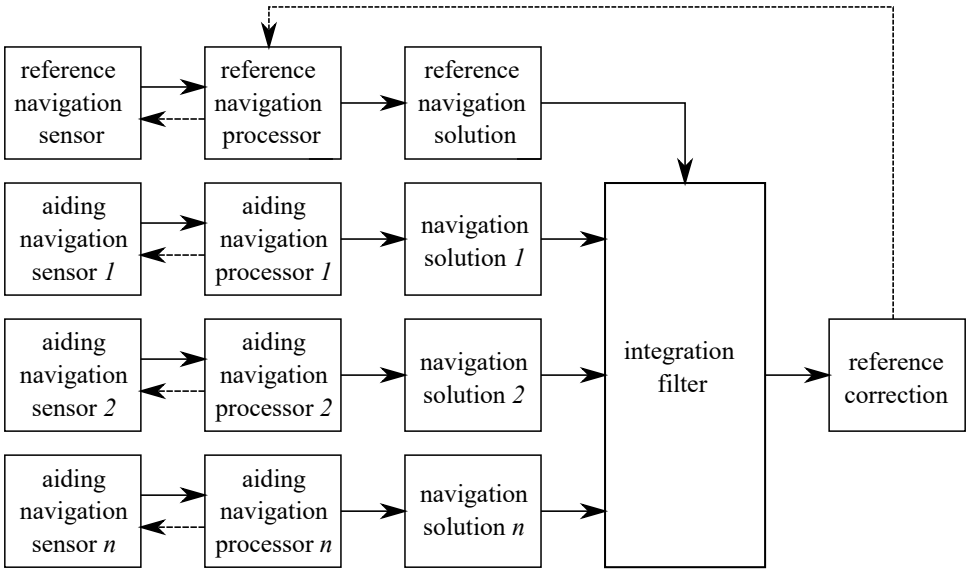


Figure 5.2: Error-state cascaded integration architecture, adapted from Groves (2013). The dashed lines indicate information that may, but does not have to be, fed back.

from time-correlated noise. This is a problem when the integration filter is a Kalman filter, as it assumes that the measurement noise is Gaussian. A simple method to account for this is to increase measurement noise covariance matrix $\mathbf{R}$ (Groves, 2013).

5.2.3 Federated Integration

A federated filter is a special type of a two-stage filter and was first proposed by Carlson (1990). It consists of several local filters that each output a navigation solution. The navigation solutions of the local filters are fused in a master filter. Carlson (1990) showed that by fusing several independent local filters, one can obtain a globally optimal solution.

Federated filters can share information, e.g. share a common reference navigation system or feed back information from the master fusion to the local filters. Depending on how the information is shared, we speak either of federated no reset, fusion reset, zero reset, or federated cascaded filters.

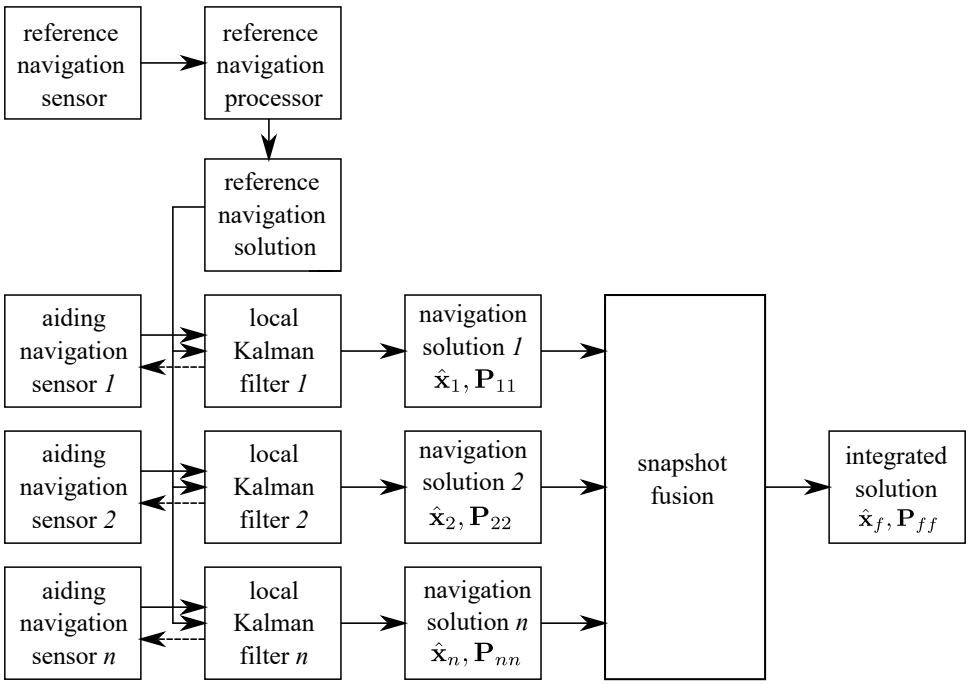


Figure 5.3: Error-state federated no-reset filter architecture, adapted from Groves (2013).

Federated No Reset

The architecture for a federated *no reset* filter is given in Figure 5.3. Here, the local filters share a common reference navigation system. Each local filter computes a navigation solution (e.g. using an EKF or UKF), which consists of a state vector $\hat{\mathbf{x}}_i$ and a covariance matrix $\mathbf{P}_{ii}$. The individual navigation solutions are fused using a snapshot (or single-epoch) fusing algorithm, e.g. a least squares adjustment. The integrated solution in terms of the fused state vector $\hat{\mathbf{x}}_f$ is obtained by

$$\hat{\mathbf{x}}_f = \left(\mathbf{H}^{\mathrm{T}}\mathbf{P}^{-1}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{P}^{-1}\begin{pmatrix}\hat{\mathbf{x}}_1\\ \hat{\mathbf{x}}_2\\ \vdots\\ \hat{\mathbf{x}}_n\end{pmatrix}, \tag{5.1}$$

where the design matrix $\mathbf{H}$ consists of n identity matrices with dimensionality $m \times m$, where m is the number of parameters that the state vectors $\hat{\mathbf{x}}_i$ contain. The covariance

matrix $\mathbf{P}$ has the following form:

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_{11} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{22} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{P}_{nn} \end{pmatrix}. \tag{5.2}$$

The covariance matrix of the integrated solution is obtained with

$$\mathbf{P}_{ff} = \left(\mathbf{H}^{\mathrm{T}}\mathbf{P}^{-1}\mathbf{H}\right)^{-1}. \tag{5.3}$$

Note that with the form of the covariance matrix in Equation 5.2, it is assumed that the individual navigation solutions are uncorrelated. However, when the local filters share a common reference navigation sensor as depicted in Figure 5.3, this is not the case. Therefore, the weighting of the local solutions within a federated no reset architecture and a common reference navigation sensor is suboptimal (Groves, 2013).

When the local filter solutions are assumed to be uncorrelated, the snapshot fusion algorithm can be simplified to (Groves, 2013):

$$\mathbf{P}_{ff} = \left(\sum_{i=1}^{n} \mathbf{P}_{ii}^{-1}\right)^{-1}, \tag{5.4}$$

$$\hat{\mathbf{x}}_f = \mathbf{P}_{ff} \sum_{i=1}^{n} \mathbf{P}_{ii}^{-1}\hat{\mathbf{x}}_i. \tag{5.5}$$

Federated Fusion Reset

In a federated *fusion reset* architecture (see Figure 5.4), the estimated state vector $\hat{\mathbf{x}}_f$ and the covariance matrix $\mathbf{P}_{ff}$ of the integrated snapshot solution are fed back to the local filters. When the local filters share information using a common reference navigation system, we introduce the information factors β_i, which must fulfil the following rule (Bancroft, 2010):

$$\beta_1 + \beta_2 + \ldots + \beta_n = 1. \tag{5.6}$$

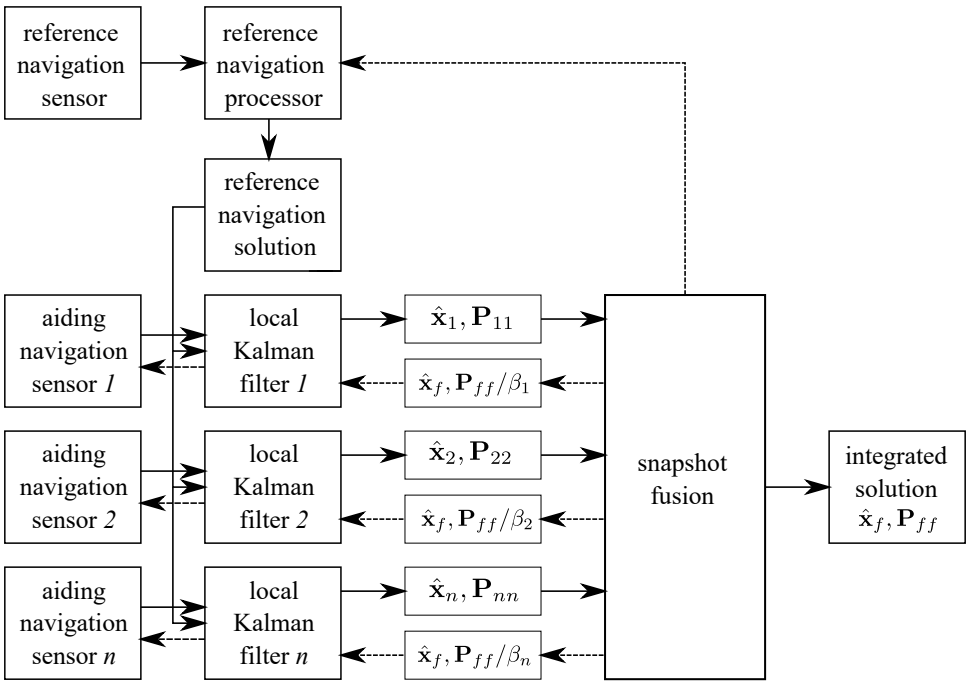


Figure 5.4: Error-state federated fusion-reset filter architecture, adapted from Groves (2013).

For n identical sensors, the information factors β_i are $\frac{1}{n}$.

The covariance matrix $\mathbf{P}_{ff}$ for the integrated solution is then computed from

$$\mathbf{P}_{ff} = \left(\sum_{i=1}^{n} \beta_i \mathbf{P}_{ii}^{-1} \right)^{-1}, \tag{5.7}$$

and the integrated state vector is then updated with (Bancroft, 2010):

$$\hat{\mathbf{x}}_f = \mathbf{P}_{ff} \sum_{i=1}^{n} \beta_i \mathbf{P}_{ii}^{-1} \hat{\mathbf{x}}_i. \tag{5.8}$$

In the federated fusion reset architecture, the estimated state vector $\hat{\mathbf{x}}_f$ is fed back to the local filters and serves as the new Taylor evaluation point. Also the covariance information is fed back, but it has to be scaled with the information factors according to

$$\mathbf{P}_{ii} = \frac{1}{\beta_i} \mathbf{P}_{ff}. \tag{5.9}$$

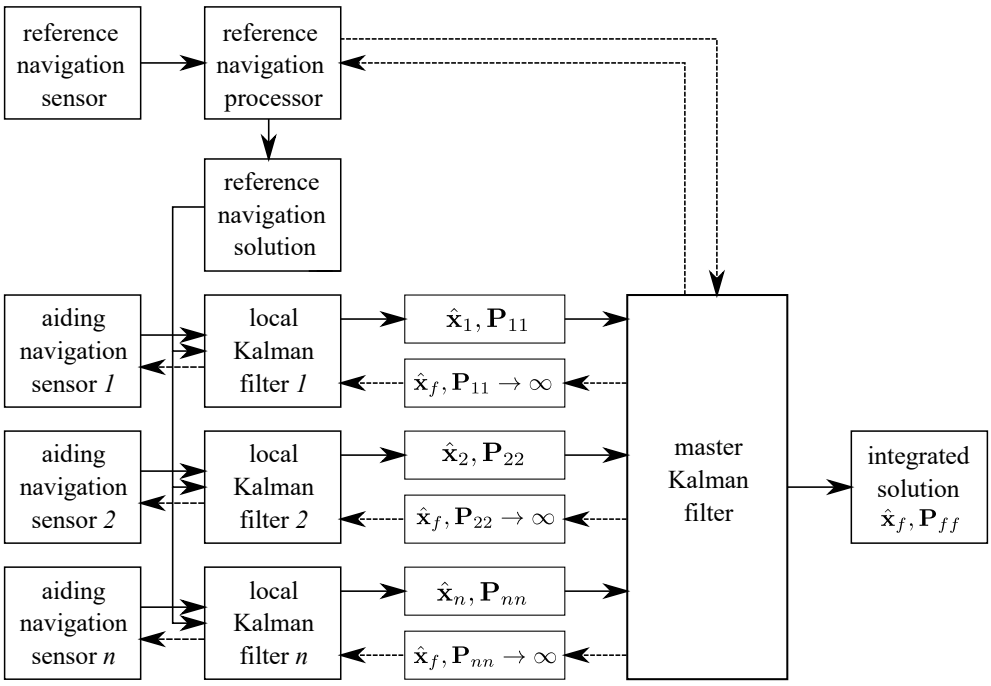


Figure 5.5: Error-state federated zero-reset filter architecture, adapted from Groves (2013).

Federated Zero Reset

In a federated *zero reset* architecture, the algorithm that fuses information from the local filters is not a snapshot fusion algorithm, but also a Kalman filter. This type of architecture is illustrated in Figure 5.5.

After each update of the master Kalman filter, a zero information reset is performed for the local filters. Each state information matrix of the local filters, i.e. the inverse of each state covariance matrix, $\mathbf{P}_{ii}^{-1}$, is set to zero (Bancroft, 2010). This corresponds to setting the individual covariances to infinity (or a very high covariance). An advantage of the zero reset architecture is that since the master fusion algorithm is a Kalman filter, it can combine the local Kalman filter solutions at different times, or whenever a new local solution becomes available (Carlson and Berarducci, 1994). By contrast, federated no reset or fusion reset architectures require the local filters to send solutions at the same epoch before the snapshot fusion algorithm can fuse them.

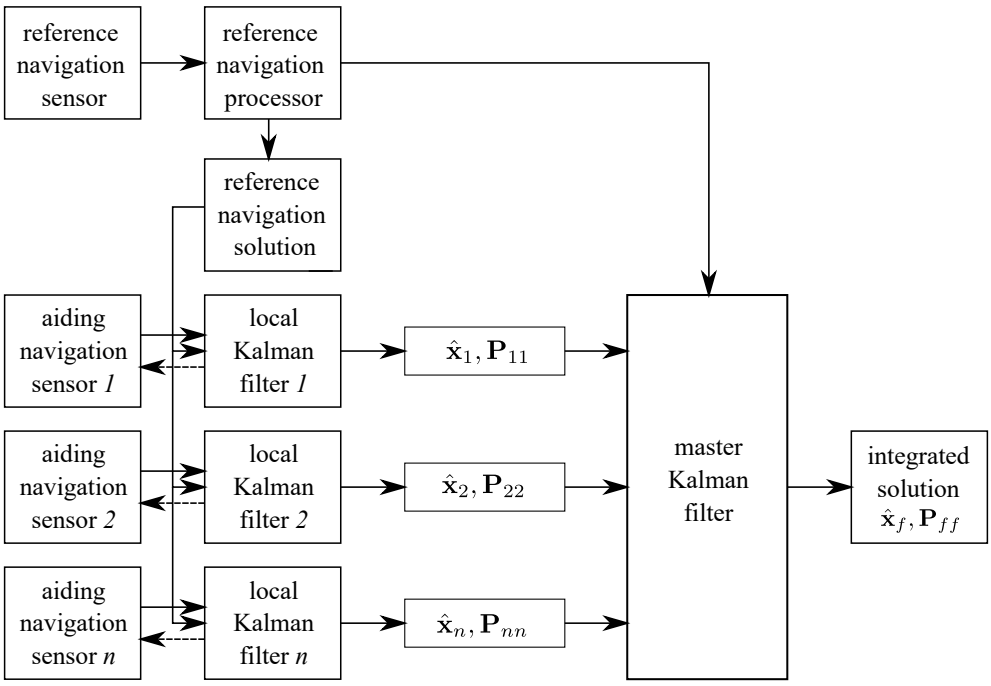


Figure 5.6: Error-state federated cascaded filter architecture, adapted from Groves (2013).

Federated Cascaded

A federated cascaded architecture, as shown in Figure 5.6, also uses a Kalman filter to fuse information from the local Kalman filters. However, it does not perform any resets, and therefore the master filter has to deal with time-correlated observation noise (Groves, 2013). Incorrect tuning in this filter architecture leads to a poor overall filter performance (Brown and Hwang, 2012).

All federated filters have the advantage that we can investigate the individual navigation solutions of the local filters before fusing them. However, a disadvantage of federated filters whose the local filters share a common reference navigation system is that the local filters then also share a common source of failure (Allerton and Jia, 2005).

5.3 Comparison of Multi-Sensor Integration Architectures

When selecting an integration architecture to fuse information from multiple sensors, the advantages and disadvantages of each integration architecture have to be taken into consideration.

Centralized and cascaded integration architectures differ in the type of observations that they process. Centralized integration architectures directly process measurements of navigation sensors (e.g. GNSS pseudoranges), while cascaded integration architectures use pre-processed information (e.g. a position computed from GNSS) as observations. Centralized integration architectures are therefore more complex and cannot be used to integrate information from black box systems, such as a heading output by a dual-antenna GNSS receiver. A cascaded filter needs at least four GNSS pseudoranges to compute a position that can be used as observation, while a centralized filter can also process three or less pseudoranges to support the navigation solution.

Federated integration architectures consist of several local filters that each compute a navigation solution. These navigation solutions of the local filters are then fused in a master filter. The local filters of the federated integration architectures can be either centralized or cascaded. The advantage of federated integration architectures is that they allow to analyse individual parts of the navigation solution, i.e. the outputs of the local filters. However, federated filters often share a common reference navigation sensor for the individual local filters and therefore also a common source of failure. Moreover, when a common reference navigation sensor is used, the individual solutions of the block filters are correlated. This causes a suboptimal weighting of the local solutions in the master filter.

For the multi-sensor positioning module developed within this thesis, federated and cascaded integration architectures will be investigated. For the federated architecture, different reference navigation sensors will be used for the local block filters so that the local solutions are uncorrelated. A centralized integration architecture will not be developed, as composting sites are open-sky environments so the case that less than four satellites are in view not likely to occur.

Part II

Developing a Multi-Sensor Positioning System for Tracked Vehicles

Chapter 6

Navigation Sensor Selection

This part of the thesis deals with the development of a multi-sensor positioning system for tracked compost turners. Chapter 6 covers the selection of navigation sensors, and Chapter 7 the development of a navigation filter using the sensors selected.

The first issue to be solved is the selection of navigation sensors that can be used in a positioning module for an automatic steering system for tracked compost turners. Section 6.1 gives an overview of the automatic steering system of which the positioning module is one part. Section 6.2 presents the methodology that is used in this thesis. In Section 6.3 the requirements which the positioning module should fulfil are outlined. Section 6.4 covers a conceptional pre-study for which several navigation sensors were mounted on a compost turner and used to record data at a composting site. Section 6.5 describes further investigations that were carried out to evaluate a dual-antenna GNSS receiver. Section 6.6 concludes the chapter with a list of sensors that were selected for the positioning module.

Some of the investigations presented in this chapter have already been published in Reitbauer et al. (2020a) and Reitbauer et al. (2020b).

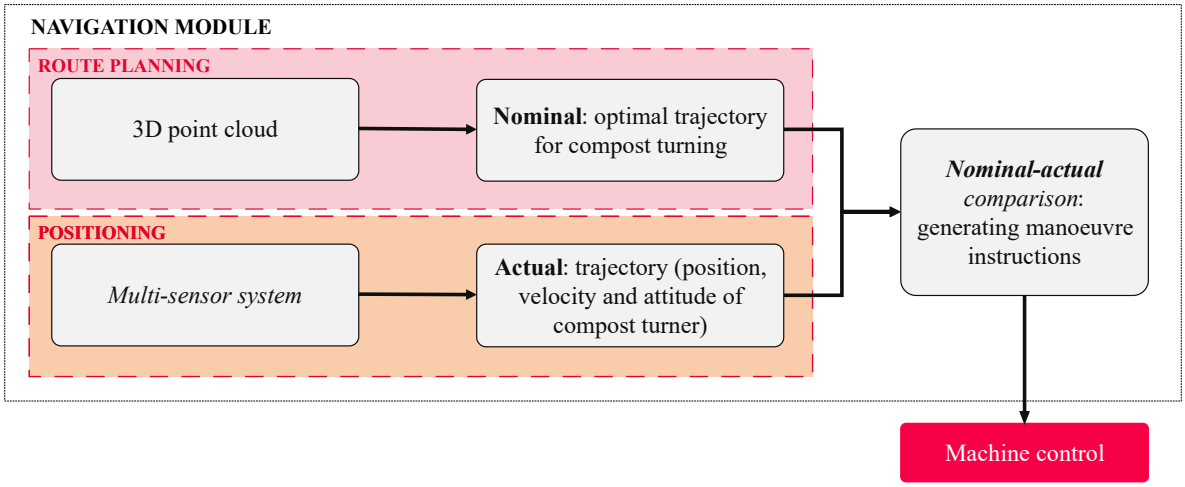


Figure 6.1: The navigation module as a part of the automatic steering system developed within the research project ANTON.

6.1 The Positioning Module as Part of an Automatic Steering System

The aim of this thesis is to develop a multi-sensor positioning module which is to be used in an automatic steering system for a tracked compost turner. Parts of the research were carried out within the research project "Autonomous Navigation for Tracked cOmpost turNers (ANTON)", which was funded by the Austrian Research Promotion Agency (FFG) in the course of the Austrian Space Applications Programme. Within this project, an automatic steering system for an electrically driven compost turner was developed.

An overview of the automatic steering system is given in Figure 6.1. The system consists of a navigation module and a machine control module. The navigation module is further divided into two submodules: a module for route planning and a module for positioning. The module for route planning computes the nominal routes or the optimal trajectory for the compost turner from a 3D point cloud of the composting site. The positioning module uses a multi-sensor system to compute the actual trajectory, i.e. it computes the state vector consisting of the position, velocity, and attitude of the compost turner in real-time. The actual position and attitude are then compared to the nominal trajectory and a steering angle (α) is computed. The definition of the steering angle is illustrated in Figure 6.2. When the nominal trajectory is on the left side seen from the vehicle's

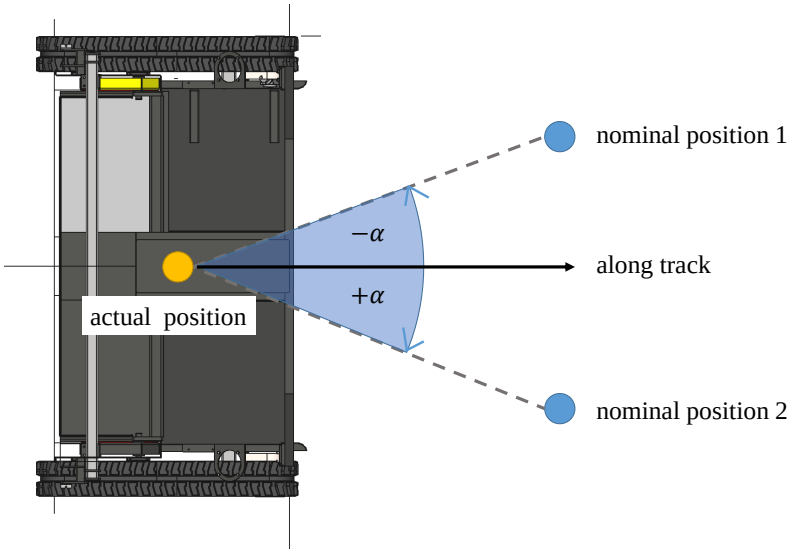


Figure 6.2: Definition of the steering angle α.

perspective, the steering angle is negative. When the nominal trajectory is on the right side, the steering angle is positive. The steering angle is passed on to the machine control module, which steers the machine towards the nominal position so that α becomes zero.

The machine control module was developed by the Institute of Logistics Engineering at Graz University of Technology. A prototype for the electrically driven self-driving compost turner was provided by the Austrian environmental engineering company Pusch & Schinnerl GmbH. The prototype vehicle is shown in Figure 6.3. Note that the prototype contains neither a seat nor a cabin for a driver (in contrast to conventional compost turners, such as the one depicted in Figure 1.2) as it is already designed to be steered automatically.

The navigation module was developed by the Institute of Geodesy at Graz University of Technology. The module for route planning was developed by Lercher (2020). The development of the positioning module is the main part of this thesis and will be outlined in the following.

Figure 6.3: Picture of the prototype of the electrically driven, self-driving compost turner.

6.2 Methodology

The following methodology is used to develop the positioning module:

1. The requirements that the module has to fulfil are analysed. This is outlined in Section 6.3.
2. A set of navigation sensors are selected and tested at a composting site. The sensor tests are presented in Sections 6.4 and 6.5.
3. Based on the results of the tests, a suitable set of navigation sensors is selected (Section 6.6).
4. Two navigation filter architectures which fuse the measurements of the sensors selected to estimate the state of the compost turner are developed (Chapter 7) and implemented in software (Section 7.3).
5. The filters developed are tested at a composting site (Chapter 8).
6. The accuracy achievable with these filters is evaluated by comparing the results to a reference trajectory (Chapter 9).
7. An evaluation is made of whether the requirements are met, and an outlook on future research is given in Chapter 10.

6.3 Requirements

This section outlines the requirements that the positioning module should fulfil.

The positioning module for the automatic steering system must be capable of determining the state vector of the compost turner in *real-time*, i.e. with a low latency. When we assume that the compost turner drives at a speed of 0.1 m/s, a latency of 250 ms would cause the position to lag 2.5 cm, which is still acceptable.

Both the *position and attitude* of the machine are needed to compute the steering angle, which is used to automatically steer the vehicle along the optimal route. Since the vehicle is steered automatically, the positioning system should ideally process information from *redundant* sensors so that outliers can be detected and eliminated.

The *accuracy* required for the automatic steering system for compost turners is less critical than for an automatic steering system used for crop rows. The accuracy requirements were discussed with the operators of a composting site. For the horizontal positioning accuracy, 10 cm were deemed sufficient. However, a higher accuracy relative to the windrows as well as a high accuracy for the yaw angle (1°) is needed. The high accuracy for the yaw angle is needed for two reasons. The first is that the trajectory that the compost turner follows determines the shape of the windrow after the turning process. When the compost turner follows a curved path, the windrow will also be curved. However, changing the shape or location of the windrows at the composting site is not desired. The second reason is that a wrong yaw angle leads to the computation of a wrong steering angle. When the compost turner, while turning a windrow, performs an unexpected rotation around its x_3-axis, the machine might get stuck in the heap of organic material.

In our project, the automatic steering system was required to work in *outdoor* environments. If the concept proves to be successful, it can also be extended to indoor environments at a later stage.

To fulfil the requirement of redundancy, the positioning system should fuse multiple sensors. Since current state-of-the-art automatic steering systems in outdoor environments use GNSS and INS, these components should be part of the system. Additionally, there should be a system that can sense the attitude of the compost turner with respect to

the windrow. Furthermore, data available from internal sensors of the machine that can be used for navigation purposes should be processed.

To obtain a highly accurate heading when the machine is standing or only moving slowly, either a highly precise IMU or a dual-antenna GNSS array can be used. Since Interferometric Fiber-Optic Gyroscope (IFOG) or Ring Laser Gyroscope (RLG) IMUs are costly, a GNSS dual-antenna approach will first be investigated.

To achieve the accuracy required for the position, a highly precise GNSS processing technique is required. Two techniques were taken into consideration: PPP and differential carrier phase positioning. Since PPP ambiguity resolution requires a longer convergence time than RTK and the positioning system should deliver a highly precise solution in real-time, we selected differential carrier phase positioning as the GNSS processing technique.

6.4 Pre-Study

A pre-study was conducted at a small composting site near Gnas, Austria in October 2019. The aim of the study was to find out which type of GNSS receiver (geodetic or low-cost) and which type of sensors that can sense orientation of the compost turner with respect to the windrows can be used in an automatic steering system. For the pre-study, a conventional compost turner (Backhus A30) was equipped with a selection of navigation sensors (Figure 6.4); these sensors should first collect the data to be analysed in post-processing. To evaluate the accuracy achievable, a reference trajectory was generated. In the following, the navigation sensor setup (Subsection 6.4.1), the collection of the reference trajectory (Subsection 6.4.2), the processing of the data collected (Subsection 6.4.3), and the results of the pre-study (Subsection 6.4.4) will be presented.

6.4.1 Navigation Sensors and Data Collection

The navigation sensors used in the pre-study are shown in Table 6.1. Two different pairs of GNSS antennas and receivers were mounted on the compost turner: two Topcon HiPer II geodetic receivers, which are capable of tracking both GPS and GLONASS on

Figure 6.4: Picture of the Backhus A30 used in the pre-study. The navigation sensors are mounted on an aluminium profile.

the carrier frequencies L1 and L2; and two uBlox M8T evaluation kits consisting of a low-cost patch antenna and receiver, which are capable of tracking GPS, GLONASS, and Galileo. Aditionally, a Leica Viva GS15 receiver, which supports GPS, GLONASS, and Galileo on L1, L2, and L5/E5a, was used as a static reference station. One MEMS IMU by XSens (MTi-G-710) was also mounted on the compost turner.

Table 6.1: List of sensors used in the pre-study.

Type of Sensor	Quantity	Name	Description
GNSS	2	Topcon HiPer II	geodetic GNSS receivers
	2	uBlox M8T	low-cost GNSS receivers
	1	Leica Viva GS15	geodetic GNSS receiver
IMU	1	XSens MTi-G-710	MEMS IMU
Image-based sensor	1	ZED	stereo camera
Range sensor	10	HCSR04	low-cost sonar
	3	VL53L1X	low-cost laser

To find out which sensor is best suited to determine the attitude of the compost turner with respect to the windrow, one stereo camera (ZED), ten low-cost sonar sensors (HCSR04) and three low-cost laser sensors (VL53L1X) were mounted on the compost turner. The investigations with the ranging and image-based sensors were carried out by Schmied (2020). In his master's thesis, he discusses new approaches to determine the attitude of the machine based on distance measurements to the triangle-shaped windrow. Therefore, the following investigations will focus on the GNSS receivers and the IMU.

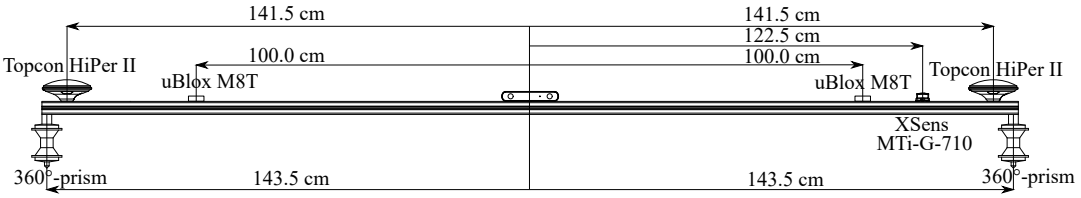


Figure 6.5: Sensor mounting on the aluminium frame, front view.

Table 6.2: Accuracies for angle and distance measurements of the total stations in tracking mode, taken from the manufacturer's data sheets.

Model	σ Angle Measurement	σ Distance Measurement
Leica Nova MS60	1" (0.3 mgon)	1 mm + 1.5 ppm
Leica TCRA1201	1" (0.3 mgon)	3 mm + 1.5 ppm

Figure 6.5 depicts the sensor mounting on the aluminium frame. The two Topcon HiPer II receivers were placed with a baseline of 283 cm between them; the two patch antennas of the uBlox M8T were mounted at a baseline of 200 cm. The XSens MTi-G-710 IMU was mounted at a distance of 122.5 cm from the center of the profile, between the uBlox M8T and the Topcon HiPer II. The IMU was configured to record accelerations and angular rates at 100 Hz. All GNSS receivers were configured to log raw data at 1 Hz.

6.4.2 Reference Trajectory

To evaluate the accuracy achievable with the individual navigation sensors, a reference trajectory had to be measured. Since differential carrier-phase positioning with GNSS typically yields an accuracy of a few centimetres, a reference trajectory with even higher accuracy had to be obtained. Therefore, two 360°-prisms were mounted on the aluminium profile on the compost turner and tracked with two robotic stations: a Leica TCRA1201 and a Leica Nova MS60. The accuracies as specified by the manufacturer (Leica Geosystems AG, 2006 and Leica Geosystems AG, 2020) for angle and distance measurements of both total stations in tracking mode are given in Table 6.2.

To compute the reference trajectory in the same coordinate frame as the navigation solution obtained through differential carrier phase positioning, a link between an ECEF and the local-level frame of the total station measurements had to be established. Therefore, three tripods with forced centering were set up at the composting site (see Fig-

(a) (b)

Figure 6.6: The reference trajectory was obtained by tracking two 360°prisms mounted on the compost turner with two robotic total stations: (a) a Leica TCRA1201, and (b) a Leica Nova MS60.

ure 6.7). The positions of these three tripods (denoted as "GPS001", "GPS002", and "GPS003" in Figure 6.7) were measured using the Leica GS15 receiver and the network-RTK service Echtzeit-Positionierung-Austria (EPOSA), which provides VRS data in the International Terrestrial Reference Frame (ITRF) 2014 at Epoch 2010.0. The coordinates of these three points were then transformed to a local-level frame. Then, three prisms were placed on the tripods and measured with the first total station, the Leica Nova MS60. A resection was performed to obtain the coordinates of the Leica Nova MS60. To obtain a high local accuracy of the total station network, the coordinates of the three prisms mounted on the tripods were measured anew with the MS60. These new local coordinates were then used to perform the resection with the second total station, the Leica TCRA1201.

For time synchronisation between the two total stations, both total stations were set to track the same 360°-prism, which was moved up and down. In Subsection 6.4.3 the process of how this time synchronisation was achieved in post-processing will be explained.

After both total stations tracked the same prism, the two 360°-prisms were mounted on the aluminium profile on the compost turner. While the compost turner turned the windrows, the two total stations were set to track the two prisms mounted on the compost turner (Figure 6.6).

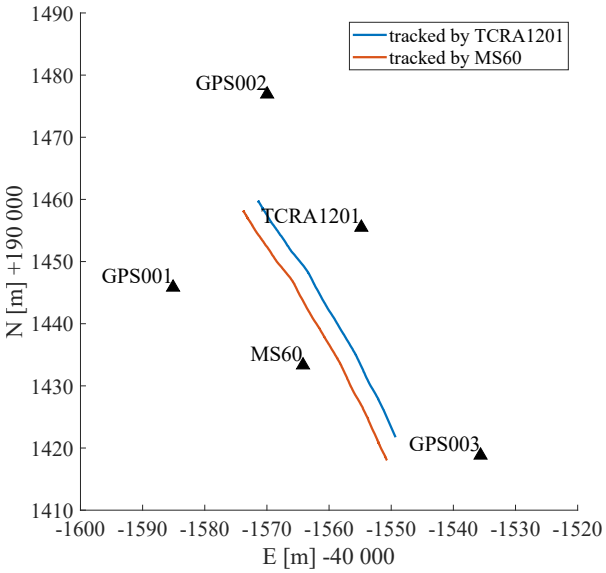


Figure 6.7: Positions of the tripods (depicted as black triangles) that were set up at the composting site. The reference trajectories that were tracked by the total stations are shown in blue and orange.

6.4.3 Post-Processing

This subsection describes how the data collected in the pre-study were post-processed.

GNSS

In a first step, the data collected by the GNSS receivers, which were stored in the receiver-proprietary formats, were converted to Receiver INdependent EXchange Format (RINEX) 3.03 files using the programme RTKCONV of the open-source software package RTKLIB (version 2.4.3 Emlid b28).

The data collected by the two Topcon Hiper II and the two uBlox M8T receivers were then post-processed in RTKPOST by RTKLIB using differential carrier phase positioning. The Leica Viva GS15 served as the base station.

As mentioned in Subsection 3.1.4, a baseline between two antennas that are mounted on a vehicle can be used to compute attitude information. Figure 6.8 shows the GNSS antenna setup of the pre-study, where the antennas were mounted along the x_2-axis of

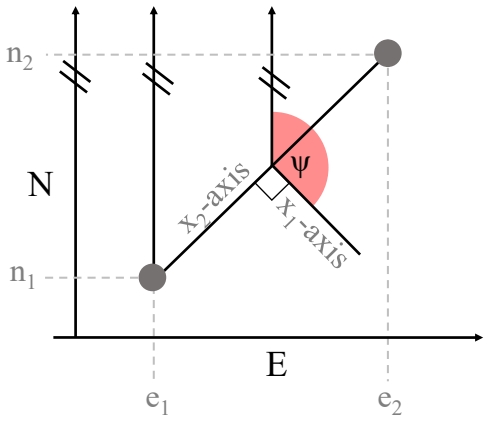


Figure 6.8: The heading ψ is computed from baseline coordinates in a local-level frame.

the compost turner. When the baseline coordinates are known in a local-level frame, the heading ψ of the machine can be computed from

$$\psi = \arctan\left(\frac{e_2 - e_1}{n_2 - n_1}\right) + \frac{\pi}{2}, \tag{6.1}$$

where $e_2 - e_1$ is the East coordinate difference, and $n_2 - n_1$ is the North coordinate difference of the baseline.

The baseline coordinate differences in East $(e_2 - e_1)$ and North $(n_2 - n_1)$ were obtained by processing the baseline between the two Topcon Hiper II and the baseline between the two uBlox M8T in RTKPOST using the positioning mode *Moving-Base*. Equation 6.1 was then used to compute the heading.

IMU

The specific forces recorded by the XSens MTi-G-710 were processed to compute the Euler angles roll (θ) and pitch (ϕ).

The specific force vector in the body frame can be written as

$$\mathbf{f}^b = \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix}. \tag{6.2}$$

When the IMU is stationary or travels at a constant velocity, we can compute roll (θ) and pitch (ϕ) from (Groves, 2013)

$$\theta = \arctan\left(\frac{-f_y}{-f_z}\right), \tag{6.3}$$

$$\phi = \arctan\left(\frac{f_x}{\sqrt{f_y^2 + f_z^2}}\right). \tag{6.4}$$

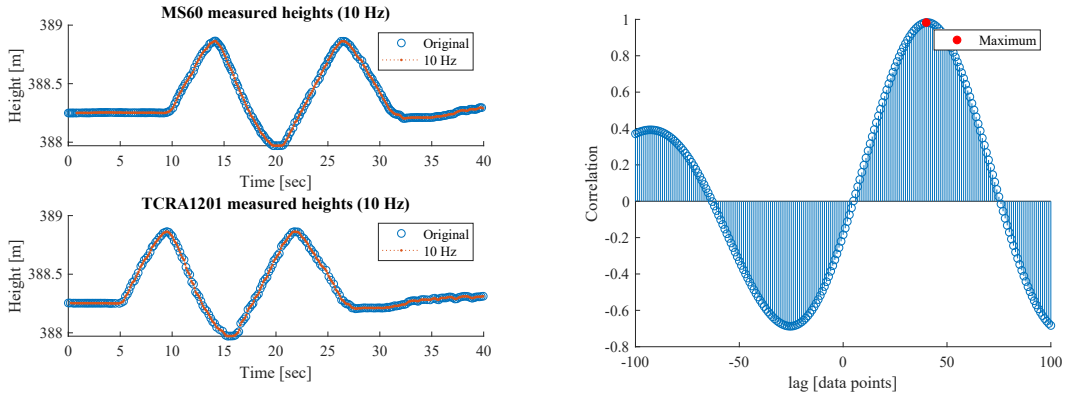
Since the compost turner is subject to strong vibrations while it turns the compost, the specific forces recorded at a rate of 100 Hz by the IMU have to be filtered. For post-processing, a moving average filter with a window size of 100 data points was chosen.

Reference Trajectory

The accuracy achievable with the individual sensors can be analyzed by comparing the trajectory computed with the observations of the navigation sensors to a reference trajectory. To compare two trajectories, not only a common coordinate reference frame, but also a common time scale is needed.

For time synchronisation, all trajectories were brought to GPS Time (GPST). The timestamps of the trajectories obtained from GNSS already refer to GPST time; since the XSens MTi-G-710 was connected to a GNSS patch antenna, the time series computed for roll and pitch also refer to GPST.

To obtain the reference trajectory in GPST, two processing steps are needed. In a first step, the measurements of the two total stations were brought to the same time frame by correlating the height time series of when the two total stations tracked the same prism which was moved up and down. This is illustrated in Figure 6.9. Part (a) shows the height time series of the prism, measured by the Leica Nova MS60 (upper time series) and the Leica TCRA1201 (lower time series). Both time series were sampled to the same data rate (10 Hz) and then correlated with each other. Part (b) of Figure 6.9 shows the result of the cross-correlation. The maximum correlation occured at a shift of 40 data points. The time series of the TCRA1201 was therefore shifted by 4.0 seconds (which



(a) **(b)**

Figure 6.9: Time synchronisation of the total stations: (a) height time series, (b) result of the cross-correlation.

corresponds to 40 data points at a sampling rate of 10 Hz) to refer to the same time as the MS60.

In a second step, the coordinate time series of the prisms that were tracked by the total stations during the tests were correlated with the coordinate time series obtained from differential carrier-phase positioning with the Topcon HiPer II receivers. The correlation was carried out in the same manner as described above for step 1; however, not the height time series, but rather the North and East time series were used here to compute the correlation and to determine the time offset.

Since the two prisms that were tracked by the total stations were mounted along the x_2-axis of the compost turner, a reference for both the heading (ψ) and the angle roll (θ) can be computed from the time-synchronised reference trajectory. The heading is computed according to Equation 6.1, where (n_1, e_1) are the coordinates of the prism that were tracked by the Leica Nova MS60 and (n_2, e_2) are the coordinates of the prism that were tracked by the Leica TCRA1201. The reference for the angle roll (θ) is computed as follows

$$\theta = \arctan\left(\frac{h_2 - h_1}{\sqrt{(n_2 - n_1)^2 + (e_2 - e_1)^2}}\right), \tag{6.5}$$

where (n_1, e_1, h_1) are the coordinates of the prism tracked by the Leica Nova MS60 and (n_2, e_2, h_2) are the coordinates of the prism tracked by the Leica TCRA1201.

6.4.4 Results

To evaluate the accuracy achievable with the individual sensors, the coordinate time series are compared to the reference trajectory. The two-dimensional (2D) Root Mean Square (RMS) error (RMS_{2D}) at epoch t is computed from

$$RMS_{2D}(t) = \sqrt{(n_r(t) - n_s(t))^2 + (e_r(t) - e_s(t))^2}, \tag{6.6}$$

where (n_r, e_r) are the North and East coordinates of the reference trajectory and (n_s, e_s) are the North and East coordinates of the sensor evaluated. Note that the coordinates of the reference were shifted to refer to the same position as the sensor.

The heading error at epoch t is computed from

$$\Delta\psi(t) = \psi_r(t) - \psi_s(t), \tag{6.7}$$

where ψ_r is the heading of the reference trajectory and ψ_s the heading of the sensor evaluated. The roll error at epoch t is computed from

$$\Delta\theta(t) = \theta_r(t) - \theta_s(t), \tag{6.8}$$

where θ_r is the roll of the reference trajectory and θ_s the roll of the sensor evaluated. Note that since the prisms for the reference trajectory were mounted along the x_2-axis of the compost turner, no reference for the pitch angle could be computed.

Figure 6.10 shows the 2D positioning error of one Topcon HiPer II and one uBlox M8T receiver as a time series. The Topcon HiPer II has a low positioning error and fulfils the accuracy requirements of $\leq$10 cm. The uBlox M8T has a large offset to the reference. Moreover, jumps with a magnitude of one wavelength (19 cm for L1) can be seen, which indicates wrongly fixed ambiguites.

Figure 6.11 shows the heading errors for the headings computed from the baseline between the two Topcon HiPer II (orange) and the two uBlox M8T (green), as well as the heading obtained from Visual Odometry (VO) with the ZED stereo camera (blue). When comparing the two headings obtained from the GNSS baseline, it can be seen that the heading from the Topcon HiPer II receivers is more accurate and has a lower standard

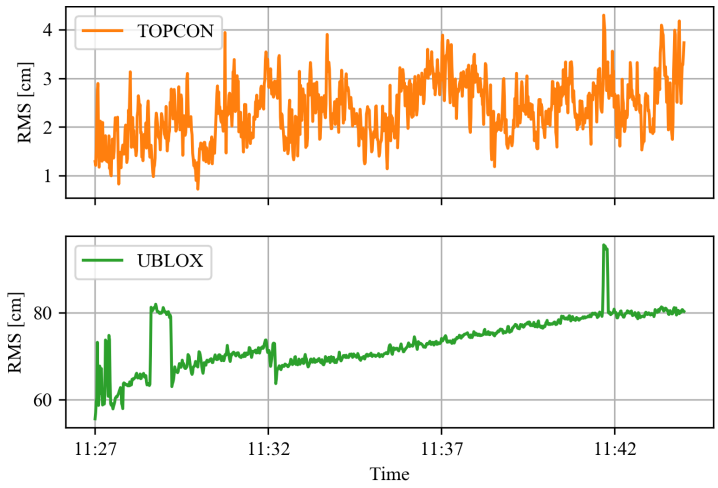


Figure 6.10: 2D positioning error of one Topcon HiPer II (orange) and one uBlox M8T (green).

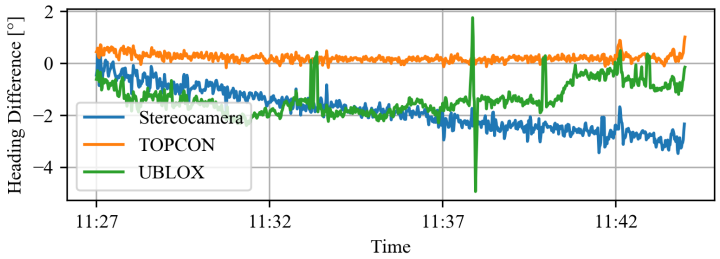


Figure 6.11: Heading error for the Topcon HiPer II (orange), uBlox M8T (green) and ZED stereo camera (blue).

deviation than the one obtained from the two uBlox M8T. The heading obtained from VO with the stereo camera was the most accurate heading that could be obtained with the image-based and range sensors (Schmied, 2020). However, it deteriorates with time, as VO is a dead reckoning technique where errors accumulate. The headings computed from the GNSS baselines are not affected by drift.

Figure 6.12 shows the deviation of the angle roll computed with the accelerations recorded by the XSens MTi-G-710 from the reference roll. The roll computed deviates less than 0.5° from the reference. Note that only the angle roll can be evaluated, as there is no reference for the angle pitch.

Table 6.3 shows the mean deviation from the reference (including the standard deviation) for the Topcon HiPer II receivers, the uBlox M8T receivers, the XSens MTi-G-710 and

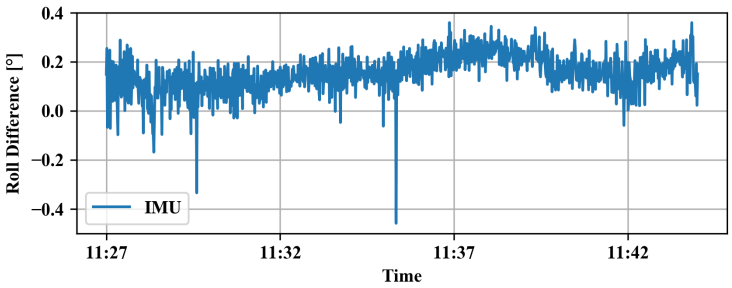


Figure 6.12: Roll error for the XSens MTi-G-710.

Table 6.3: Mean deviation from the reference trajectory and standard deviation.

Sensor	**RMS$_{2D}$ [cm]**	**$\Delta\theta$ [°]**	**$\Delta\psi$ [°]**
Topcon HiPer II	3.30 ± 0.60	0.30 ± 0.29	0.25 ± 0.14
uBlox M8T	25.20 ± 7.60	32.71 ± 0.29	1.49 ± 0.61
XSens MTi-G-710	-	0.20 ± 0.13	-
ZED	92.70 ± 11.00	0.59 ± 0.45	1.01 ± 0.83

the ZED stereo camera. The deviations are given in the form of a horizontal positioning error (RMS_{2D}), a roll error ($\Delta\theta$), and a heading error ($\Delta\psi$). Note that the horizontal positioning error for the GNSS receivers refers to both receivers of the same type, i.e. the values stated in the row for Topcon HiPer II are the mean errors for both Topcon HiPer II receivers; the same goes for the uBlox M8T.

The results in Table 6.3 show that with a high-end GNSS receiver such as the Topcon HiPer II, it is possible to meet the accuracy requirements of a horizontal positioning error of less than 10 cm and a heading error or less than 1°. The low-cost uBlox M8T does not fulfil the accuracy requirements. The XSens MTi-G-710 IMU allows computing an accurate roll angle, with a mean error of 0.20° and a standard deviation of only 0.13°. Stereo VO with the ZED allows computing both the position and the attitude of the compost turner. However, as VO is a technique of dead reckoning, errors accumulate; this leads to a mean horizontal deviation of 92.7 cm from the reference trajectory.

6.5 Evaluation of a Dual-Antenna GNSS Receiver

The pre-study conducted at the composting site showed that a high-end GNSS receiver is required to obtain a sufficiently accurate position and heading for the automatic steering of compost turners. The two Topcon HiPer II receivers used in the pre-study are only capable of tracking GPS and GLONASS. Furthermore, they cannot directly output a heading in real-time, as they are two separate receivers. Therefore, a Trimble BD992 dual-antenna receiver with two Trimble GA810 antennas was selected for the navigation module. This receiver is capable of tracking GPS, Galileo, GLONASS, and BeiDou. When the receiver is connected to the network-RTK service EPOSA (through an LTE modem), Virtual Reference Station (VRS) data can be received. The receiver can then output RTK positions and a highly accurate heading in real-time.

In June 2020, tests with the Trimble BD992 were conducted at the roof of the Geodesy building of Graz University of Technology. The aim of the tests was to evaluate the real-time accuracy achievable for both position and heading.

The sensor evaluation setup is shown in Figure 6.13. The two Trimble GA810 antennas, which were connected to the BD992, were mounted on an aluminium profile with a baseline of 2.8 m. The aluminium profile was connected to a motorized rotation stage (RAK 200), which was mounted on a pillar with known coordinates. The motorized rotation stage allows rotating the aluminium profile at different angular velocities. To generate a reference for both the position and heading, a 360°-prism was mounted below the position antenna. The prism was tracked with a Leica Nova MS60 total station. Note that this time it was sufficient to track only one prism, since the center of rotation was the pillar with known coordinates. The reference heading could therefore later be computed from the baseline of the tracked prism to the known pillar coordinates.

For the tests, the GNSS receiver was configured to output the position and heading at a rate of 10 Hz. In total, six time series were recorded. For the time series one to four, the aluminium profile was rotated by ±350°; in the rounds five and six, it was rotated twice by ±90°. The angular velocities varied between 1°/s and 20°/s. The detailed configurations are stated in Table 6.4.

(a) (b)

Figure 6.13: Sensor evaluation setup on a pillar with known coordinates. (a) A 360°-prism was mounted below the GNSS antenna for reference tracking. (b) A motorized rotation stage rotates the aluminium profile on which the sensors are mounted.

To compare the results to the reference trajectory, the positions output by the receiver were brought to the same local-level frame as the total station measurements. To synchronise the total station with GPST, the position time series of the prism were correlated with the position time series of the GNSS position antenna. The reference heading was computed according to Equation 6.1, where the coordinate differences in North and East are the coordinate differences of the position of the prism with respect to the known coordinates of the pillar.

Table 6.4 shows the measurement setup for the six test rounds as well as the accuracy achieved. The accuracy is given in the form of a horizontal positioning error (RMS_{2D}) computed according to Equation 6.6, a heading error ($\Delta\psi$) computed from Equation 6.7, and a height error (Δ_h). The height error at epoch t is computed from

$$\Delta_h(t) = h_r(t) - h_s(t), \tag{6.9}$$

where h_r is the height of the reference and h_s is the height measured by the sensor (Trimble BD992).

When taking a look at the results in Table 6.4, it can be seen that the mean horizontal positioning error is below 3 cm for every setup. The mean error in height is approximately 14 cm; the mean yaw error reaches a maximum of 0.1°.

Table 6.4: Horizontal positioning error, height error, and heading error for the Trimble BD992. Mean deviation from the reference trajectory and standard deviation are shown.

Round	**Rotation**	**Angular Velocity [°/s]**	**RMS_{2D} [cm]**	**Δh [cm]**	**$\Delta\psi$ [°]**
1	±350°	20	2.8 ± 1.8	-15.4 ± 0.8	0.0 ± 1.2
2	±350°	10	2.1 ± 1.1	-14.5 ± 0.6	0.0 ± 0.7
3	±350°	5	2.4 ± 1.1	-13.4 ± 0.7	0.1 ± 0.7
4	±350°	1	1.8 ± 0.4	-13.8 ± 0.6	0.1 ± 0.1
5	2x ± 90°	10	1.9 ± 1.0	-13.4 ± 0.8	0.0 ± 0.7
6	2x ± 90°	20	2.5 ± 1.3	-13.6 ± 0.9	0.1 ± 0.9

Both the standard deviation of the horizontal positioning error and the standard deviation of the heading error increase for increasing angular velocities. This can be explained by the method in which the reference trajectory is measured. The Leica Nova MS60 can track the prism at 7 Hz. The prism is 1.4 m away from the center of rotation. At an angular velocity of 1°/s, the prism moves at a speed of 0.02 m/s; at an angular velocity of 20°/s, it moves at 0.49 m/s. At a tracking rate of 7 Hz, a reference point is measured approximately every 7 cm when the profile rotates at 20°/s, and at every 3.5 mm when the profile rotates at 1°/s. Therefore, the reference trajectory is less precise at higher speeds or higher angular velocities. This leads to an increase in the standard deviation of the horizontal deviation from the reference and the heading deviation from the reference at higher angular velocities.

Compost turners operate at speeds lower than 10 cm/s, which translates to an angular velocity of ≤ 4°/s for the given measurement setup. Therefore, the requirement that the heading error should not exceed 1° is fulfilled with the Trimble BD992.

Figure 6.14 shows the detailed results of the fourth test round, where the aluminium profile was rotated at an angular velocity of $1°/s$. The plot on the left shows the positions estimated by the Trimble BD992 in a local-level frame. It can be seen that the profile was rotated by ±350°. The plots on the right side show the horizontal deviation from the reference trajectory and the heading deviation from the reference trajectory. The horizontal deviation is always below 4 cm and the heading deviation is lower than 0.8°.

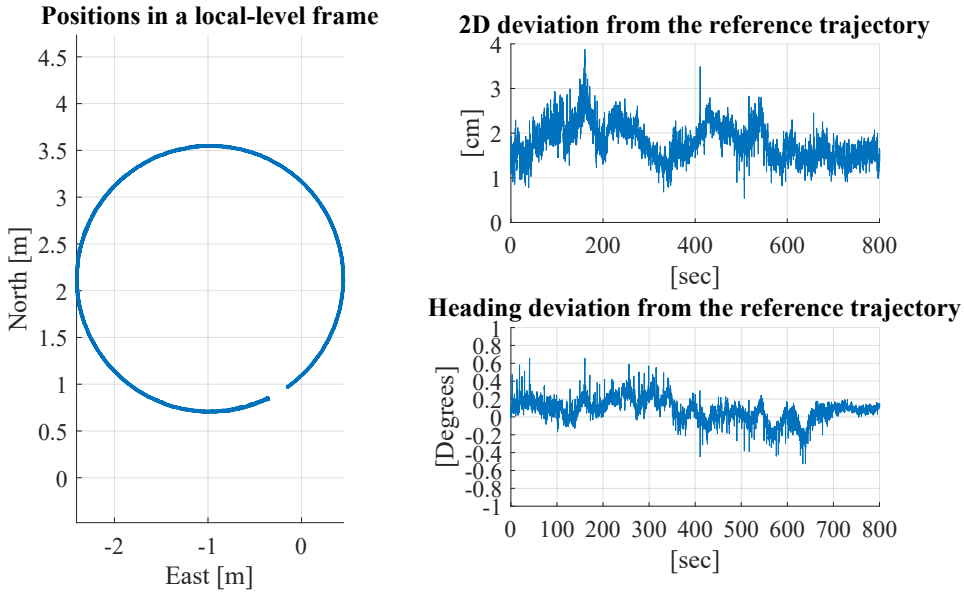


Figure 6.14: Results for round 4 with a rotation of ±350° at an angular velocity of $1°/s$. On the left: positions estimated by the Trimble BD992 in a local-level frame. On the right: horizontal deviation from the reference trajectory and heading deviation from the reference trajectory.

6.6 Navigation Sensors Selected

Based on the results of the pre-study, which was conducted with the available sensors, the following navigation sensors were selected for the automatic steering module of the compost turner:

- A Trimble BD992 multi-frequency multi-constellation dual-antenna GNSS receiver with two GA810 antennas and access to the network-RTK service EPOSA, as it provides sufficiently accurate horizontal position and heading in real-time;
- An XSens MTi-G-710 IMU, as it can provide sufficiently accurate roll and pitch angles;
- A ZED stereo camera, as it can provide both the position and the attitude of the machine;
- The two rotary encoders (2 Channel HallD SDN6.FK10.E05R) of the compost turner which are connected to the motors that drive the tracks, as they can be used to compute odometry information.

Note that the ZED stereo camera was selected even though it did not fulfil the accuracy requirements in the pre-study. In the pre-study, stereo VO, a dead reckoning technique, was used to compute the trajectory of the stereo camera. For the automatic steering system, however, a highly precise 3D point cloud of the composting site will be available, as it is needed to compute the nominal trajectories. By registering the point cloud recorded by the stereo camera to the 3D point cloud of the composting site using Normal Distributions Transform (NDT), the trajectory should no longer be affected by drift.

Chapter 7

Navigation Filter for Tracked Compost Turners

In the previous chapter, a dual-antenna GNSS receiver, a MEMS IMU, a stereo camera, and two magnetic rotary encoders were selected as navigation sensors to be used in the automatic steering module. Moreover, a highly precise 3D point cloud of the composting site (obtained either from Unmanned Aerial Vehicle (UAV) photogrammetry or from terrestrial laser scanning) is available, as it is needed for the route planning module. This point cloud can also be used for positioning, e.g. when the point cloud observed by the stereo camera is registered to the point cloud of the composting site.

To fuse information from all sensors mentioned, a suitable Bayes filter is needed. Since a fast navigation solution is needed in real time, an error-state Extended Kalman Filter (EKF) was selected as navigation filter as it is computationally efficient. However, there are different ways in which information coming from the navigation sensors can be combined to optimally estimate the position and attitude of the compost turner in real-time. In this chapter, two different filtering architectures will be presented: a modified federated integration architecture (Section 7.1) and a cascaded integration architecture (7.2). In Section 7.3, the software architecture is shown. The filters developed will later be evaluated in Part III of the thesis.

A part of what is presented in this chapter has already been published in Reitbauer and Schmied (2021a) and Reitbauer and Schmied (2021b).

7.1 Federated Integration

Federated integration architectures (see Section 5.2.3) usually integrate a common reference navigation solution with several aiding sensors in local Kalman filters. Each local Kalman filter estimates a navigation solution; the individual navigation solutions are then combined to obtain an integrated navigation solution.

For fusing the sensors selected for the navigation module, we propose a slightly different approach. The information should be split up into two parts which are fed into two respective local filters. Since both the IMU and the rotary encoders allow obtaining a dead reckoning solution, both can be used to propagate the state vector in the time update. So instead of sharing a common reference navigation solution between both local filters, a different reference navigation solution can be used for each local filter. The advantage of this approach is that the solutions estimated by the local filters are uncorrelated.

Figure 7.1 gives an overview of the proposed federated integration architecture. Observations from GNSS and the IMU are fused in the first local Kalman filter. Here, GNSS is used as an aiding sensor and the IMU as the reference navigation sensor. Information from the 3D point cloud of the composting site, the stereo camera, and the rotary encoders is fused in the second local Kalman filter, where information from the rotary encoders and a tailored odometry model for tracked vehicles is used to compute the reference navigation solution. Both local filters output the position and attitude of the compost turner. This information is then fused with a snapshot fusion algorithm.

Both local filters are error-state Kalman filters (see Section 4.1.3). The equations for the filters and the snapshot fusion algorithm will be explained in the following.

7.1.1 Local Filter Fusing GNSS and the IMU

The first local filter fuses observations of GNSS and the IMU. The error state vector of this filter has the following form:

$$\delta\mathbf{x}_t = \begin{pmatrix} \delta\mathbf{p} & \delta\mathbf{v} & \delta\mathbf{e} & \delta\mathbf{b_f} & \delta\mathbf{b_g} \end{pmatrix}^T, \tag{7.1}$$

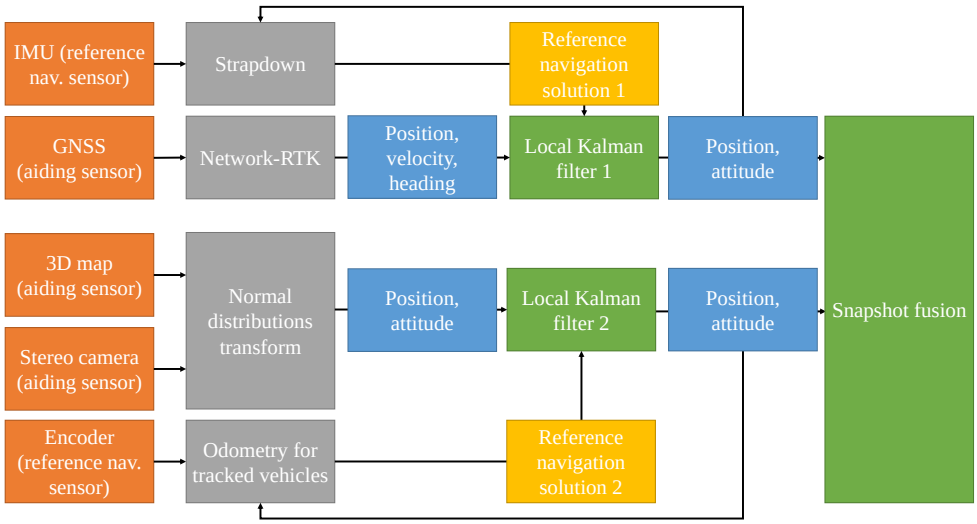


Figure 7.1: Federated filter architecture for fusing observations of an IMU, a GNSS receiver, a 3D map, a stereo camera, and encoders with an error-state EKF.

where $\delta\mathbf{p} = \begin{pmatrix}\delta N & \delta E & \delta D\end{pmatrix}^T$ stands for the position error in a local-level (North-East-Down)-frame, $\delta\mathbf{v}$ represents the velocity error in a local-level (North-East-Down)-frame, $\delta\mathbf{e}$ are the attitude errors regarding Euler angles (roll, pitch, yaw). $\delta\mathbf{b_f}$ is the error of the accelerometer biases, and $\delta\mathbf{b_g}$ is the error of the gyroscope biases of the IMU in three axes.

Note that the state vector $\mathbf{x}_t$ contains the position in the ECEF-frame in the form of geographic coordinates (φ, λ, h). The error state vector $\delta\hat{\mathbf{x}}_t$ is used to update the state vector $\hat{\mathbf{x}}_t$ according to Equation 4.25. Since the positions of the error state vectors are given in a local-level frame in metres and the positions of the state vector in geographic coordinates, the first three elements of the state vector have to be updated using

$$\begin{pmatrix}\hat{\varphi}_t \\ \hat{\lambda}_t \\ \hat{h}_t\end{pmatrix} = \begin{pmatrix}\tilde{\varphi}_{t-1} \\ \tilde{\lambda}_{t-1} \\ \tilde{h}_{t-1}\end{pmatrix} - \begin{pmatrix}\delta\hat{N}/(M_{ell} + \tilde{h}) \\ \delta\hat{E}/[(N_{ell} + \tilde{h}) \cdot \cos\tilde{\varphi}] \\ -\delta\hat{D}\end{pmatrix}, \tag{7.2}$$

where M_{ell} is the radius of curvature in the meridian and N_{ell} is the radius of curvature of the ellipsoid in the prime vertical plane (Jekeli, 2001). The radius of curvature in the

meridian M_{ell} is defined as (Jekeli, 2001)

$$M_{ell} = \frac{a\,(1-e^2)}{\left(1-e^2\sin^2\varphi\right)^{3/2}}, \tag{7.3}$$

where a is the semi-major axis of the ellipsoid and $e = \sqrt{2f - f^2}$ is the first eccentricity of the ellipsoid (with f being the flattening of the ellipsoid). The radius of curvature in the prime vertical plane N_{ell} is computed from (Jekeli, 2001)

$$N_{ell} = \frac{a}{\sqrt{1-e^2\sin^2\varphi}}. \tag{7.4}$$

In the following, the time update with the IMU and the measurement update with GNSS will be explained.

Time Update with the IMU

The time update with the IMU is performed in three steps. In the first step, the transition matrix $\mathbf{\Phi}_{t-1}$ is computed. In a second step, we compute the system noise matrix $\mathbf{Q}_{t-1}$. In the last step, the state vector and the covariance matrix are predicted to the current epoch t. In the following, these three steps will be described.

Transition Matrix

To obtain the system model, we can apply the following linear model to the error-state vector (Groves, 2013):

$$\delta\dot{\mathbf{x}}(t) = \mathbf{F}(t)\delta\mathbf{x}(t) + \mathbf{G}(t)\mathbf{w}_s(t), \tag{7.5}$$

where $\mathbf{F}$ is the system matrix, $\mathbf{G}$ is the system noise distribution matrix, and $\mathbf{w}_s$ is the system noise vector.

When we know the system matrix $\mathbf{F}_{t-1}$, we can compute the transition matrix $\mathbf{\Phi}_{t-1}$ from the following power-series expansion:

$$\mathbf{\Phi}_{t-1} = \mathbf{I} + \mathbf{F}_{t-1}\tau + \frac{\mathbf{F}_{t-1}^2}{2}\tau^2 + \frac{\mathbf{F}_{t-1}^3}{6}\tau^3, \tag{7.6}$$

where τ stands for the propagation interval from epoch $t-1$ to epoch t.

For our state vector, the system model has the following form (Noureldin et al., 2013):

$$\begin{pmatrix} \delta\dot{\mathbf{p}} \\ \delta\dot{\mathbf{v}} \\ \delta\dot{\mathbf{e}} \\ \delta\dot{\mathbf{b}}_\mathbf{f} \\ \delta\dot{\mathbf{b}}_\mathbf{g} \end{pmatrix} = \mathbf{F}_{IMU,t-1} \begin{pmatrix} \delta\mathbf{p} \\ \delta\mathbf{v} \\ \delta\mathbf{e} \\ \delta\mathbf{b}_\mathbf{f} \\ \delta\mathbf{b}_\mathbf{g} \end{pmatrix} + \mathbf{G}\mathbf{w}_s. \tag{7.7}$$

The system matrix $\mathbf{F}_{IMU,t-1}$, which is sometimes also called dynamic matrix, has the following block structure (Wendel, 2011):

$$\mathbf{F}_{IMU,t-1} = \begin{pmatrix} \mathbf{F}_{11} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_{22} & \mathbf{F}_{23} & \mathbf{R}_b^l & \mathbf{0} \\ \mathbf{F}_{31} & \mathbf{F}_{32} & \mathbf{F}_{33} & \mathbf{0} & \mathbf{R}_b^l \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{44} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{F}_{55} \end{pmatrix}, \tag{7.8}$$

where each block $\mathbf{F}_{ii}$ stands for a 3×3 matrix. The matrix $\mathbf{I}$ is the 3×3 identity matrix; the matrix $\mathbf{0}$ is a 3×3 matrix containing zeros. $\mathbf{R}_b^l$ is the rotation matrix from the body frame to the local-level frame. The matrix $\mathbf{R}_b^l$ is the transpose of the matrix $\mathbf{R}_l^b$ (Equation 2.8).

The individual matrices $\mathbf{F}_{ii}$ are obtained by linearising the equations for inertial navigation (Subsection 3.2.3). To give an example of how this linearisation is achieved, we will take a look at the equation describing the change of the latitude $\dot{\varphi}$:

$$\dot{\varphi} = \frac{v_N}{M_{ell} + h}. \tag{7.9}$$

The change in latitude mainly depends on the velocity error in North (δv_N) and the height error (δh). We assume that the radius of curvature in the meridian M_{ell} is constant and develop the Taylor series expansion of the above equation as (Wendel, 2011)

$$\delta\dot{\varphi} = \frac{\partial\dot{\varphi}}{\partial h} \cdot \delta h + \frac{\partial\dot{\varphi}}{\partial v_N} \cdot \delta v_N. \tag{7.10}$$

As we want to express the height error for the Down-component of the error state vector (δD) and $\delta h = -\delta D$, we rewrite the above equation to:

$$\delta\dot{\varphi} = \frac{\partial\dot{\varphi}}{\partial h} \cdot -\delta D + \frac{\partial\dot{\varphi}}{\partial v_N} \cdot \delta v_N = \frac{v_N}{(M_{ell} + h)^2}\delta D + \frac{1}{M_{ell} + h}\delta v_N. \tag{7.11}$$

Since we want to express the position error in North δN in metres, we have to multiply Equation 7.11 by the factor $(M_{ell} + h)$:

$$\delta\dot{N} = (M_{ell} + h) \cdot \delta\dot{\varphi} = \frac{v_N}{(M_{ell} + h)}\delta D + \delta v_N. \tag{7.12}$$

The first row of the matrix $\mathbf{F}_{IMU,t-1}$ therefore has the form

$$\begin{pmatrix} 0 & 0 & \frac{v_N}{(M_{ell}+h)} & 1 & 0 & 0 & \mathbf{0}_{1\times 9} \end{pmatrix}. \tag{7.13}$$

For the detailed derivations of all components of the dynamic matrix, the reader is referred to Wendel (2011). The individual matrices with the linearised equations that are part of the large dynamic matrix in Equation 7.8 are stated in the following, as they are fundamental for the filter used in this thesis. The equations are largely drawn from Wendel (2011) and Noureldin et al. (2013).

The matrix $\mathbf{F}_{11}$ has the form

$$\mathbf{F}_{11} = \begin{pmatrix} 0 & 0 & \frac{v_N}{M_{ell}+h} \\ \frac{v_E \tan\varphi}{M_{ell}+h} & 0 & \frac{v_E}{N_{ell}+h} \\ 0 & 0 & 0 \end{pmatrix}, \tag{7.14}$$

where v_N and v_E are the velocities of the previously estimated state vector in North and East; φ is the latitude and h is the height component of the previously estimated state vector. M_{ell} and N_{ell} are the radii of curvature in the meridian and in the prime vertical plane (Equations 7.3 and 7.4).

$\mathbf{F}_{22}$ is computed from

$$\mathbf{F}_{22} = \begin{pmatrix} \frac{v_D}{M_{ell}+h} & -2\omega_e \sin\varphi - \frac{2v_E \tan\varphi}{N_{ell}+h} & \frac{v_N}{M_{ell}+h} \\ 2\omega_e \sin\varphi + \frac{v_E \tan\varphi}{N_{ell}+h} & \frac{v_D}{N_{ell}+h} + \frac{v_N \tan\varphi}{N_{ell}+h} & 2\omega_e \cos\varphi + \frac{v_E}{N_{ell}+h} \\ -\frac{2v_N}{M_{ell}+h} & -2\omega_e \cos\varphi - \frac{2v_E}{N_{ell}+h} & 0 \end{pmatrix}, \tag{7.15}$$

where v_N, v_E, v_D are the North-, East-, and Down- components of the velocity of the previously estimated state vector; φ is the latitude and h the height component of the state vector, and ω_e is the Earth's rotation rate. Again, M_{ell} and N_{ell} are the radii of curvature in the meridian and in the prime vertical plane (Equations 7.3 and 7.4).

To obtain the matrix $\mathbf{F}_{23}$, the measured specific forces $\mathbf{f}^b$ of the IMU have to be brought to the local level frame

$$\mathbf{f}^l = \begin{pmatrix} f_N \\ f_E \\ f_D \end{pmatrix} = \mathbf{R}_b^l \mathbf{f}^b. \tag{7.16}$$

The matrix $\mathbf{F}_{23}$ is the negative skew-symmetric axiator of $\mathbf{f}^l$

$$\mathbf{F}_{23} = \begin{pmatrix} 0 & f_D & -f_E \\ -f_D & 0 & f_N \\ f_E & -f_N & 0 \end{pmatrix}. \tag{7.17}$$

$\mathbf{F}_{31}$ is obtained from

$$\mathbf{F}_{31} = \begin{pmatrix} -\frac{\omega_e \sin\varphi}{M_{ell}+h} & 0 & 0 \\ 0 & 0 & 0 \\ -\frac{\omega_e \cos\varphi}{M_{ell}+h} & 0 & 0 \end{pmatrix}, \tag{7.18}$$

where ω_e is the rotation rate of the Earth, and φ and h are the latitude and height of the previously estimated state vector. M_{ell} is the radius of curvature in the meridian (Equation 7.3).

$\mathbf{F}_{32}$ is computed as follows:

$$\mathbf{F}_{32} = \begin{pmatrix} 0 & \frac{1}{N_{ell}+h} & 0 \\ -\frac{1}{M_{ell}+h} & 0 & 0 \\ 0 & -\frac{\tan\varphi}{N_{ell}+h} & 0 \end{pmatrix}, \tag{7.19}$$

where h is the height component and φ the latitude of the previously estimated state vector. N_{ell} is the radius of curvature in the prime vertical plane (Equation 7.4) and M_{ell} is the radius of curvature in the meridian (Equation 7.3).

$\mathbf{F}_{33}$ has the form

$$\mathbf{F}_{33} = \begin{pmatrix} 0 & -\omega_e \sin\varphi - \frac{v_E \tan\varphi}{N_{ell}+h} & \frac{v_N}{M_{ell}+h} \\ \omega_e \sin\varphi + \frac{v_E \tan(\varphi)}{N_{ell}+h} & 0 & \omega_e \cos\varphi + \frac{v_E}{N_{ell}+h} \\ -\frac{v_N}{M_{ell}+h} & -\omega_e \cos\varphi - \frac{v_E}{N_{ell}+h} & 0 \end{pmatrix}, \tag{7.20}$$

where φ and h are the latitude and height components of the state vector, ω_e is the rotation rate of the Earth, and N_{ell} and M_{ell} (Equations 7.3 and 7.4) are the radii of curvature in the prime vertical plane and in the meridian.

Note that the matrices $\mathbf{F}_{44}$ and $\mathbf{F}_{55}$ are only needed when a bias drift for the accelerometer and gyroscopes is modelled. A bias drift can be modelled as a first order Gauss-Markov process (Wendel, 2011)

$$\dot{b} = -\frac{1}{\tau_b} b + n, \tag{7.21}$$

where b is the bias and $\dot{b}$ is the bias drift; τ_b is the correlation time, and n is the white noise. The bias drift has to be modelled for MEMS IMUs. For ring laser or fibre optic gyroscopes, the bias drift can be neglected and the matrices $\mathbf{F}_{44}$ and $\mathbf{F}_{55}$ are zero.

When a bias drift is modelled, the matrices $\mathbf{F}_{44}$ and $\mathbf{F}_{55}$ account for the systematic part of the sensor biases (the Gaussian part is modelled in the system noise matrix). The matrix $\mathbf{F}_{44}$ has the following form (Noureldin et al., 2013):

$$\mathbf{F}_{44} = \begin{pmatrix} -\frac{1}{\tau_{fx}} & 0 & 0 \\ 0 & -\frac{1}{\tau_{fy}} & 0 \\ 0 & 0 & -\frac{1}{\tau_{fz}} \end{pmatrix}, \tag{7.22}$$

where τ_{fx}, τ_{fy}, and τ_{fz} are the correlation times of the autocorrelation sequences of the accelerometer bias errors in three axes. The matrix $\mathbf{F}_{55}$ has the form (Noureldin et al., 2013)

$$\mathbf{F}_{55} = \begin{pmatrix} -\frac{1}{\tau_{gx}} & 0 & 0 \\ 0 & -\frac{1}{\tau_{gy}} & 0 \\ 0 & 0 & -\frac{1}{\tau_{gz}} \end{pmatrix}, \tag{7.23}$$

where τ_{gx}, τ_{gy}, and τ_{gz} are the correlation times of the autocorrelation sequences of the gyroscope bias errors in three axes.

The matrices in Equations 7.14 to 7.23 are used to compute the dynamic matrix $\mathbf{F}_{IMU,t-1}$ (Equation 7.8). With the dynamic matrix, the transition matrix $\mathbf{\Phi}_{t-1}$ can then be computed according to Equation 7.6.

System Noise Matrix

To obtain the system noise matrix, we first model the covariance matrix for the IMU as follows:

$$\mathbf{C}_{IMU} = \begin{pmatrix} \mathbf{C_f} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{C_\omega} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{C}_{\mathbf{b}f} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{C}_{\mathbf{b}g} \end{pmatrix}, \tag{7.24}$$

where $\mathbf{C_f}$ is the 3×3 matrix that has the variances of the accelerometer measurements $(\sigma^2_{fx}, \sigma^2_{fy}, \sigma^2_{fz})$ on the main diagonal, and $\mathbf{C_\omega}$ is the 3×3 matrix with the variances of the gyroscope measurements $(\sigma^2_{\omega x}, \sigma^2_{\omega y}, \sigma^2_{\omega z})$ on the main diagonal. The matrices $\mathbf{C}_{\mathbf{b}f}$ and $\mathbf{C}_{\mathbf{b}g}$ contain the variances of the white noise-component of the accelerometer bias and gyroscope bias on the main diagonal.

The time-independent part of the system noise matrix $\bar{\mathbf{Q}}$ is obtained from a variance propagation with

$$\bar{\mathbf{Q}} = \mathbf{A} \cdot \mathbf{C}_{IMU} \cdot \mathbf{A}^T, \tag{7.25}$$

with $\mathbf{A}$ as follows:

$$\mathbf{A} = \begin{pmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{R}^l_b & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{R}^l_b & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \end{pmatrix}, \tag{7.26}$$

where $\mathbf{R}_b^l$ is the 3×3 rotation matrix describing the rotation from the body-frame to the local-level-frame (see Equation 2.8).

The final system noise matrix is obtained from (Wendel, 2011)

$$\mathbf{Q}_{IMU,t-1} = (\mathbf{I} + \mathbf{\Phi}_{t-1})\bar{\mathbf{Q}}(\mathbf{I} + \mathbf{\Phi}_{t-1})^T \frac{\tau}{4}, \tag{7.27}$$

where τ is the propagation interval from epoch $t-1$ to t.

Predicting the State Vector and Covariance

To predict the state vector to the current epoch t using IMU measurements, the position changes, velocity, and attitude changes are computed with a strapdown algorithm (see Section 3.2.3).

The covariance matrix of the state vector is updated with the previously computed transition matrix $\mathbf{\Phi}_{t-1}$ and the system noise matrix $\mathbf{Q}_{t-1}$. The update is computed with Equation 4.9.

Measurement Update with GNSS

The dual-antenna Trimble BD992 GNSS receiver is used as an aiding sensor in the measurement update of the filter. When the receiver is connected to an LTE modem, it can receive and process network-RTK corrections in real time. The receiver outputs the position vector $\mathbf{p}_{GNSS,t} = \begin{pmatrix} \varphi_{GNSS,t} & \lambda_{GNSS,t} & h_{GNSS,t} \end{pmatrix}^T$, the velocity vector $\mathbf{v}_{GNSS,t}$, and the heading $\psi_{GNSS,t}$ at each epoch t.

For the measurement update of the error-state EKF, the reduced observations have to be computed (Equation 4.23). Since the IMU and the GNSS antenna cannot be located at the same place, a lever arm correction has to be applied. The lever arm $\mathbf{l}_{GNSS}^b$ from the IMU to the GNSS antenna is known in the body frame. It can be transformed to the local level frame as follows:

$$\mathbf{l}_{GNSS}^l = \begin{pmatrix} l_N \\ l_E \\ l_D \end{pmatrix} = \mathbf{R}_b^l \mathbf{l}_{GNSS}^b. \tag{7.28}$$

For the lever arm correction for the velocity, we first have to compute the angular rate of the local-level frame with respect to the quasi-inertial frame, expressed in the local-level frame:

$$\boldsymbol{\omega}_{il}^{l} = \begin{pmatrix} \left(\omega_e + \frac{\tilde{v}_{E,IMU,t}}{(N_{ell}+\tilde{h}_{IMU,t})\cos\tilde{\varphi}_{IMU,t}}\right)\cos\tilde{\varphi}_{IMU,t} \\ \frac{-\tilde{v}_{N,IMU,t}}{M_{ell}+\tilde{h}_{IMU,t}} \\ -\left(\omega_e + \frac{\tilde{v}_{E,IMU,t}}{(N_{ell}+\tilde{h}_{IMU,t})\cos\tilde{\varphi}_{IMU,t}}\right)\sin\tilde{\varphi}_{IMU,t} \end{pmatrix}, \tag{7.29}$$

where the variables with subscript IMU,t are parameters of the state vector that was predicted to the current epoch t using IMU observations. Of these variables, $\tilde{v}_E$ and $\tilde{v}_N$ are the velocities in East- and North- direction; $\tilde{\varphi}$ and $\tilde{h}$ are the latitude and height in geographic coordinates. ω_e is the angular rotation rate of the Earth, N_{ell} and M_{ell} are the radii of curvature in the prime vertical plane and in the meridian (Equations 7.4 and 7.3).

In the next step, the angular rates $\boldsymbol{\omega}_{lb}^{b}$ from the body to the local-level frame, expressed in the body frame, are computed from

$$\boldsymbol{\omega}_{lb}^{b} = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \boldsymbol{\omega}_{ib}^{b} - \mathbf{R}_{l}^{b}\boldsymbol{\omega}_{il}^{l}, \tag{7.30}$$

where $\boldsymbol{\omega}_{ib}^{b}$ are the angular rates measured by the IMU, and $\mathbf{R}_{l}^{b}$ is the rotation matrix from the local-level to the body frame. The lever arm correction for the velocity can now be computed from

$$\mathbf{l}_{v}^{l} = \begin{pmatrix} l_{v,N} \\ l_{v,E} \\ l_{v,D} \end{pmatrix} = \mathbf{R}_{b}^{l} \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \mathbf{l}_{GNSS}^{b}. \tag{7.31}$$

With the lever arm corrections defined, we can now compute the reduced observations of the first local filter. This is done by subtracting the GNSS observations at epoch t

from the predicted position, velocity, and heading as follows:

$$\delta \mathbf{z}_{GNSS,t} = \begin{pmatrix} (M_{ell} + \tilde{h}_{IMU,t})\tilde{\varphi}_{IMU,t} - \Big((M_{ell} + \tilde{h}_{IMU,t})\varphi_{GNSS,t} - l_N\Big) \\ \Big((N_{ell} + \tilde{h}_{IMU,t})\cos\tilde{\varphi}_{IMU,t}\Big)\,\tilde{\lambda}_{IMU,t} - \Big(\Big((N_{ell} + \tilde{h}_{IMU,t})\cos\tilde{\varphi}_{IMU,t}\Big)\lambda_{GNSS,t} - l_E\Big) \\ -\tilde{h}_{IMU,t} - (-h_{GNSS,t} - l_D) \\ \tilde{v}_{N,IMU,t} - (v_{N,GNSS,t} - l_{v,N}) \\ \tilde{v}_{E,IMU,t} - (v_{E,GNSS,t} - l_{v,E}) \\ \tilde{v}_{D,IMU,t} - (v_{D,GNSS,t} - l_{v,D}) \\ \tilde{\psi}_{IMU,t} - \psi_{GNSS,t} \end{pmatrix}, \tag{7.32}$$

where the subscript IMU, t indicates that the state was propagated to the current epoch t using the IMU observations in the time update. The subscript $GNSS, t$ denotes a GNSS observation at epoch t. The variables φ, λ, h are the geographic coordinates, v_N, v_E, v_D are the velocity-components in a North-East-Down-frame, and ψ is the heading. The variables l_N, l_E, l_D are components of the lever arm from the IMU to the GNSS antenna in the local-level frame (Equation 7.28); $l_{v,N}, l_{v,E}, l_{v,D}$ are the components of the lever arm correction for the velocity (Equation 7.31). M_{ell} and N_{ell} are (Equations 7.3 and 7.4) are the radii of curvature in the meridian and in the prime vertical plane.

The design matrix $\mathbf{H}_{GNSS,t}$ has the following form:

$$\mathbf{H}_{GNSS,t} = \begin{pmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{A}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{B}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{D}_{3\times3} \\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} & \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} \end{pmatrix}, \tag{7.33}$$

where $\mathbf{I}$ is the identity matrix, and $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{D}$ are skew-symmetric axiator matrices for the lever arm correction. The skew-symmetric axiator matrix of a vector $\mathbf{u} = \begin{pmatrix} u_1 & u_2 & u_3 \end{pmatrix}^T$ is defined as

$$\begin{pmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{pmatrix}. \tag{7.34}$$

The lever arm $\mathbf{l}^b_{GNSS}$ is known in the body frame and can be rotated to the local-level frame with the rotation matrix $\mathbf{R}^l_b$ (Equation 7.28). The axiator matrix $\mathbf{A}$ in Equation

7.33 is the axiator matrix with respect to the lever arm in the local-level frame:

$$\mathbf{A} = \begin{pmatrix} 0 & -l_D & l_E \\ l_D & 0 & -l_N \\ -l_E & l_N & 0 \end{pmatrix}. \tag{7.35}$$

The axiator matrix $\mathbf{B}$ is the axiator matrix of

$$\mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \mathbf{R}_b^l \mathbf{\Omega}_{lb}^b \mathbf{l}_{GNSS}^b, \tag{7.36}$$

where $\mathbf{\Omega}_{lb}^b$ is the angular rate matrix from the body frame to the local-level frame, expressed in the body frame. $\mathbf{D}$ is the matrix

$$\mathbf{D} = \mathbf{A}\mathbf{R}_s^b, \tag{7.37}$$

where the rotation matrix $\mathbf{R}_s^b$ describes the rotation from the sensor frame of the IMU to the body frame (Wendel, 2011).

The observation noise $\mathbf{R}_{GNSS,t}$ is modelled as follows

$$\mathbf{R}_{GNSS,t} = \begin{pmatrix} \sigma_N^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_E^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_D^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{vN}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{vE}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{vD}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_\psi^2 \end{pmatrix}, \tag{7.38}$$

where $\sigma_N^2, \sigma_E^2, \sigma_D^2$ are the variances of the North-, East-, and Down- components of the estimated GNSS position; $\sigma_{vN}^2, \sigma_{vE}^2, \sigma_{vD}^2$ are the variances of the GNSS velocity (in North, East, Down), and σ_ψ^2 is the variance of the GNSS heading.

To compute the measurement update, the Kalman weight matrix is computed according to Equation 4.10. The error state vector is then updated using Equation 4.24, and the corresponding covariance matrix is updated with Equation 4.12.

7.1.2 Local Filter Fusing Observations from the Stereo Camera, Encoders, and the 3D Point Cloud

The second local filter fuses observations of the stereo camera, a 3D point cloud of the environment, and measurements from the rotary encoders of the machine to estimate the position and the attitude. The error state vector of this filter is defined as follows:

$$\delta\mathbf{x}_t = \begin{pmatrix} \delta\mathbf{p} & \delta\mathbf{e} \end{pmatrix}^T, \tag{7.39}$$

where $\delta\mathbf{p}$ are the position errors in a local-level (North-East-Down)-frame, and $\delta\mathbf{e}$ are the attitude errors.

The first three elements of the state vector for the position in geographic coordinates (φ, λ, h) are updated in the same manner as in Equation 7.2. The rest of the state vector is updated according to Equation 4.25.

In the following, the time update with rotary encoders and the measurement update with the stereo camera and the 3D map of the composting site will be explained.

Time Update with Odometry from Rotary Encoders

The prototype of the electrically driven compost turner used in this thesis is equipped with magnetic rotary encoders at the motors of each track. To obtain the track speed in $[m/s]$ from the revolutions measured at the motor, we have to consider that a) the motor drives the tracks with a gear ratio of 79.5:1; and b) the diameter of the track wheel has to be known precisely. We can then compute the track speed v_{track} in $[m/s]$ from

$$v_{track} = \frac{\omega}{79.5} \cdot d \cdot \pi, \tag{7.40}$$

where ω is the radial velocity in $[Hz]$, and d is the diameter of the track wheel in $[m]$. The track speed can be obtained for the left and the right tracks.

To compute how the state of the compost turner changes from the track speeds, a suitable odometry model is needed. As mentioned in Subsection 3.3.2, tracked vehicles use a concept of differential drive known as skid steering. When both tracks are steered at the same speed, the vehicle follows a straight line; when the tracks are steered at different speeds, the vehicle drives a curved path.

When tracked vehicles drive curved paths, they are affected by slip. The slip ratios (Equations 3.40 and 3.41) describe the relationship between the speeds measured by the encoders and the true ground speeds of the tracks.

The equations that describe how the velocity in North ($\dot{N}$) and East ($\dot{E}$) can be computed from the measured velocities at the right (v_r) and left (v_l) tracks are as follows (Endo et al., 2007):

$$\dot{N} = \frac{v_r\,(1 - a_r) + v_l\,(1 - a_l)}{2} \cos\psi, \tag{7.41}$$

$$\dot{E} = \frac{v_r\,(1 - a_r) + v_l\,(1 - a_l)}{2} \sin\psi, \tag{7.42}$$

where a_r and a_l are the slip ratios for the right and left tracks. The heading rate $\dot{\psi}$ can be computed from

$$\dot{\psi} = \frac{v_r\,(1 - a_r) - v_l\,(1 - a_l)}{2d}, \tag{7.43}$$

where $2d$ is the track width. Note that the equations above account for longitudinal slip, while lateral slip is neglected. Lateral slip can be neglected because it is almost zero when the velocity of the vehicle is low and the lateral friction force is high (Nagatani et al., 2007).

When a yaw rate $\widetilde{\dot{\psi}}$ is measured (e.g. by an IMU or a stereo camera through stereo VO), Equation 7.43 and an additional constraint can be used to compute the slip ratios.

Nagatani et al. (2007) propose a method using the following condition for the slip ratios:

$$a_r = -\mathrm{sgn}(v_r \cdot v_l) a_l, \tag{7.44}$$

where sgn is the signum function. Nagatani et al. (2007) derived the condition in Equation 7.44 intuitively from the following two considerations:

- When the left and right tracks rotate in opposite directions, both tracks generate a traction force and therefore cause a positive slip ratio.
- When the left and right tracks rotate in the same direction, then the faster track causes a traction force and a positive slip ratio, while the slower track causes a braking force and a negative slip ratio.

By inserting the condition in Equation 7.44 into Equation 7.43, we can solve the right and left slip ratios as follows:

$$a_r = \frac{v_r - v_l - 2d\widetilde{\dot{\psi}}}{v_r + \operatorname{sgn}(v_r \cdot v_l) v_l}, \tag{7.45}$$

$$a_l = \frac{v_l - v_r + 2d\widetilde{\dot{\psi}}}{v_r + \operatorname{sgn}(v_r \cdot v_l)\, v_l}. \tag{7.46}$$

To compute the time update for the error-state EKF using the rotary encoders and the odometry model that accounts for slip, we need to derive the equations for the system model. For error-state filters, where the small error state vector rather than the full state vector is estimated, a linear system model can be applied in the following form (Groves, 2013):

$$\delta\dot{\mathbf{x}}(t) = \mathbf{F}(t)\delta\mathbf{x}(t) + \mathbf{G}(t)\mathbf{w}_s(t), \tag{7.47}$$

where $\mathbf{F}$ is the system matrix, $\mathbf{G}$ is the system noise distribution matrix, and $\mathbf{w}_s$ is the system noise vector. The system matrix $\mathbf{F}$

$$\mathbf{F}_{t-1} = \left.\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{t-1}} \tag{7.48}$$

contains the derivatives of the system model which describes the dynamics of the vehicle. The equations $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ can be obtained from the odometry model that takes slip into account (Equations 7.41 to 7.43).

The system matrix $\mathbf{F}_{t-1}$ therefore has the following form:

$$\mathbf{F}_{t-1} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \frac{\partial \dot{N}}{\partial \psi} \\ 0 & 0 & 0 & 0 & 0 & \frac{\partial \dot{E}}{\partial \psi} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{7.49}$$

where all derivatives except for $\frac{\partial \dot{N}}{\partial \psi}$ and $\frac{\partial \dot{E}}{\partial \psi}$ are zero. The non-zero derivatives are

$$\frac{\partial \dot{N}}{\partial \psi} = \frac{v_r\,(1 - a_r) + v_l\,(1 - a_l)}{2} \cdot -\sin(\psi) \tag{7.50}$$

and

$$\frac{\partial \dot{E}}{\partial \psi} = \frac{v_r\,(1 - a_r) + v_l\,(1 - a_l)}{2} \cdot \cos(\psi). \tag{7.51}$$

By inserting Equations 7.50 and 7.51 into Equation 7.49, the system matrix $\mathbf{F}_{t-1}$ is obtained.

The transition matrix $\mathbf{\Phi}$ needed in the time update of the Kalman filter can be computed from a power-series expansion of the matrix $\mathbf{F}_{t-1}$:

$$\mathbf{\Phi}_{\mathrm{t-1}} = \mathbf{I} + \mathbf{F}_{t-1}\tau + \frac{\mathbf{F}_{t-1}^2}{2}\tau^2 + \frac{\mathbf{F}_{t-1}^3}{6}\tau^3, \tag{7.52}$$

where τ is the propagation interval from epoch $t-1$ to epoch t.

To model the system noise, the variances $\sigma_{v_r}^2$ and $\sigma_{v_l}^2$ of the measured track speeds are assumed to be uncorrelated:

$$\mathbf{C}_{vtrack} = \begin{pmatrix} \sigma_{v_r}^2 & 0 \\ 0 & \sigma_{v_l}^2 \end{pmatrix}. \tag{7.53}$$

The system noise matrix $\mathbf{Q}_{t-1}$ is obtained from a variance propagation with

$$\mathbf{Q}_{t-1} = \mathbf{J} \cdot \mathbf{C}_{vtrack} \cdot \mathbf{J}^T, \tag{7.54}$$

where $\mathbf{J}$ is defined as

$$\mathbf{J} = \begin{pmatrix} \frac{\partial N}{\partial v_r} & \frac{\partial N}{\partial v_l} \\ \frac{\partial E}{\partial v_r} & \frac{\partial E}{\partial v_l} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{\partial \psi}{\partial v_r} & \frac{\partial \psi}{\partial v_l} \end{pmatrix}. \tag{7.55}$$

The derivatives that are inserted into Equation 7.55 are the following:

$$\frac{\partial N}{\partial v_r} = \cos\psi \cdot \frac{1 - a_r}{2}, \tag{7.56}$$

$$\frac{\partial N}{\partial v_l} = \cos\psi \cdot \frac{1 - a_l}{2}, \tag{7.57}$$

$$\frac{\partial E}{\partial v_r} = \sin\psi \cdot \frac{1 - a_r}{2}, \tag{7.58}$$

$$\frac{\partial E}{\partial v_l} = \sin\psi \cdot \frac{1 - a_l}{2}, \tag{7.59}$$

$$\frac{\partial \psi}{\partial v_r} = \frac{1 - a_r}{2d}, \tag{7.60}$$

$$\frac{\partial \psi}{\partial v_l} = \frac{1 - a_l}{2d}. \tag{7.61}$$

Since the differential equations of the odometry model (Equations 7.41 tp 7.43) only consider horizontal position changes and a heading change, any change in height, roll, and pitch is neglected by the dynamic model. To model these uncertainties, three additional variances (σ_h^2 for the height, σ_ϕ^2 for roll, σ_θ^2 for pitch) are added to the system noise matrix:

$$\bar{\mathbf{Q}}_{t-1} = \mathbf{Q}_{t-1} + \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_h^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_\phi^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_\theta^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{7.62}$$

The matrices $\mathbf{\Phi}_{t-1}$ (Equation 7.52) and $\bar{\mathbf{Q}}_{t-1}$ (Equation 7.62) are then used to compute the time update of the covariance matrix (Equation 4.22). The time update for the state vector (Equation 4.21) is computed through numerical integration of Equations 7.41 to 7.43, i.e. they are multiplied by the propagation interval τ and the position and heading changes obtained are added to the state vector of the previous epoch.

Extending the Odometry Model by Adding a Scale Factor

Note that one of the assumptions for our odometry model was that we have to know the diameter of the track wheels precisely. However, the diameter might vary due to changes in load, speed, pressure or temperature (Groves, 2013). When a position fixing sensor (e.g. GNSS) is integrated with odometry, a scale factor error can be estimated.

To estimate the scale factor, we can extend the state vector as follows:

$$\mathbf{x}_t = \begin{pmatrix} \varphi & \lambda & h & \theta & \phi & \psi & b_o \end{pmatrix}^T, \tag{7.63}$$

where b_o is an odometry bias or scale factor error. The error state vector is also extended by one element (when compared to Equation 7.39):

$$\delta\mathbf{x}_t = \begin{pmatrix} \delta N & \delta E & \delta D & \delta\theta & \delta\phi & \delta\psi & \delta b_o \end{pmatrix}^T, \tag{7.64}$$

where δb_o is the change in scale factor. The scale factor error affects the equations for the North and East velocities in the following form:

$$\dot{N} = (1 - b_o) \cdot \frac{v_r\,(1 - a_r) + v_l\,(1 - a_l)}{2} \cos(\psi)\,, \tag{7.65}$$

$$\dot{E} = (1 - b_o) \cdot \frac{v_r\,(1 - a_r) + v_l\,(1 - a_l)}{2} \sin(\psi)\,. \tag{7.66}$$

Since we now have a seventh state in the state vector, the system matrix $\mathbf{F}_{t-1}$ has to be extended from 6×6 (Equation 7.49) to 7×7:

$$\mathbf{F}_{t-1} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \frac{\partial \dot{N}}{\partial \psi} & \frac{\partial \dot{N}}{\partial b_o} \\ 0 & 0 & 0 & 0 & 0 & \frac{\partial \dot{E}}{\partial \psi} & \frac{\partial \dot{E}}{\partial b_o} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{7.67}$$

The derivatives needed for the above matrix are the following:

$$\frac{\partial \dot{N}}{\partial \psi} = (b_o - 1) \cdot \sin(\psi) \cdot \frac{v_r\,(1 - a_r) + v_l\,(1 - a_l)}{2}, \tag{7.68}$$

$$\frac{\partial \dot{E}}{\partial \psi} = (1 - b_o) \cdot \cos(\psi) \cdot \frac{v_r\,(1 - a_r) + v_l\,(1 - a_l)}{2}, \tag{7.69}$$

$$\frac{\partial \dot{N}}{\partial b_o} = -\cos(\psi) \cdot \frac{v_r\,(1 - a_r) + v_l\,(1 - a_l)}{2}, \tag{7.70}$$

$$\frac{\partial \dot{E}}{\partial b_o} = -\sin(\psi) \cdot \frac{v_r\,(1 - a_r) + v_l\,(1 - a_l)}{2}. \tag{7.71}$$

Another matrix that has to be adapted is the system noise matrix $\mathbf{Q}_{t-1}$, which also becomes a 7×7 matrix. The derivatives of the matrix $\mathbf{J}$ (Equation 7.55) needed for the variance propagation have to be extended to include the scale factor. Equations 7.56 to 7.59 become

$$\frac{\partial N}{\partial v_r} = (1 - b_o) \cdot \cos\psi \cdot \frac{1 - a_r}{2}, \tag{7.72}$$

$$\frac{\partial N}{\partial v_l} = (1 - b_o) \cdot \cos\psi \cdot \frac{1 - a_l}{2}, \tag{7.73}$$

$$\frac{\partial E}{\partial v_r} = (1 - b_o) \cdot \sin\psi \cdot \frac{1 - a_r}{2}, \tag{7.74}$$

$$\frac{\partial E}{\partial v_l} = (1 - b_o) \cdot \sin\psi \cdot \frac{1 - a_l}{2}. \tag{7.75}$$

Measurement Update with Normal Distributions Transform

For the route planning module of the automatic steering system (see Figure 6.1), a precise 3D point cloud of the composting site is available. The ZED stereo camera, which was selected as a sensor for the navigation module, can output a point cloud. When the point cloud observed by the stereo camera mounted on the compost turner is registered to the 3D point cloud of the composting site, the pose (position and attitude) of the compost turner can be estimated.

The point cloud observed by the stereo camera is registered to the point cloud of the composting site using Normal Distributions Transform (NDT) (see Subsection 3.4.3). NDT was chosen as a registration algorithm due to its computational efficiency: the mean $\boldsymbol{\mu}$ and covariance $\mathbf{Q}$ for the voxels of the target point cloud, i.e. the point cloud of the composting site, only have to be computed in the beginning.

The estimated pose of NDT is used as an observation in the second local Kalman filter. The reduced observations are computed from

$$\delta\mathbf{z}_{NDT,t} = \begin{pmatrix} (M_{ell} + \tilde{h}_{ODO,t})\tilde{\varphi}_{ODO,t} - \left((M_{ell} + \tilde{h}_{ODO,t})\varphi_{NDT,t} - l_N\right) \\ \left((N_{ell} + \tilde{h}_{ODO,t})\cos\tilde{\varphi}_{ODO,t}\right)\tilde{\lambda}_{ODO,t} - \left(\left((N_{ell} + \tilde{h}_{ODO,t})\cos\tilde{\varphi}_{ODO,t}\right)\lambda_{NDT,t} - l_E\right) \\ -\tilde{h}_{ODO,t} - (-h_{NDT,t} - l_D) \\ \tilde{\theta}_{ODO,t} - \theta_{NDT,t} \\ \tilde{\phi}_{ODO,t} - \phi_{NDT,t} \\ \tilde{\psi}_{ODO,t} - \psi_{NDT,t} \end{pmatrix}, \tag{7.76}$$

where the subscript NDT, t denotes an observation from NDT at epoch t and the subscript ODO, t indicates that the respective parameter of the state vector was propagated to the epoch t using the odometry model for tracked vehicles. φ, λ, h are the geographic coordinates of the vehicle and θ, ϕ, ψ are the attitude parameters as Euler angles. The parameters l_N, l_E, l_D are the components of the lever arm in the local-level frame (comparably obtained as in Equation 7.28). The lever arm points from the center of the vehicle, where the odometry solution is computed, to the stereo camera, where the NDT-solution is computed.

The design matrix $\mathbf{H}_{NDT,t}$ is computed from

$$\mathbf{H}_{NDT,t} = \begin{pmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{pmatrix}, \tag{7.77}$$

where $\mathbf{I}$ is the identity matrix.

The observation noise $\mathbf{R}_{NDT,t}$ is modelled as uncorrelated with the variances of the position and attitude on the main diagonal:

$$\mathbf{R}_{NDT,t} = \begin{pmatrix} \sigma_N^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_E^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_D^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_\theta^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_\phi^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_\psi^2 \end{pmatrix}, \tag{7.78}$$

where $\sigma_N^2, \sigma_E^2, \sigma_D^2$ are the variances of the position in a local-level (North-East-Down)-frame, and $\sigma_\theta^2, \sigma_\phi^2, \sigma_\psi^2$ are the variances of the Euler angles roll, pitch, and yaw.

7.1.3 Snapshot Fusion

For the snapshot fusion, the position and attitude estimated by both local filters are combined in a least squares adjustment to compute the final position and attitude of the compost turner. Since the local Kalman filters do not share information from the same sensors, the state vectors estimated by the local Kalman filters are uncorrelated. Therefore, a recursive least squares adjustment can be used for the snapshot fusion (Hofmann-Wellenhof et al., 2003).

For uncorrelated groups of observations, the observation model can be partitioned (Hofmann-Wellenhof et al., 2003):

$$\begin{pmatrix} \mathbf{z}_0 \\ \mathbf{z}_1 \end{pmatrix} = \begin{pmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \end{pmatrix} \mathbf{x} + \begin{pmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \end{pmatrix}, \tag{7.79}$$

where $\mathbf{z}_0$ and $\mathbf{z}_1$ are the uncorrelated observation vectors with the corresponding noise vectors $\mathbf{v}_0$ and $\mathbf{v}_1$; $\mathbf{H}_0$ and $\mathbf{H}_1$ are the design matrices; $\mathbf{x}$ is the parameter vector to be

estimated. Given that the observation groups are uncorrelated, the covariance matrix of the observations can be written as

$$\begin{pmatrix} \mathbf{R}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_1 \end{pmatrix}. \tag{7.80}$$

The solution for $\mathbf{x}$ using only the first observation group is extimated from (Hofmann-Wellenhof et al., 2003)

$$\mathbf{P}_0 = \left(\mathbf{H}_0^{\mathrm{T}}\mathbf{R}_0^{-1}\mathbf{H}_0\right)^{-1} \tag{7.81}$$

and

$$\hat{\mathbf{x}}_0 = \mathbf{P}_0\mathbf{H}_0^{\mathrm{T}}\mathbf{R}_0^{-1}\mathbf{z}_0. \tag{7.82}$$

The solution is then updated with the second observation group as follows (Hofmann-Wellenhof et al., 2003):

$$\mathbf{K}_1 = \mathbf{P}_0\mathbf{H}_1^{\mathrm{T}}\left(\mathbf{H}_1\mathbf{P}_0\mathbf{H}_1^{\mathrm{T}} + \mathbf{R}_1\right)^{-1}, \tag{7.83}$$

$$\mathbf{P}_1 = \mathbf{P}_0 - \mathbf{K}_1\mathbf{H}_1\mathbf{P}_0, \tag{7.84}$$

$$\hat{\mathbf{x}}_1 = \hat{\mathbf{x}}_0 + \mathbf{K}_1\left(\mathbf{z}_1 - \mathbf{H}_1\hat{\mathbf{x}}_0\right). \tag{7.85}$$

For the snapshot fusion algorithm, which is used to fuse information from both local filters, the observation vectors consist of the estimated positions $(\hat{\varphi}, \hat{\lambda}, \hat{h})$ and attitude parameters $(\hat{\theta}, \hat{\psi}, \hat{\phi})$ of the local Kalman filters.

The first observation vector contains the parameters estimated from the GNSS/INS fusion (first filter) at epoch t:

$$\mathbf{z}_0 = \begin{pmatrix} \hat{\varphi} & \hat{\lambda} & \hat{h} & \hat{\theta} & \hat{\phi} & \hat{\psi} \end{pmatrix}^T_{GNSS/INS,t}. \tag{7.86}$$

The corresponding observation noise is taken from the estimated covariance matrix $\mathbf{P}_{(GNSS/INS),t}$ of the first filter:

$$\mathbf{R}_0 = \mathbf{P}^{(\mathbf{p},\mathbf{e})}_{GNSS/INS,t}. \tag{7.87}$$

Note that the superscript $(\mathbf{p}, \mathbf{e})$ should indicate that not the full 15×15 covariance matrix $\mathbf{P}_{GNSS/INS,t}$ is used, but only the 6×6 subset that refers to the variances and covariances of the position and the attitude.

The state vector that we want to estimate contains the position and attitude parameters

$$\mathbf{x} = \begin{pmatrix} \varphi & \lambda & h & \theta & \phi & \psi \end{pmatrix}^T. \tag{7.88}$$

Therefore, the design matrix $\mathbf{H}_0$ for the first set of observations is a 6×6 identity matrix:

$$\mathbf{H}_0 = \mathbf{I}_{6\times 6}. \tag{7.89}$$

Since $\mathbf{H}_0 = \mathbf{I}$, Equations 7.81 and 7.82 can be simplified to:

$$\mathbf{P}_0 = \mathbf{R}_0, \tag{7.90}$$

$$\hat{\mathbf{x}}_0 = \mathbf{z}_0. \tag{7.91}$$

The second group of observations is the estimated state vector of the second local Kalman filter at epoch t, denoted by NDT/ODO to indicate that NDT and odometry were used in the filter:

$$\mathbf{z}_1 = \begin{pmatrix} \hat{\varphi} & \hat{\lambda} & \hat{h} & \hat{\theta} & \hat{\phi} & \hat{\psi} \end{pmatrix}^T_{NDT/ODO,t}. \tag{7.92}$$

The observation covariance matrix is the estimated 6×6 covariance matrix from the second local Kalman filter:

$$\mathbf{R}_1 = \mathbf{P}_{NDT/ODO,t}. \tag{7.93}$$

In the same way as for the first observation group, the design matrix for the second observation group is the identity matrix:

$$\mathbf{H}_1 = \mathbf{I}_{6\times 6}. \tag{7.94}$$

Equation 7.83 is therefore simplified to:

$$\mathbf{K}_1 = \mathbf{R}_0(\mathbf{R}_0 + \mathbf{R}_1)^{-1}. \tag{7.95}$$

Equation 7.84 becomes

$$\mathbf{P}_1 = \mathbf{R}_0 - \mathbf{K}_1\mathbf{R}_0. \tag{7.96}$$

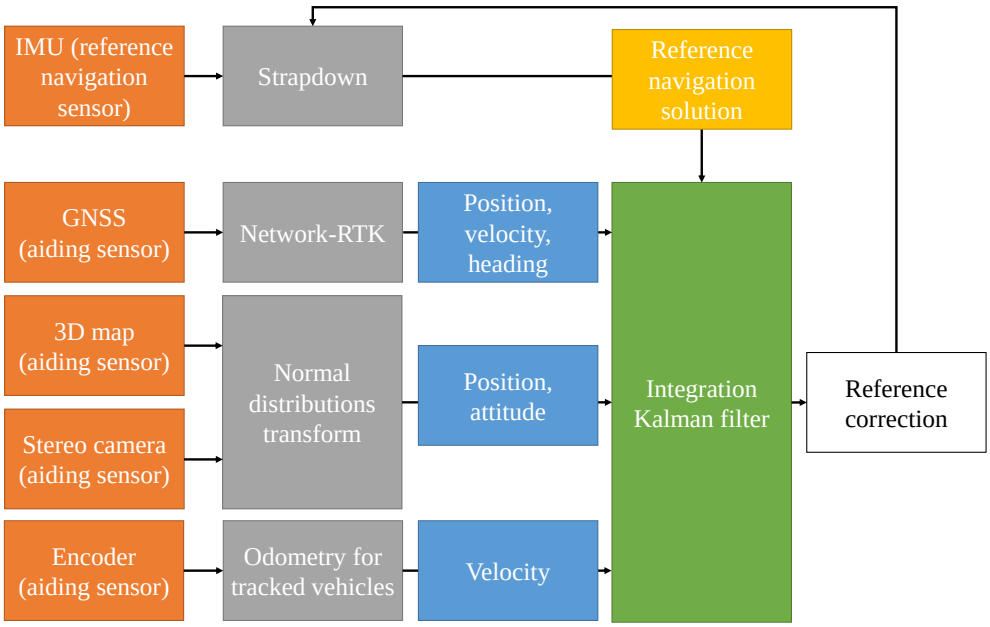


Figure 7.2: Error-state cascaded architecture for fusing observations of an IMU, a GNSS receiver, a 3D map, a stereo camera, and encoders.

Finally, the parameter vector $\mathbf{x}_1$, which contains the position and attitude of the machine, is estimated from a simplification of Equation 7.85 as follows:

$$\hat{\mathbf{x}}_1 = \mathbf{z}_0 - \mathbf{K}_1(\mathbf{z}_1 - \mathbf{z}_0). \tag{7.97}$$

7.2 Cascaded Integration

In a cascaded integration architecture, the observations are pre-processed before they are fused in a common integration Kalman filter. Figure 7.2 gives an overview of the proposed error-state cascaded integration architecture for fusing observations of the IMU, GNSS, the 3D point cloud, the stereo camera, and the rotary encoders. Here, the IMU is used as a reference navigation sensor to propagate the state vector in the time update. GNSS, the 3D map, the stereo camera, and the encoders serve as aiding sensors. The integration Kalman filter uses information from all available aiding sensors to estimate the error state vector.

Since the IMU is used as reference navigation sensor, the error state vector includes the errors for accelerometer and gyroscope biases and has the following form:

$$\delta\mathbf{x}_t = \begin{pmatrix} \delta\mathbf{p} & \delta\mathbf{v} & \delta\mathbf{e} & \delta\mathbf{b_f} & \delta\mathbf{b_g} \end{pmatrix}^T, \tag{7.98}$$

where $\mathbf{p}$ is the position error in a local-level (North-East-Down)-frame, $\delta\mathbf{v}$ is the velocity error in a local-level (North-East-Down)-frame, $\delta\mathbf{e}$ are the attitude errors, and $\delta\mathbf{b_f}$ and $\delta\mathbf{b_g}$ are the errors of the accelerometer and gyroscope biases in three axes.

The state vector contains the position in geographic coordinates (φ, λ, h) as described in Subsection 7.1.1, and is therefore updated with Equation 7.2 in the same manner as for the first local filter of the federated filter.

Time Update with the IMU

The time update with the IMU is performed in the same way as for the first local filter of the federated filter in Subsection 7.1.1.

Measurement Update with GNSS

The equations for the measurement update with GNSS are the same as for the first local Kalman filter of the federated filter (see Subsection 7.1.1).

Measurement Update with Normal Distributions Transform

The point cloud observed by the stereo camera is registered to the available point cloud of the composting site using NDT. The estimated pose resulting from the registration with NDT is used to compute the reduced observations that are used in the measurement update of the filter. This is done in the same way as described in Subsection 7.1.2 for the second local Kalman filter of the federated filter. The only difference is that the reduced observations of Equation 7.76 are now computed by subtracting the pose from NDT from the pose predicted with the IMU observations instead of the odometry observations.

Rotary Encoders as Aiding Sensors

As already mentioned in Subsection 7.1.2, the readings from the rotary encoders can be processed to yield the track speeds of the left and right tracks using Equation 7.40. The track speeds can then be used to compute odometry information of the vehicle.

For tracked vehicles such as compost turners, a suitable skid-steering odometry model is needed which accounts for slip. When the yaw rate $\tilde{\dot{\psi}}$ of the vehicle is measured (e.g. using the stereo camera and stereo VO, or using the IMU), the longitudinal slip ratios a_r and a_l can be computed (Equations 7.45 and 7.46).

With the slip ratios known, the velocities in North (v_N) and East (v_E) can now be computed from the left and right track speeds (v_l, v_r) as follows (also see Equations 7.41 and 7.42):

$$v_{N,ODO,t} = \frac{v_r\,(1-a_r) + v_l\,(1-a_l)}{2} \cdot \cos\psi, \tag{7.99}$$

$$v_{E,ODO,t} = \frac{v_r\,(1-a_r) + v_l\,(1-a_l)}{2} \cdot \sin\psi. \tag{7.100}$$

These pre-processed velocities from odometry are then used to compute the reduced observations $\delta\mathbf{z}_{ODO,t}$ for the error-state EKF:

$$\delta\mathbf{z}_{ODO,t} = \begin{pmatrix} \widetilde{v}_{N,IMU,t} - v_{N,ODO,t} \\ \widetilde{v}_{E,IMU,t} - v_{E,ODO,t} \\ \widetilde{v}_{D,IMU,t} - 0 \end{pmatrix}. \tag{7.101}$$

Here, $\widetilde{v}_{N,IMU,t}, \widetilde{v}_{E,IMU,t}$ and $\widetilde{v}_{D,IMU,t}$ are the predicted North-, East-, and Down-components of the velocity vector predicted with observations from the IMU. Note that since odometry only yields horizontal velocities, a zero is used as a pseudo-observation for the Down-component of the velocity vector in Equation 7.101.

The 2×2 observation covariance matrix $\mathbf{R}^*_{ODO,t}$ for the North and East compontents of the velocity is obtained through variance propagation:

$$\mathbf{R}^*_{ODO,t} = \mathbf{N} \cdot \mathbf{C}_{vtrack} \cdot \mathbf{N}^T, \tag{7.102}$$

where $\mathbf{C}_{vtrack}$ contains the variances for the speed of the left and right tracks (see Equation 7.53), and where the matrix $\mathbf{N}$ contains the following derivatives:

$$\mathbf{N} = \begin{pmatrix} \frac{\partial v_{N,ODO}}{\partial v_r} & \frac{\partial v_{N,ODO}}{\partial v_l} \\ \frac{\partial v_{N,ODO}}{\partial v_r} & \frac{\partial v_{N,ODO}}{\partial v_l} \end{pmatrix} = \begin{pmatrix} \frac{(1-a_r)\cos\psi}{2} & \frac{(1-a_l)\cos\psi}{2} \\ \frac{(1-a_r)\sin\psi}{2} & \frac{(1-a_l)\sin\psi}{2} \end{pmatrix}. \tag{7.103}$$

To obtain the 3×3 observation covariance matrix $\mathbf{R}_{ODO,t}$ needed for the measurement update with the 3×1 reduced observation vector $\delta\mathbf{z}_{ODO,t}$ of Equation 7.101, a variance σ^2_{vD} is added for the pseudo-observation of zero speed in the Down-component. The speed in the Down-component is assumed to be uncorrelated with the speeds in the North- and East-components. Thus, the observation covariance $\mathbf{R}_{ODO,t}$ has the following form:

$$\mathbf{R}_{ODO,t} = \begin{pmatrix} & & 0 \\ & \mathbf{R}^*_{ODO,t} & 0 \\ & & \\ 0 & 0 & \sigma^2_{vD} \end{pmatrix}. \tag{7.104}$$

The design matrix for the measurement update with odometry from rotary encoders has the form

$$\mathbf{H}_{ODO,t} = \begin{pmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{pmatrix}, \tag{7.105}$$

where $\mathbf{I}$ is the identity matrix.

Adding a Scale Factor

As already mentioned, a scale factor for the velocities from odometry can be estimated in the Kalman filter when odometry is integrated with position fixing sensors. This is not only the case when odometry is used in the time update, but it can also be done when odometry is used in the measurement update.

We therefore add the scale factor b_o as a parameter to the state vector and the state vector error δb_o to the error state vector $\delta\mathbf{x}_t$:

$$\delta\mathbf{x}_t = \begin{pmatrix} \delta N & \delta E & \delta D & \delta v_N & \delta v_E & \delta v_D & \dots & \delta b_o \end{pmatrix}^T. \tag{7.106}$$

The matrix for the reduced observations (Equation 7.101) becomes

$$\delta \mathbf{z}_{ODO,t} = \begin{pmatrix} \widetilde{v}_{N,IMU,t} - (1 - b_o) \cdot v_{N,ODO,t} \\ \widetilde{v}_{E,IMU,t} - (1 - b_o) \cdot v_{N,ODO,t} \\ \widetilde{v}_{D,IMU,t} - 0 \end{pmatrix}. \tag{7.107}$$

and the design matrix H (Equation 7.105) becomes

$$\mathbf{H}_{ODO,t} = \begin{pmatrix} 0 & 0 & 0 & (1 - b_o) & 0 & 0 & \dots & v_N \\ 0 & 0 & 0 & 0 & (1 - b_o) & 0 & \dots & v_E \\ 0 & 0 & 0 & 0 & 0 & 1 & \dots & 0 \end{pmatrix}. \tag{7.108}$$

7.3 Software Implementation

The multi-sensor positioning module for the automatic steering system was implemented in C++ using the Robot Operating System (ROS) framework. ROS is an open-source framework which provides software libraries and tools for robotics. The software for the positioning module was written by Eva Reitbauer and Christoph Schmied.

An overview of the program flow is given in Figure 7.3. The navigation sensors are shown in red and their corresponding nodes in blue. The subscriber nodes are depicted in yellow, the publisher-subscriber nodes in grey, and the ROS master in turquoise.

A major advantage of using ROS is that the sensor hardware communication is decoupled from the main program flow. This is realized through a publisher-subscriber architecture.

The GNSS receiver is connected to the navigation module via a Local Area Network (LAN) cable, the IMU and the stereo camera are connected via Universal Serial Bus (USB), and the encoders via Controller Area Network (CAN). Each sensor node publishes a ROS message.

The NDT node subscribes to the ROS message published by the VO node, i.e. the point cloud observed by the stereo camera, computes the camera pose and publishes it. The EKF node is the node of the filter (either the federated or the cascaded filter). It

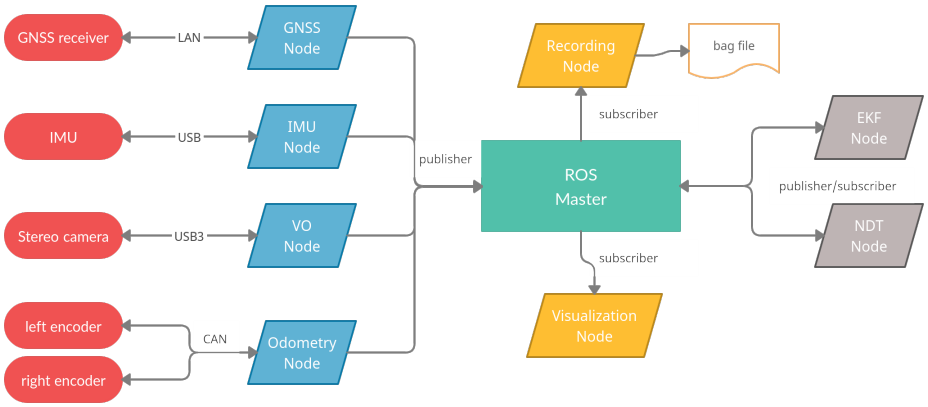


Figure 7.3: Program flow of the multi-sensor positioning module in ROS.

subscribes to the GNSS node, the IMU node, the Odometry node and the NDT node. The EKF node computes the position and attitude of the compost turner, as well as the steering angle, and publishes them. The EKF node and the NDT node run in parallel on two different threads.

Note that multiple nodes can subscribe to the same message. This allows simultaneously running the filter in real time and recording the sensor messages with the recording node. Moreover, the pose can be visualized in real time through the visualization node.

Part III

Practical Investigations and Results

Chapter 8

Data Collection

This part of the thesis deals with the practical investigations which were carried out at the composting site to validate the filters developed. This chapter focuses on the data collection of two test campaigns. The data collected will later be analysed in Chapter 9 and used to evaluate the accuracy achievable with the sensors selected and the filters developed. In Section 8.1, the first tests are described which were carried out with the prototype of the electrically driven self-driving compost turner. Section 8.2 deals with the final tests and the demonstration of the automatic steering system.

8.1 First Tests with the Prototype

In March 2021, the first tests were carried out with the prototype vehicle at a composting site in Riedlingsdorf, Austria. The aim of the tests was to collect data from all navigation sensors and to store them in ROS bag files. A ROS bag file is a file format used to store ROS messages. Such bag files can later be replayed to simulate a real-time environment. By replaying the same bag files, we can test the two different filter architectures, namely federated and cascaded integration, with the same data.

Another aim of the tests was to collect a reference trajectory. By comparing the estimated trajectories of the filters to a reference, it is possible to evaluate the accuracy achievable. In the following, the data collection of the first tests will be explained.

8.1.1 3D Point Cloud of the Composting Site

The automatic steering system (see Section 6.1) uses a 3D point cloud as input data to compute the nominal routes. This point cloud is also used in the positioning algorithm, where the point cloud recorded by the stereo camera is registered to the 3D point cloud of the composting site using Normal Distributions Transform (NDT) to estimate the pose of the machine.

The 3D point cloud of the composting site was collected on 17 March 2021. The point cloud was obtained from UAV photogrammetry. 16 ground control points were measured with a Leica GS18 T GNSS RTK rover, which received correction data from the network-RTK service EPOSA. Figure 8.1 gives an overview of how the ground control points were distributed at the composting site. The ground control points were placed around and in between the two windrows which were later used in the tests with the prototype vehicle. These two windrows were not turned or moved in the time between the 3D point cloud generation and the tests with the prototype vehicle.

The UAV flight was carried out by Dr. Gernot Seier of the Institute of Geography and Regional Science at the University of Graz. The UAV, a DJI Phantom 4, was equipped with an FC330 camera. Figure 8.2 gives an overview of the distribution of the images that were captured with the camera of the UAV. In total, two flights were carried out. The first flight (48 photos) was conducted at a height of 19.5 metres above ground with a Ground Sampling Distance (GSD) of 0.82 cm; the second flight (45 photos) was conducted at 30.5 metres above ground with a GSD of 1.29 cm.

After the measurement data were collected, the ground control points were brought to a local-level frame. The ground control point with the number 200 was defined as the origin of the local-level frame. The images were then processed in Agisoft Metashape by Prof. Viktor Kaufmann of the Institute of Geodesy at Graz University of Technology. Prof. Kaufmann provided the digital elevation model in the form of a point cloud and an orthophoto.

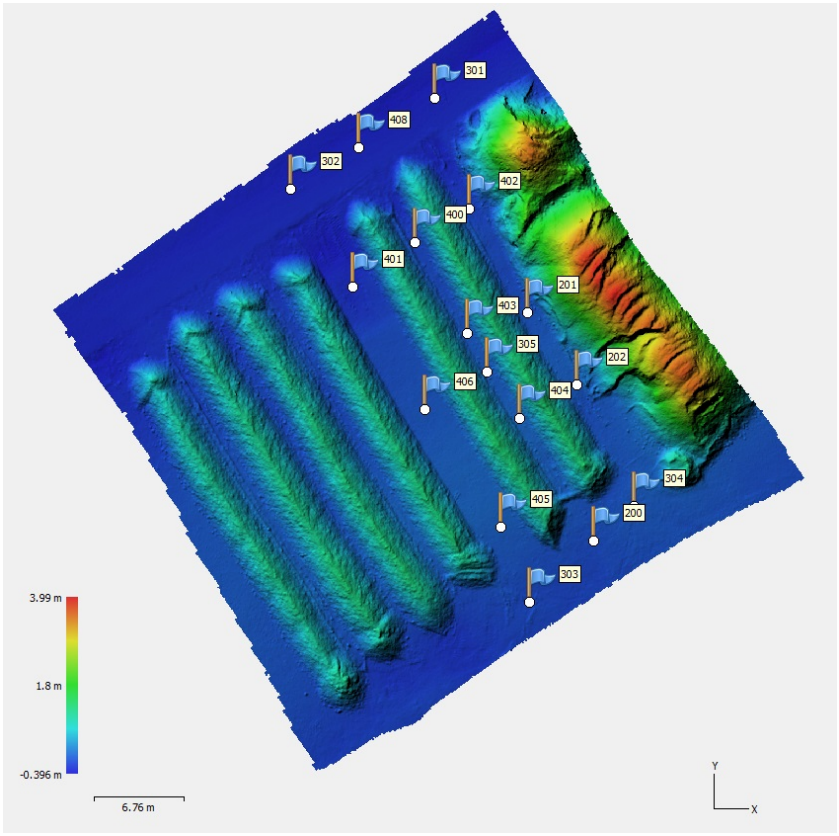


Figure 8.1: Overview of the ground control points.

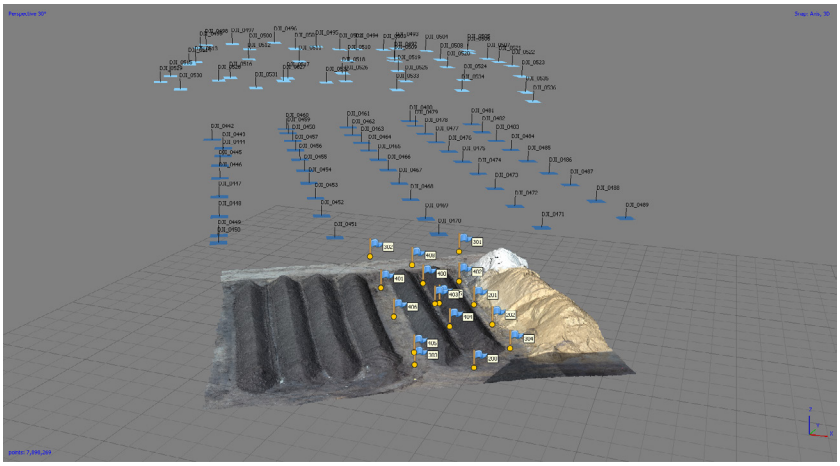


Figure 8.2: Overview of the images that were captured to obtain the 3D point cloud.

8.1.2 Tests with the Prototype Vehicle

The first tests with the prototype were carried out on 24 March 2021 at the composting site of Sonnenerde GmbH in Riedlingsdorf, Austria.

Sensor Mounting

The prototype vehicle, an eWender 35eco produced by the Austrian environmental engineering company Pusch & Schinnerl GmbH, was equipped with an aluminium profile, as shown in Figure 8.3. On the aluminium profile, we mounted the two Trimble GA810 GNSS antennas, the XSens MTi-G-710 IMU, and the ZED stereo camera. The sensor mounting is shown in Figure 8.4. The two Trimble GA810 GNSS antennas were connected to the Trimble BD992 GNSS receiver. The GNSS receiver was connected to the computing unit of the navigation module with a LAN cable. The computing unit is a Jetson Nano by NVIDIA. To protect it from dust and dirt, it was put into a box (see Figure 8.5) on the roof of the vehicle. The stereo camera was connected to the Jetson Nano via a USB3 cable, and the IMU was connected via a USB cable. Moreover, the Jetson Nano was connected to the CAN bus of the vehicle, from which it had access to the readings of the two rotary encoders (two 2 Channel HallD SDN6.FK10.E05R sensors).

Reference Tracking

To generate a reference trajectory, two Leica 360°-prisms were mounted on the aluminium profile below the GNSS antennas (Figure 8.4). Two robotic total stations, a Leica TCRA1201 and a Leica Nova MS60, were used to track the prisms. The accuracies of the total stations are stated in Table 6.2.

The two total stations were set up on tripods. Additionally, three tripods with a forced centering were placed at the composting site. The positions of these tripods were measured with the Trimble BD992 receiver, which had access to the network-RTK service EPOSA. The positions of the tripods were then brought to the same local-level frame as the 3D point cloud, using the ground control point 200 as the origin. Prisms were then

Figure 8.3: Prototype during the tests in March 2021.

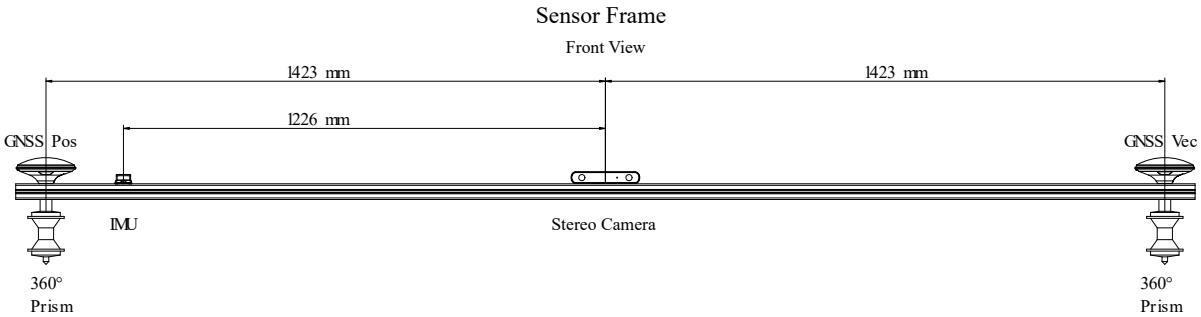


Figure 8.4: Sensor mounting on the aluminium profile.

Figure 8.5: Picture of the box protecting the Jetson Nano computing unit and the GNSS receiver from dust and dirt.

set up on the three tripods and measured with the Leica Nova MS60. A resection was performed to obtain the coordinates of the Leica Nova MS60 total station. The three prisms on the tripods were then measured anew with the Leica Nova MS60, to obtain highly accurate local coordinates for them. Then the newly measured local coordinates of the prisms were used to perform a resection with the second total station, the Leica TCRA1201.

For time synchronisation between the two total stations, the same 360°-prism was lifted and set to the ground again while it was being tracked by both total stations. After that, the prism was mounted on the compost turner again. During the tests, one total station tracked the left and the other the right prism on the machine. The reference trajectory was later processed in the same way as described in Subsection 6.4.3.

Trajectories and Data Logging

Figure 8.6 gives an overview of the area where the field tests were conducted. During the tests, the compost turner was steered manually (with a remote control) through the two

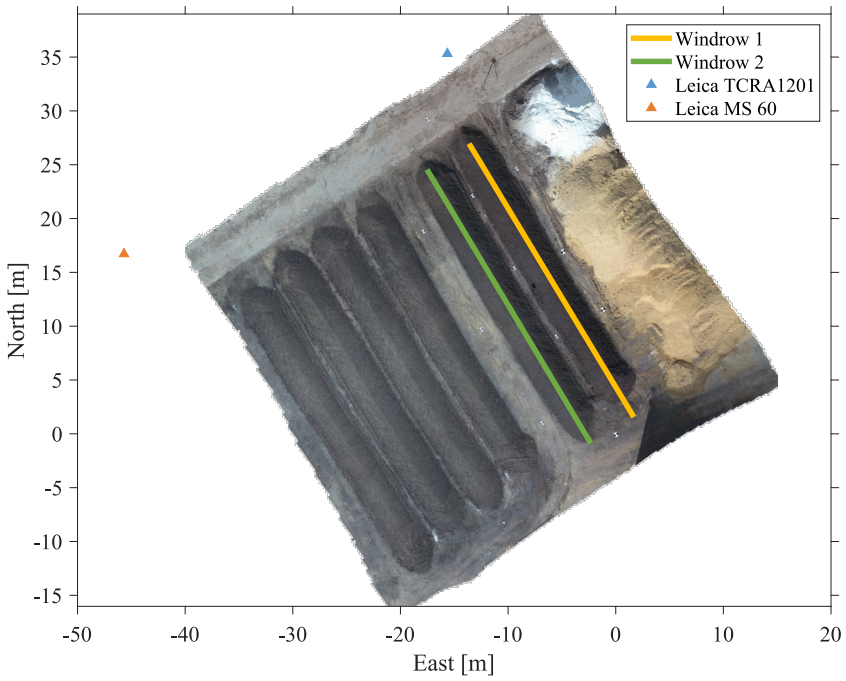


Figure 8.6: Overview of the trajectories of the first tests in March 2021.

windrows at the composting site which are marked yellow and green. Both windrows are approximately 30 metres long. The positions of the total stations that were used to generate the reference trajectory are shown as triangles.

At the time of the first tests, the spiked drum of the prototype compost turner was not working. Therefore, the spiked drum had to be unhinged and the compost turner drove through the two windrows without turning them.

During the tests, the sensor messages published by each sensor node in ROS were recorded into bag files. An overview of the messages recorded is given in Table 8.1.

8.2 Final Tests and Demonstration of the Automatic Steering System

In September 2021, final tests were carried out at the composting site of Sonnenerde GmbH in Riedlingsdorf, Austria. One aim of the final tests was to demonstrate the

Table 8.1: Overview of the sensor messages recorded during the tests in March 2021.

Sensor	Type	Observation	Data Rate [Hz]
GNSS dual-antenna receiver	Trimble BD992, two GA810 antennas	position, velocity, heading	10
MEMS IMU	XSens MTi-G-710	accelerations, angular rates	100
Stereo camera	ZED	point cloud	10
Rotary encoders	2 Channel HallD SDN6.FK10.E05R	left track speed, right track speed	60

automatic steering system in real-time. Another aim was to record data from the navigation sensors and the real-time filter to bag files while the compost turner was driving through the windrows and turning them with its spiked drum. By comparing the filter results to a reference trajectory, the accuracy achievable was evaluated (see Chapter 9). In the following, the data collection of the final tests is outlined.

8.2.1 3D Point Cloud of the Composting Site

On 13 September 2021, a 3D point cloud of the windrows at the composting site was measured with terrestrial laser scanning.

First, two tripods with a forced centering were set up at the composting site. The positions of these two tripods were measured with GNSS-RTK using the Trimble BD992, a GA810 antenna, and access to the network-RTK service EPOSA. Then, a Leica TS15 total station was set up on the first tripod and a prism on the second tripod. The coordinates of the tripods were brought to a local-level frame, where the coordinates of the tripod carrying the total station were used as the origin of the local-level frame. The orientation of the total station was then set in the direction of the second tripod.

To allow for accurate registration of the individual scans, eight targets were set up on tripods at the composting site. The positions of the targets were measured with the Leica TS15 total station.

Figure 8.7: Panorama picture taken by the Leica RTC360.

Then the terrestrial laser scanner, a Leica RTC360, was set up at 16 different positions on the composting site. At each scan position, a scan with a 360° horizontal and 300° vertical field of view as well as a panorama picture was captured (Figure 8.7).

According to the manufacturer's datasheet, the Leica RTC360 laser scanner has an angular accuracy of 18" and a range accuracy of 1.0 mm + 10 ppm. This leads to a 3D accuracy of 1.9 mm for points at distance of 10 m from the scan position, and a 5.3 mm 3D accuracy for points at a distance of 40 m from the scan position (Leica Geosystems AG, 2018).

8.2.2 Tests with the Prototype Vehicle

The demonstration of the automatic steering system took place on 16 September 2021.

Sensor Mounting

The navigation sensors were mounted on the aluminium profile on the compost turner in the same manner as for the first tests (Section 8.1). Two pictures of the compost turner with the navigation sensors mounted are shown in Figure 8.8. The lever arms between the sensors are shown in Figure 8.4.

(a) **(b)**

Figure 8.8: eWender e35eco equipped with the navigation sensors for the automatic steering system (a) seen from the front, (b) seen from the side.

Reference Tracking

In the same manner as for the first tests, a reference trajectory was generated by tracking two 360°-prisms that were mounted on the compost turner with two robotic total stations. The prisms were tracked by a Leica TS15 and a Leica TCRA1201 total station. Figure 8.9 shows two pictures of the total stations during the final tests.

Trajectories and Data Logging

For the final tests and the demonstration of the automatic steering system, two windrows were turned automatically by the compost turner. Figure 8.10 shows these two windrows. The first windrow that was turned is coloured in blue, the second one is coloured in green.

As for the first tests, data from the navigation sensors were recorded into ROS bag files. Moreover, additional parameters such as the battery status, revolutions of the spiked drum, the steering angle, and other parameters on the CAN bus were logged. This led to problems with the bandwidth. Due to these bandwidth problems, the data were logged at a lower data rate than for the first tests. Table 8.2 shows the navigation sensor messages that were recorded with their respective data rates.

(a) (b)

Figure 8.9: Tracking the reference trajectory with (a) a Leica TS15, and (b) a Leica TCRA1201.

Figure 8.10: The two windrows that were turned during the demonstration of the automatic steering system are shown in blue and green.

Table 8.2: Overview of the sensor messages recorded during the tests in September 2021.

Sensor	Type	Observation	Data Rate [Hz]
GNSS dual-antenna receiver	Trimble BD992, two GA810 antennas	position, velocity, heading	6
MEMS IMU	XSens MTi-G-710	accelerations, angular rates	60
Stereo camera	ZED	point cloud	2.5
Rotary encoders	2 Channel HallD SDN6.FK10.E05R	left track speed, right track speed	37

Chapter 9

Results

In this chapter, the results of the tests are presented and analysed. Section 9.1 discusses the results of the first tests and Section 9.2 covers the results of the final tests and the demonstration of the automatic steering system.

9.1 Results of the First Tests: Evaluation of the Filter Designs

In this section, the two different filter architectures that were presented in Chapter 7 are evaluated using the data collected during the first tests in March 2021. Part of what is presented in this section has already been published in Reitbauer and Schmied (2021a) and Reitbauer and Schmied (2021b).

To analyze the accuracy achievable, the positions and headings estimated by the filters are compared to the reference trajectory. The reference trajectory is estimated to have an accuracy of $\leq$1 cm. The 2D and 3D position errors (RMS_{2D} and RMS_{3D}) at epoch t are computed from

$$RMS_{2D}(t) = \sqrt{(n_r(t) - n_f(t))^2 + (e_r(t) - e_f(t))^2}, \tag{9.1}$$

$$RMS_{3D}(t) = \sqrt{(n_r(t) - n_f(t))^2 + (e_r(t) - e_f(t))^2 + (h_r(t) - h_f(t))^2}, \tag{9.2}$$

where n, e, and h are the North-, East-, and height-coordinates. The subscript r stands for the reference trajectory and the subscript f stands for the filtered solution.

The heading error $\Delta\psi$ at epoch t is computed as follows:

$$\Delta\psi(t) = \psi_r(t) - \psi_f(t), \tag{9.3}$$

where ψ_r is the reference heading and ψ_f the heading estimated by the filter.

Note that, for the first tests, the compost turner drove through the windrows without turning them, as the spiked drum was not working at the time of the tests. Furthermore, it was steered at different speeds, as illustrated in Figure 9.1. The left side (a) of Figure 9.1 shows the left and right track speeds for the first windrow, and the right side (b) shows the left and right track speeds for the second windrow. For the first windrow, the speed was approximately 0.2 m/s, while for the second windrow it was approximately 0.1 m/s.

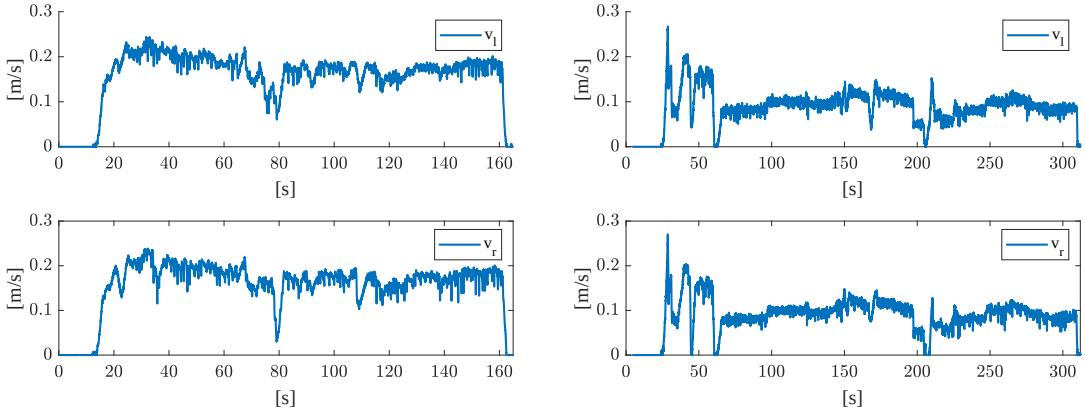


Figure 9.1: Left (v_l) and right (v_r) track speeds for (a) the first windrow, and (b) the second windrow. The track speeds were calculated according to Formula (7.40), where a diameter $d = 38.5$ [cm] was used.

9.1.1 Analysis of the Results of the Federated Filter

In this section, the results of the federated filter are analysed. First, the accuracy achievable with the two local filters is evaluated. Then, the results of the snapshot fusion are shown.

Figure 9.2 shows the accuracy achievable with the first local filter fusing observations of GNSS and the IMU. The left side (a) shows the results of the first windrow and the right side (b) shows the results of the second windrow. The upper part of the figure shows the 2D and 3D position errors (RMS_{2D} and RMS_{3D}) in cm and the lower part shows the heading deviation from the reference $\Delta\psi$ in degrees. For both windrows, a very high accuracy is achieved. The 3D position error is less than 3 cm and the heading error is less than 1°.

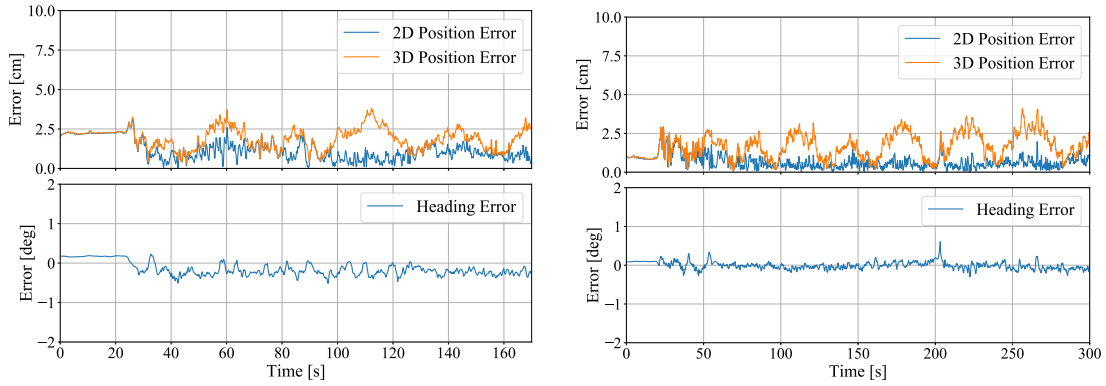


Figure 9.2: Local filter fusing GNSS and the IMU: deviation from the reference trajectory of the 2D/3D positions and headings for (a) the first windrow, and (b) the second windrow.

The second local filter uses the pose which is obtained from registering the 3D point cloud recorded by the stereo camera to the 3D point cloud of the composting site with NDT in the measurement update. The readings from the encoders are used to propagate the state vector in the time update of the filter. Figure 9.3 shows the accuracy achievable with the second local filter. The left side (a) shows the position and heading errors for the first windrow, and the right side (b) shows the position and heading errors for the second windrow. The accuracies achieved by the second local filter are worse than those

achieved by the first local filter. The position errors are up to 50 cm and the heading error is up to 5°.

For both windrows, the position error of the second local filter increases with time. To understand why this is happening, we have to take a closer look at the test setup and the 3D point cloud of the composting site (Figure 8.6). The 3D point cloud contains the windrows along which the compost turner drove during the tests, as well as a region around them. When the compost turner is placed in front of a windrow, the stereo camera observes a point cloud straight ahead in the direction of the windrow that can be matched to the 3D point cloud of the composting site. When the compost turner is in the middle of the windrow, the point cloud is homogeneous in the driving direction and difficult to register. Towards the end of the windrow, the stereo camera observes features which are located beyond the borders of the 3D point cloud. When this happens, the observed point cloud can no longer be registered to the point cloud of the site. When the registration fails, the state vector is still propagated with data from the rotary encoders and the odometry model for tracked vehicles. However, without an update of a position fixing sensor and only with dead reckoning observations, the state vector estimated by the filter starts to drift.

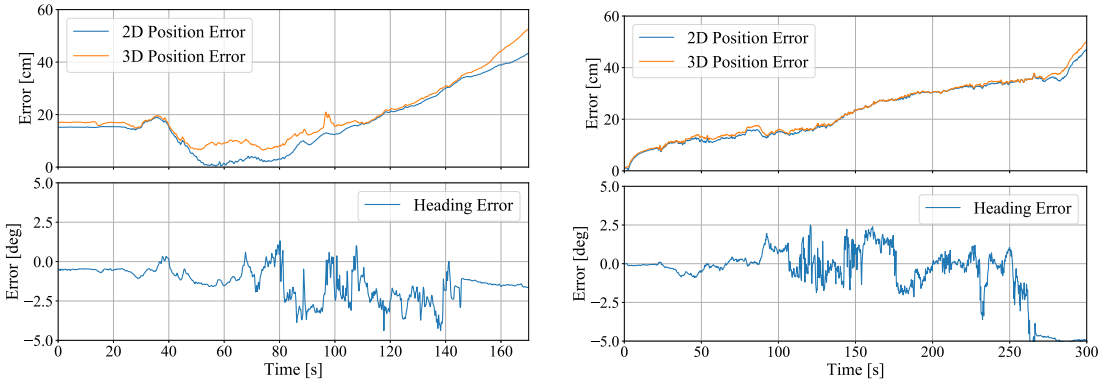


Figure 9.3: Local filter fusing NDT and odometry: deviation from the reference trajectory of the 2D/3D positions and headings for (a) the first windrow, and (b) the second windrow.

The snapshot fusion algorithm fuses the estimated state vectors of both local filters (Figures 9.2 and 9.3) and therefore contains information from all sensors used. Figure

9.4 shows the 2D and 3D position errors and the heading errors for the first windrow (a) and the second windrow (b) of the trajectories obtained with the snapshot fusion approach. For both windrows, the heading deviates less than 1° from the reference. The 2D position error is below 10 cm; however, a drift similar to the one of the second local filter (Figure 9.3) is noticeable. Since the snapshot fusion is strongly influenced by the results of the first filter (as they have a lower standard deviation), the drift is not as strong as for the second filter. Nevertheless, the influence of the second local filter deteriorates the overall solution.

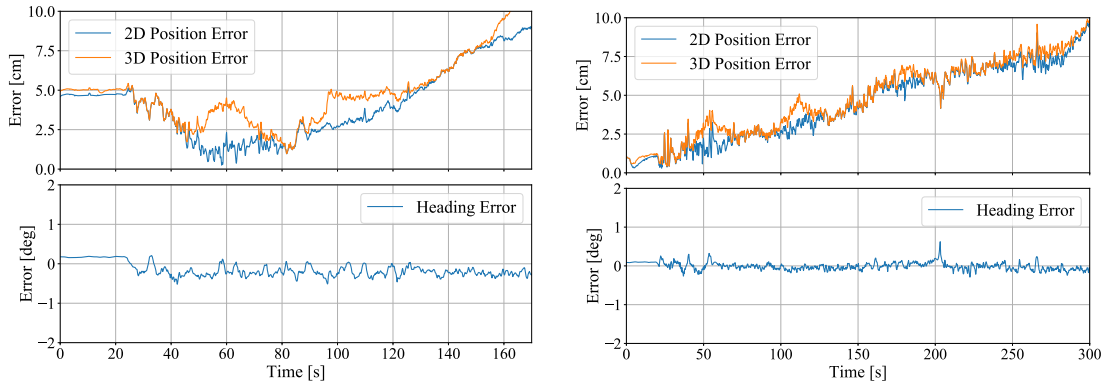
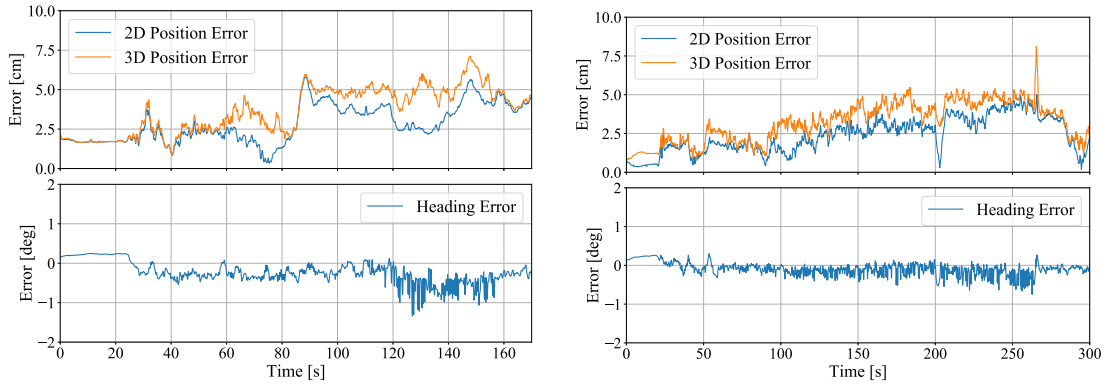


Figure 9.4: Result of the snapshot fusion combining the estimates of both local filters: deviation from the reference trajectory of the 2D/3D positions and headings for (a) the first windrow, and (b) the second windrow.

9.1.2 Analysis of the Results of the Cascaded Filter

The cascaded filter fuses pre-processed information from all sensors in one error-state EKF. Figure 9.5 shows how the trajectory estimated by the cascaded filter deviates from the reference trajectory. The upper part of the figure shows the 2D and 3D position errors while the lower part shows the heading errors. The left side (a) shows the results for the first windrow and the right side (b) the results for the second windrow. For both windrows, the 2D and 3D position errors are below 10 cm. In contrast to the results of the federated filter, the position estimated by the cascaded filter is not affected by drift. For the second windrow, the heading deviates less than 1° from the reference trajectory. For the first windrow, the heading error is below 1° most of the time. There is a short

period of outliers occurring between the seconds 125-130, where the maximum heading deviation is -1.34°.



(a) **(b)**

Figure 9.5: Result of the cascaded integration architecture: deviation from the reference trajectory of the 2D/3D positions and headings for (a) the first windrow, and (b) the second windrow.

9.1.3 Comparison of the Results of the Federated and the Cascaded Integration Architecture

When we compare the results of the federated filter (Figure 9.4) to those of the cascaded filter (Figure 9.5), we can see that the federated filter yields a more accurate estimate of the heading. However, the cascaded filter shows lower 2D and 3D positioning errors and is not affected by drift.

The federated filter allows evaluating individual components that contribute to the navigation solution, e.g. only the GNSS/INS fusion or only the NDT/odometry fusion. Furthermore, it allows detecting when the individual navigation solutions of the local filters diverge. However, it cannot detect which of the two navigation solutions that it fuses is correct and which one is affected by errors.

The cascaded filter fuses pre-processed information from all sensors in one Kalman filter. This makes it more difficult to evaluate how much individual sensors contribute to the overall navigation solution, but it allows for better outlier detection. For example,

when the estimated pose from NDT does not sufficiently match the predicted pose, the cascaded filter discards this information and then only computes the new state vector from GNSS, the IMU, and odometry.

When we consider that the filter is to be used in an automatic steering system for tracked compost turners, both filters achieve a sufficiently accurate solution for the position and heading. However, when outliers occur, the federated filter only allows detecting that the local navigation solutions drift apart, but since the snapshot fusion only fuses two state vectors, it cannot tell which one is affected by the outlier. This means that for an automatic steering system, the machine would have to be stopped when the two independently estimated state vectors start to drift. The cascaded integration architecture allows detecting and eliminating erroneous measurements and should therefore be preferred for the automatic steering system.

9.1.4 Simulation of GNSS Outages

The results of the filter evaluation with the data from the first tests showed that the accuracy achievable for the positioning solution heavily depends on the accuracy of GNSS-RTK. However, GNSS outages can always occur, be it through intentional or unintentional interference.

Composting sites are often located close to motorways. A study by Stanisak et al. (2016) analysed the impact of GNSS jamming on a receiver located next to a motorway. The study showed that the carrier-to-noise ratio of the receiver can drop for a duration of up to 30 seconds when a vehicle with a jammer passes by.

GNSS outages of up to 30 seconds cannot be bridged with sufficient accuracy for automatic steering with MEMS IMUs (Zhang et al., 2021). Therefore, a study was carried out in the course of this thesis to analyse whether outages of this length can be bridged with the tailored odometry model. For this study, a cascaded filter was set up which uses the odometry model in the time update and observations from GNSS and the IMU in the measurement update. The filter was tested with data from the first tests. To analyse how well GNSS outages can be bridged, 30-second GNSS outages were simulated in

post-processing. The accuracy achievable with and without outages was then analysed by comparing the estimated positions and attitudes to the reference trajectory.

In the following, the filter architecture and the results are described. The results were already published in Reitbauer and Schmied (2021b).

Filter Architecture

The proposed filter is a cascaded filter architecture based on an error-state EKF (Figure 9.6). The error state vector contains the position errors in a local-level North-East-Down-frame and the attitude errors expressed as Euler angles (see Equation 7.39). The filter uses the tailored odometry model for tracked vehicles with slip estimation in the time update (see Subsection 7.1.2). The GNSS position, velocity, and heading are used in the measurement update (see Subsection 7.1.1). In contrast to the previously introduced filter architectures in Chapter 7, the IMU is now used in the measurement update.

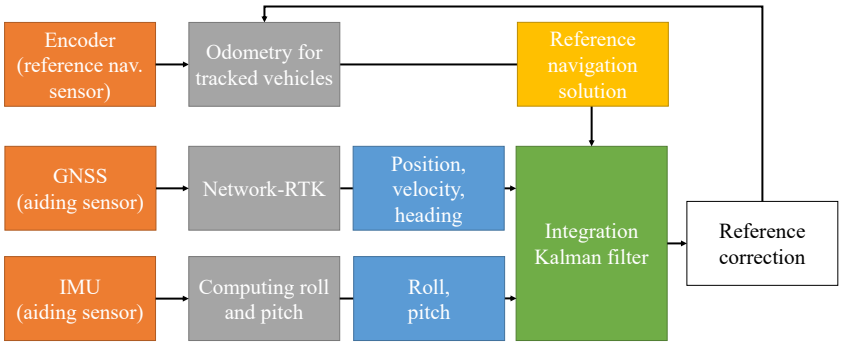


Figure 9.6: Cascaded integration architecture for fusing GNSS, IMU, and encoder data.

The accelerations measured by the IMU are used to compute the angles roll (θ_{IMU}) and pitch (ϕ_{IMU}) using Equations 6.3 and 6.4. Note that θ_{IMU} and ϕ_{IMU} are pre-filtered with an Alpha-Beta filter.

For the Alpha-Beta filter, we model the angle (e.g. the roll angle θ) and a velocity or angle change ($\dot{\theta}$). These two parameters describe the state $\mathbf{y}_t$ at an epoch t:

$$\mathbf{y}_t = \begin{pmatrix} \theta_t \\ \dot{\theta}_t \end{pmatrix}. \tag{9.4}$$

The observation ($\mathbf{l}_t$) at each epoch t is the unfiltered angle (e.g. the roll angle $\theta_{IMU,t}$) computed from the accelerations observed by the IMU:

$$\mathbf{l}_t = \begin{pmatrix} \theta_{IMU,t} \end{pmatrix}. \tag{9.5}$$

The state is estimated from the previously estimated state ($\hat{\mathbf{y}}_{t-1}$) and the new observation ($\mathbf{l}_t$) as follows:

$$\hat{\mathbf{y}}_t = \mathbf{F}\hat{\mathbf{y}}_{t-1} + \begin{pmatrix} \alpha \\ \beta/\tau \end{pmatrix} (\mathbf{l}_t - \begin{pmatrix} 1 & 0 \end{pmatrix} \mathbf{F}\hat{\mathbf{y}}_{t-1}), \tag{9.6}$$

where α, β are the parameters Alpha and Beta of the Alpha-Beta filter that determine how much influence the observations have and how strongly the result is smoothed. τ is the time difference from epoch $t-1$ to epoch t, and the matrix $\mathbf{F}$ has the following form:

$$\mathbf{F} = \begin{pmatrix} 1 & \tau \\ 0 & 1 \end{pmatrix}. \tag{9.7}$$

Figure 9.7 shows the time series for roll (θ_{IMU}) and pitch (ϕ_{IMU}) computed from the IMU observations for the first (a) and second (b) windrows. The unfiltered angles are shown in blue, and the filtered angles are shown in orange. An Alpha-Beta filter with $\alpha = 10^{-2}$ and $\beta = 10^{-8}$ was used to obtain the filtered angles. This strong filtering is necessary since the vibrations of the compost turner cause the accelerometer measurements to be noisy.

The filtered angles $\theta_{IMU,filt,t}$ and $\phi_{IMU,filt,t}$ for each epoch t with IMU data are used to compute the reduced observations

$$\delta\mathbf{z}_{IMU,t} = \begin{pmatrix} \tilde{\theta}_{ODO,t} - \theta_{IMU,filt,t} \\ \tilde{\phi}_{ODO,t} - \phi_{IMU,filt,t} \end{pmatrix}, \tag{9.8}$$

where $\tilde{\theta}_{ODO,t}$ and $\tilde{\phi}_{ODO,t}$ are the predicted roll and pitch angles using the odometry model.

The observation noise is modelled with

$$\mathbf{R}_{IMU,t} = \begin{pmatrix} \sigma^2_{\theta IMU} & 0 \\ 0 & \sigma^2_{\phi IMU} \end{pmatrix}, \tag{9.9}$$

where $\sigma^2_{\theta IMU}$ and $\sigma^2_{\phi IMU}$ are the variances for roll and pitch in $[radians^2]$.

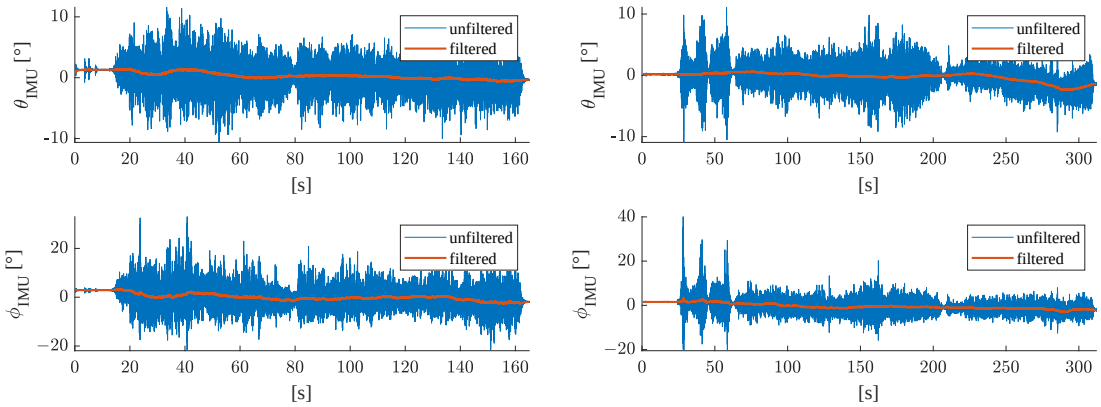


Figure 9.7: Attitude angles roll (θ_{IMU}) and pitch (ϕ_{IMU}) computed from accelerations measured with the XSens MTi-G-710. Results for (a) the first windrow, and (b) the second windrow.

Results: Cascaded Filter with the Odometry Model in the Time Update without Outages

To evaluate the accuracy achievable, the estimated positions of the filter are subtracted from the reference trajectory. The North deviation from the reference $\Delta N(t)$ is computed from

$$\Delta N(t) = n_r(t) - n_f(t), \tag{9.10}$$

where $n_r(t)$ is the North coordinate of the reference at epoch t and $n_f(t)$ is the North coordinate of the filter result at epoch t. The East deviation from the reference $\Delta E(t)$ is obtained from

$$\Delta E(t) = e_r(t) - e_f(t), \tag{9.11}$$

where $e_r(t)$ is the East coordinate of the reference at epoch t and $e_f(t)$ the East coordinate of the filter at epoch t. The horizontal positioning error (RMS_{2D}) is computed according to Equation 9.1. The heading deviation from the reference trajectory ($\Delta\psi$) is computed from Equation 9.3.

Figure 9.8 shows the accuracy achievable for the proposed filter for (a) the first windrow, and (b) the second windrow. The North and East deviations of the filtered trajectory from the reference, as well as the 2D positioning error (RMS_{2D}) are shown as time series. For both windrows, the maximum 2D positioning error is below 10 cm.

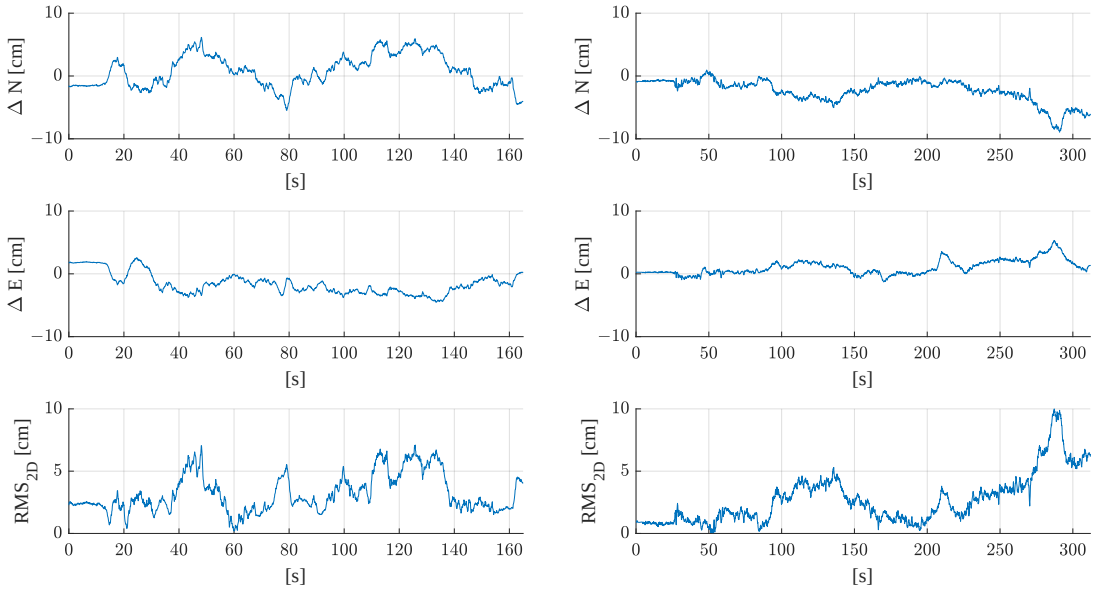
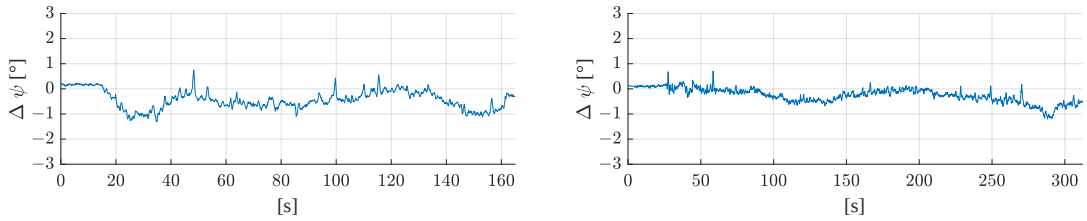


Figure 9.8: Deviation from the reference trajectory in North (ΔN), East (ΔE), and 2D positioning error (RMS_{2D}) for (a) the first windrow, and (b) the second windrow.

Figure 9.9 shows the heading deviation from the reference heading for both windrows as time series. The maximum deviation is 1.3° for the first windrow and 1.2° for the second windrow.



(a) **(b)**

Figure 9.9: Deviation of the filtered heading from the reference heading ($\Delta\psi$) for (a) the first windrow, and (b) the second windrow.

In Table 9.1, the mean deviation of the filtered solutions in North, East, and heading, as well as the 2D positioning error is shown. The results are given for both windrows as the mean value ± a standard deviation. For both windrows, a similar accuracy was achieved for the horizontal position, with a mean 2D error of 3.2 cm for the first and 2.7 cm for the second windrow. The mean heading deviation from the reference was -0.4° for the first and -0.3° for the second windrow.

Table 9.1: Mean deviation of the filtered trajectory from the reference trajectory for both windrows.

Deviation	**Windrow 1**	**Windrow 2**
North (ΔN)	0.8 ± 2.6 cm	-2.4 ± 1.8 cm
East (ΔE)	-1.6 ± 1.7 cm	1.0 ± 1.2 cm
Horizontal (RMS_{2D})	3.2 ± 1.5 cm	2.7 ± 2.0 cm
Heading ($\Delta\psi$)	-0.4 ± 0.4°	-0.3 ± 0.3°

Results: Accuracy Achievable with 30 Seconds of GNSS Outages

To analyse how well the odometry model is suited to bridge GNSS outages, GNSS outages with a duration of 30 seconds were simulated in post-processing. In Figure 9.10 the positions of the GNSS antenna which were used as observations in the filter are shown in orange. The gap where no GNSS observations were present is larger for the first windrow (a) than for the second (b) windrow, since the driving speed was higher for the first windrow than for the second one (see Figure 9.1). The trajectory estimated by

the filter is shown in blue. Note that there is an offset between the filtered trajectory and the GNSS positions, which is due to the fact the GNSS position antenna was mounted on the left side of the vehicle with a lever arm of 1.423 m to the centre (Figure 8.4) while the filtered solution refers to the centre of the vehicle.

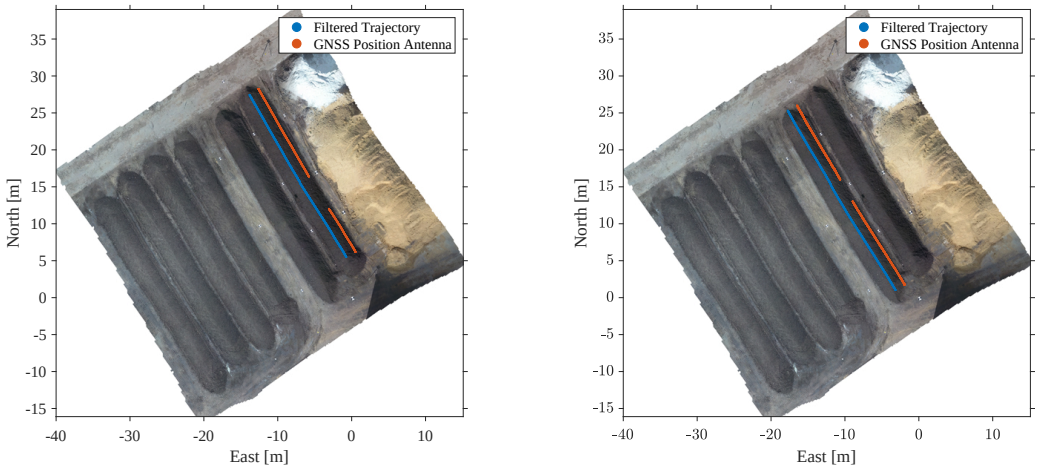


Figure 9.10: Simulated 30 second GNSS outages for (a) the first windrow, and (b) the second windrow. The available GNSS positions are shown in orange, the trajectory resulting from the Kalman filter is shown in blue.

Figure 9.11 shows the North (ΔN) and East (ΔE) deviations of the filtered trajectory from the reference trajectory as time series for both windrows. The 2D horizontal positioning error (RMS_{2D}) is also shown. The red boxes indicate the epochs where no GNSS observations were present. For both windrows, the 2D horizontal positioning error is below 10 cm. The maximum horizontal deviation from the reference is 8.7 cm for the first windrow and 8.3 cm for the second windrow. In the periods where no GNSS observations are available, the 2D positioning error grows. When no GNSS observations are present, the position of the vehicle is only estimated from the odometry model. As odometry is a dead reckoning technique, errors accumulate with time and cause the position error to grow. For the first windrow (part (a) of Figure 9.11), the positioning error grows more than for the second windrow (part (b) of Figure 9.11). This can be explained by the different driving speeds (Figure 9.1). The compost turner was faster when it was driving through the first windrow than when it was driving through the

second windrow. For unaided odometry, the error grows with revolutions of the wheels or tracks. Since the vehicle was steered at a higher speed for the first windrow, more track revolutions were completed during the 30 seconds of GNSS outage for the first windrow.

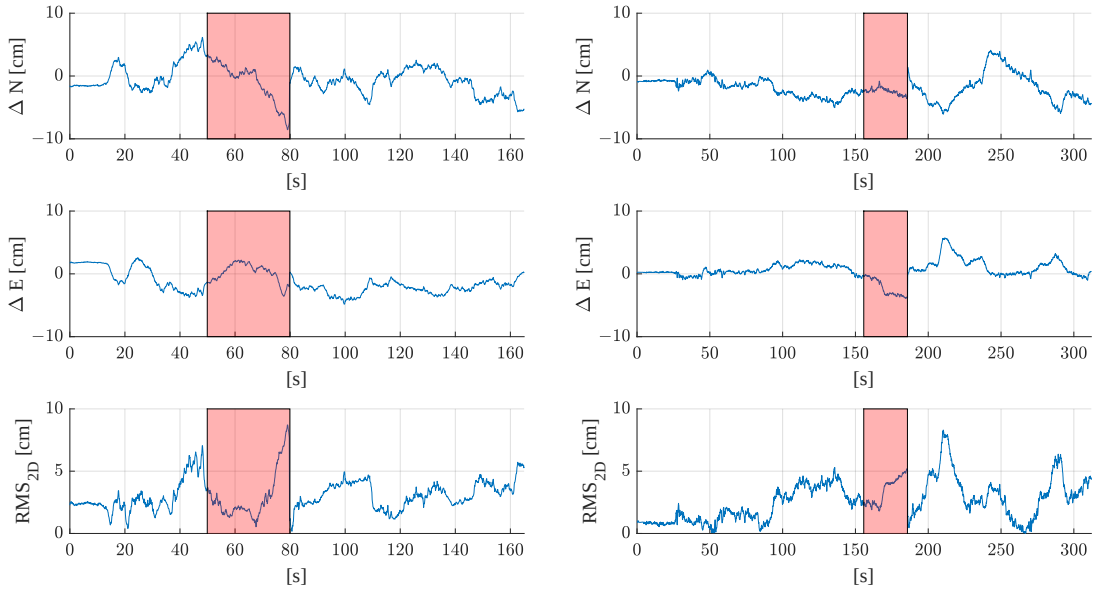
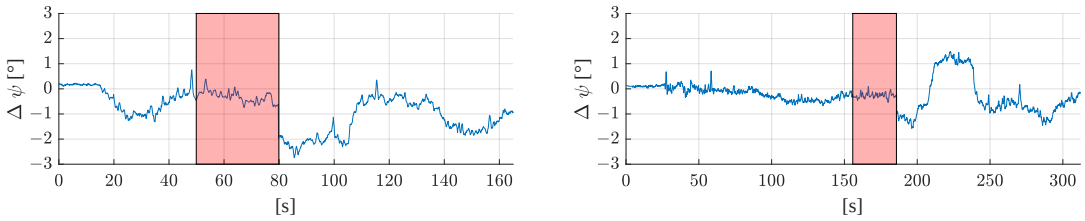


Figure 9.11: Deviation from the reference trajectory with simulated GNSS outages of 30 seconds in North (ΔN), East (ΔE), and 2D positioning error (RMS_{2D}) for (a) the first windrow, and (b) the second windrow. The epochs where no GNSS was available are shown in red.

Figure 9.12 shows the deviation of the filtered heading from the reference heading as time series for the first (a) and second (b) windrows. Again, the period where no GNSS observations were present is indicated with red boxes. During the GNSS outages, the estimated heading hardly deviates from the heading of the reference trajectory.



(a) **(b)**

Figure 9.12: Deviation of the filtered heading from the reference heading ($\Delta\psi$)for (a) the first windrow, and (b) the second windrow. The red areas indicate the periods where GNSS outages were simulated.

Table 9.2 shows the mean deviation of the North and East coordinates estimated by the filter from the reference trajectory. It also presents the mean horizontal error and the heading deviation from the reference. All results refer to the trajectories where 30 seconds of GNSS outages were simulated. When comparing these results to the results without GNSS outages (Table 9.1), we can see that the horizontal positioning errors are in a similar range. For the first windrow, the mean horizontal error without outage was 3.0 cm and with outage it is 3.2 cm. The mean horizontal error of the second windrow was 2.7 cm without outage and 2.6 cm with the outage. The heading deviation from the reference was lower when no GNSS outages occurred. For the first windrow, the mean heading error with standard deviation was $-0.4 \pm 0.4°$ without outage and $-0.8 \pm 0.7°$ with the outage. For the second windrow, the mean heading error was $0.3 \pm 0.3°$ without outage and $-0.2 \pm 0.6°$ with the outage.

Table 9.2: Mean deviation of the filtered solution from the reference trajectory with simulated GNSS outages of 30 seconds for both windrows.

Deviation	**Windrow 1**	**Windrow 2**
North (ΔN)	-0.7 ± 2.4 cm	-1.9 ± 1.9 cm
East (ΔE)	-1.2 ± 1.8 cm	0.6 ± 1.5 cm
Horizontal (RMS_{2D})	3.0 ± 1.3 cm	2.6 ± 1.6 cm
Heading ($\Delta\psi$)	-0.8 ± 0.7°	-0.2 ± 0.6°

Discussion of the Results

The aim of the study carried out in Subsection 9.1.4 was to find out how well the odometry model for tracked vehicles is suited to bridge GNSS outages. Therefore, a cascaded error state EKF was set up which uses the tailored odometry model in the time update. The EKF uses the angles roll and pitch computed from IMU observations in the measurement update. When an RTK solution with fixed ambiguities is available, the filter uses the GNSS position, velocity, and heading in the measurement update.

To analyse the accuracies achievable, the filter was tested with the dataset recorded in March 2021 at the composting site in Riedlingsdorf, Austria. The estimated positions and heading angles of the filter were compared to the positions and heading angles of the reference trajectories obtained through tracking with two total stations. The results showed that a horizontal positioning error of less than 10 cm was achieved. The maximum heading deviation from the reference heading was 1.3°.

To evaluate how well the tailored odometry model for tracked vehicles is suited to bridge GNSS outages, GNSS outages with a duration of 30 seconds were simulated in post-processing. Since observations from the IMU are only used to compute the angles roll and pitch, the position and heading changes are computed from the tailored odometry model and the track speeds measured by the rotary encoders. The results showed that after 30 seconds without GNSS, a horizontal positioning accuracy of less than 10 cm can still be achieved.

9.2 Results of the Final Tests

For the final tests, the functionality of the automatic steering system had to be demonstrated. To do so, a reliable navigation filter was needed. The final tests were also the tests where the compost turner turned the windrows with its spiked drum for the first time. It was therefore expected that strong vibrations would occur. For this reason, it was decided that the IMU should not be used as the reference navigation sensor, but only as an aiding sensor to provide roll and pitch for the measurement update. Since there were problems with registering the point cloud of the stereo camera to the point

cloud of the composting site during the first tests (due to the lack of distinct features and the homogeneous appearance of the point cloud in the driving direction), it was also decided not to include the NDT algorithm in the real-time filter for demonstration, but to analyse it later in post-processing.

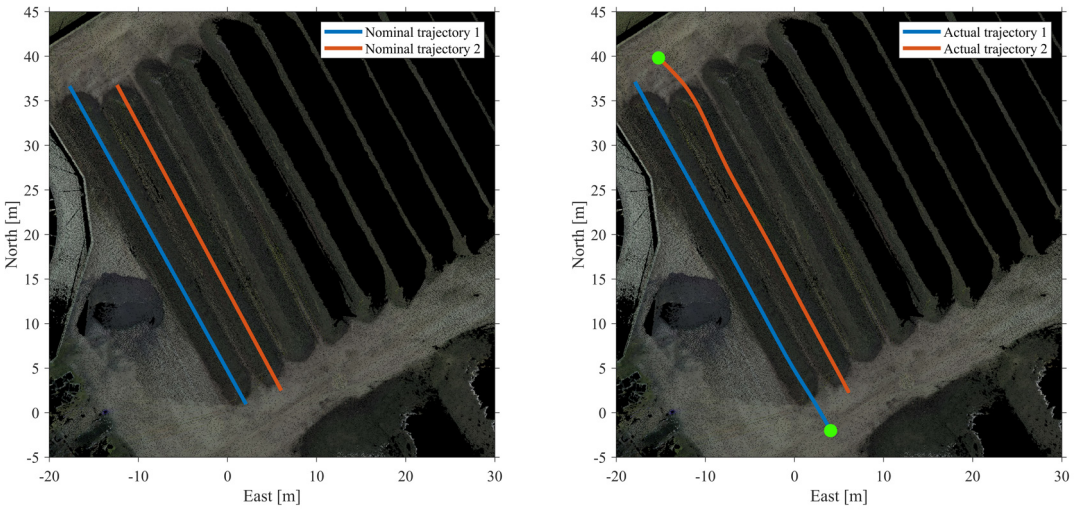
The filter selected for the demonstration of the automatic steering system is the cascaded filter architecture which uses the tailored odometry model in the time update, and GNSS and the IMU in the measurement update (see Subsection 9.1.4).

In the following, the results of the final tests will be analysed, using the following structure: First, the trajectories are shown along which the compost turner was steered for the demonstration of the automatic steering system (Subsection 9.2.1). Then, the measured speeds and accelerations are analysed and compared to those of the first tests (Subsection 9.2.2). In Subsection 9.2.3, the accuracy achieved with the filter fusing GNSS, IMU, and odometry is analysed. Then, the observations with the stereo camera and the NDT registration algorithm are added and the results of the filter fusing all sensors are analysed in Subsection 9.2.4. Lastly, the results of the tests are discussed in Subsection 9.2.5.

9.2.1 Demonstration of the Automatic Steering System

Figure 9.13 shows the nominal and actual trajectories of the tests where the automatic steering system was demonstrated. The left side (a) shows the nominal trajectories that were computed prior to the tests. The right side (b) shows the actual trajectories that the compost turner followed while it was being steered by the automatic steering system. The green circles indicate the start of the two trajectories.

Note that for the first trajectory (blue line), the compost turner was steered along a straight line, while for the second trajectory (orange line), it drove a slightly curved path in the beginning. This difference can be explained by taking a closer look at the orientation that the compost turner had with respect to the nominal trajectory before the automatic steering system was started. Table 9.3 shows the nominal and actual headings at the beginning of the trajectory. For the first trajectory, the nominal heading was -29.0° and the actual heading of the machine was -30.0° when the automatic



(a) **(b)**

Figure 9.13: Trajectories of the final tests in September 2021: (a) nominal trajectories, (b) actual trajectories.

Table 9.3: Nominal and actual headings at the beginning of the trajectories before the automatic steering system was started.

Heading at the Start	**Trajectory 1**	**Trajectory 2**
Nominal	-29.0°	151.7°
Actual	-30.0°	130.9 °

steering system was started, so there was only a heading difference of 1° that had to be corrected. For the second trajectory, the nominal heading was 151.7° and the actual heading was 130.9°. This means that there was a heading difference of 20.8° which had to be corrected by the automatic steering system.

During the demonstration, the compost turner was automatically steered along both trajectories. While it was being steered, it was turning the windrows with its spiked drum. The compost turner kept a constant speed and the correct orientation, so the windrows were kept in the same shape and at the same location as before the turning process. The automatic steering system therefore achieved a similar result as a human operator. A picture of the shape of the windrows after being turned by the automatically steered compost turner is shown in Figure 9.14.

Figure 9.14: Picture of the first windrow (on the right side) after being turned by the automatically steered compost turner, and the second windrow (on the left side) while being turned.

9.2.2 Analysis of the Raw Data

As already mentioned, the final tests were the tests where the compost turner turned the windrows with its spiked drum for the first time. Moreover, it was also the first time the compost turner was steered by the automatic steering system. This led to two differences in the navigation data. First, the speed of the compost turner was lower than in the first tests; second, the accelerations recorded by the IMU were higher due to the strong vibrations of the machine.

Figure 9.15 shows the speed of the compost turner as it was steered along the two trajectories during the final tests. The upper part of the figure shows the speeds output by the Trimble BD992 GNSS receiver while the lower part shows the left and right track speeds computed from the encoder readings. When comparing the speeds to the speeds of the first tests (Figure 9.1), it can be noted that the speeds of the final tests were even lower. For the first tests, the speeds were between 0.1-0.2 m/s, for the final test the speed was 0.05 m/s.

When taking a closer look at the speeds in Figure 9.15, it can be seen that the speed output by the Trimble BD992 GNSS receiver is lower than that computed from the encoder readings. A reason for this might be that the diameter that was assumed for

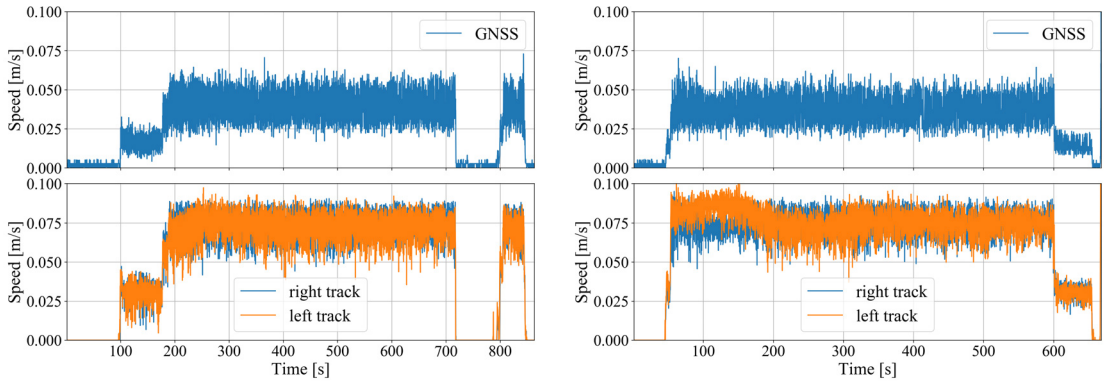


Figure 9.15: Speed of the compost turner during the tests in September for (a) the first trajectory, and (b) the second trajectory.

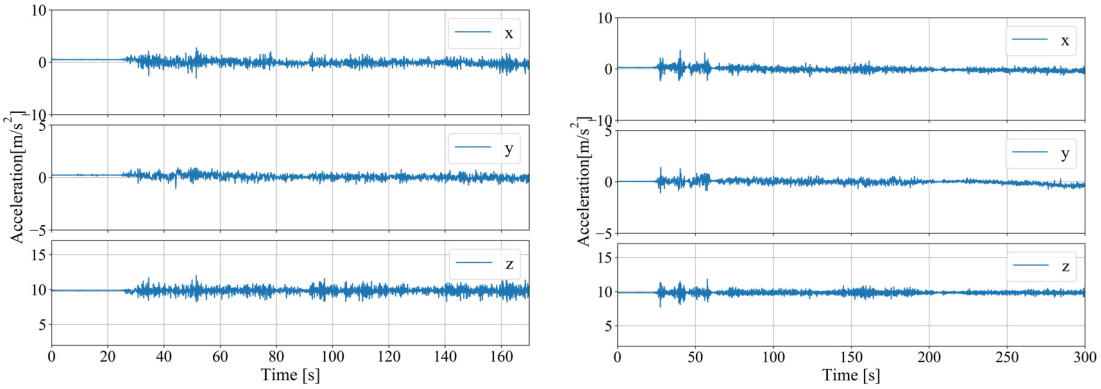
the track wheel was too large. The diameter of the track wheel is multiplied by the radial velocity measured by the encoders in Equation 7.40 to obtain the track speed. If the assumed diameter is too large, the track speeds computed with Equation 7.40 are too high. Another reason might be that the tracks slip as the compost turner turns the windrows with its spiked drum, therefore the speed computed from the track revolutions is higher than the actual speed. This needs to be accounted for (e.g. by estimating a scale factor for odometry with the EKF).

Figure 9.15 (b) shows the speeds for the second trajectory. Note that the track speeds of the left and right tracks differ in the beginning, as the compost turner drove a slightly curved path (see Figure 9.13 (b)). The speed of the left track is higher than that of the right track since the compost turner turned to the right.

When the compost turner turns the windrows with its spiked drum, it is subject to strong vibrations. This can be seen in the IMU data, specifically in the accelerations that were measured with the XSens MTi-G-710.

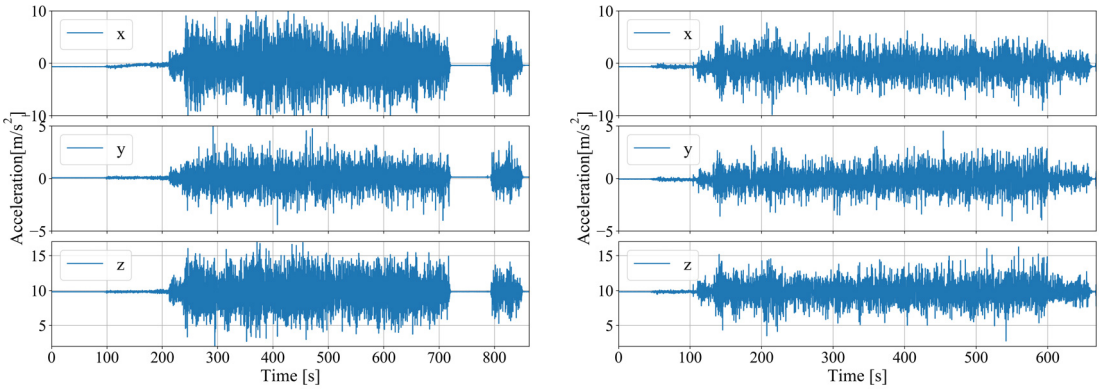
Figure 9.16 shows the accelerations recorded with the XSens MTi-G-710 during the first tests in March 2021. Figure 9.17 shows the accelerations recorded by the same XSens MTi-G-710 during the final tests in September 2021. When comparing these two figures, it can be seen that the noise of the accelerations of the test in September is higher. The

noise increases when the spiked drum starts rotating (after 215 seconds for the first trajectory and after 110 seconds for the second trajectory).



(a) **(b)**

Figure 9.16: Accelerations measured by the XSens MTi-G-710 IMU during the first tests in March 2021 for (a) the first trajectory, and (b) the second trajectory.



(a) **(b)**

Figure 9.17: Accelerations measured by the XSens MTi-G-710 IMU during the final tests in September 2021 for (a) the first trajectory, and (b) the second trajectory.

9.2.3 Accuracy Achievable Through Fusing GNSS, IMU, and Odometry

In the following, the accuracy achievable with the cascaded filter fusing GNSS, IMU and odometry will be analysed, showing the 2D and 3D position errors (RMS_{2D} and RMS_{3D}), as well as the heading deviation from the reference trajectory. These values were computed according to Equations 9.1 to 9.3.

As mentioned in the previous subsection, the speeds obtained from the rotary encoders were higher than the speeds obtained from GNSS, due to an overly optimistic assumption of the diameter of the track wheel. This systematic error has to be taken into consideration, e.g. by estimating a scale factor.

What happens when this systematic error is not accounted for and no scale factor is estimated is shown in Figure 9.18. The figure shows the 2D and 3D position errors, as well as the heading deviation from the reference trajectory. The position error for both trajectories is large. For the first trajectory, where the compost turner was steered along a straight line, the position error increases with time and reaches up to 30 cm towards the end of the windrow. For the second trajectory, where the compost turner was steered along a curved path, the position error is even larger (>50 cm).

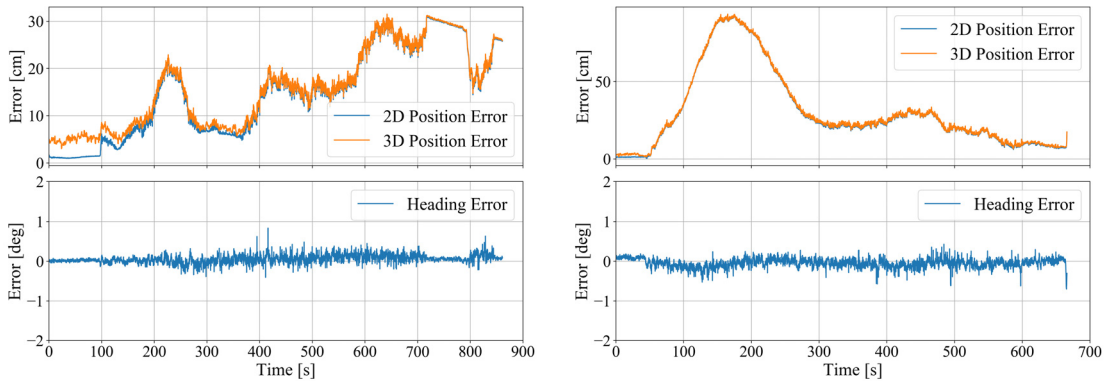


Figure 9.18: Results of the GNSS, IMU, and odometry integration without a scale factor for odometry. Deviation of the filtered solution from the reference trajectory as 2D/3D position errors and heading deviation for (a) the first trajectory, and (b) the second trajectory.

Figure 9.19 shows the results of the cascaded filter fusing GNSS, IMU, and odometry when a scale factor for odometry is estimated by the filter, giving the 2D and 3D position errors, as well as the heading deviation from the reference. When comparing these results with the results without a scale factor (Figure 9.18), it can be seen that the overall accuracy increases when a scale factor for odometry is estimated by the EKF. However, when the results are compared to the results of the first tests, where the compost turner did not turn the windrows with its spiked drum, it can be seen that the same 2D accuracy (< 10 cm) as for the first tests cannot be reached. The maximum 2D position error is 16.1 cm for the first trajectory and 22.8 cm for the second trajectory. The mean 2D and 3D errors as well as the heading deviation from the reference with their respective standard deviations are shown in Table 9.4.

Table 9.4: Mean deviation of the cascaded filter fusing GNSS, IMU, and odometry from the reference trajectory for the final tests.

Deviation	**Trajectory 1**	**Trajectory 2**
Horizontal (RMS_{2D})	6.0 ± 3.6 cm	6.9 ± 5.0 cm
3D (RMS_{3D})	7.5 ± 2.8 cm	8.1 ± 4.9 cm
Heading ($\Delta\psi$)	0.05 ± 0.09°	-0.04 ± 0.10°

The largest error occurs for the second trajectory from seconds 100-200, where the compost turner was steered along a curved path (see Figure 9.15, where the track speeds of the left and right tracks differ for this time period). One reason for this might be that the slip ratios are not estimated with sufficient accuracy. The slip ratios are estimated from the angular rate of the compost turner around its x_3-axis (see Equations 7.45 and 7.46) and from an additional constraint (see Equation 7.44). However, the angular rates measured with the IMU are also influenced by the vibrations as the compost turner turns the windrows. This is illustrated in Figure 9.20. The Figure shows the angular rates recorded by the XSens MTi-G-710 during the final tests as the compost turner was steered along the second trajectory. The angular rate about the x_3-axis (denoted with z in the plot) is strongly affected by noise. Another reason might be that a common scale factor for odometry is estimated for both tracks. Differences in scale factors for the left and right tracks are not taken into consideration and might lead to a larger error when the vehicle drives a curved path.

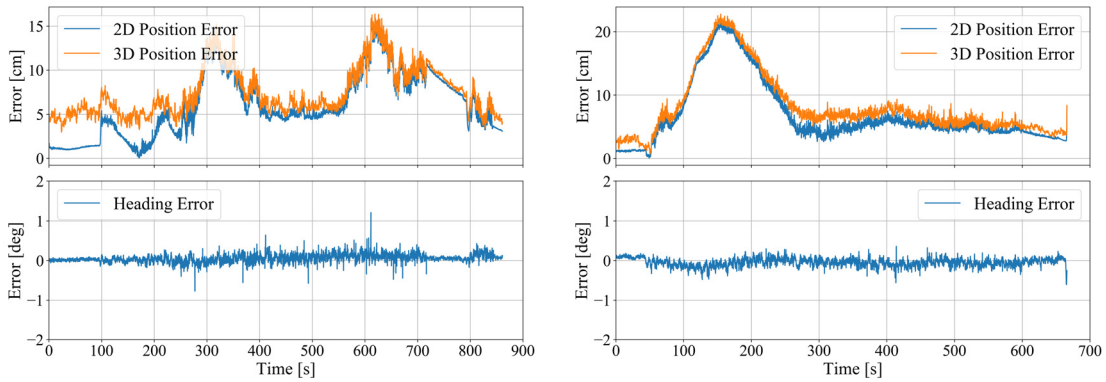


Figure 9.19: Results of the GNSS, IMU, and odometry integration with a scale factor for odometry. Deviation of the filtered solution from the reference trajectory as 2D/3D position errors and heading deviation for (a) the first trajectory, and (b) the second trajectory.

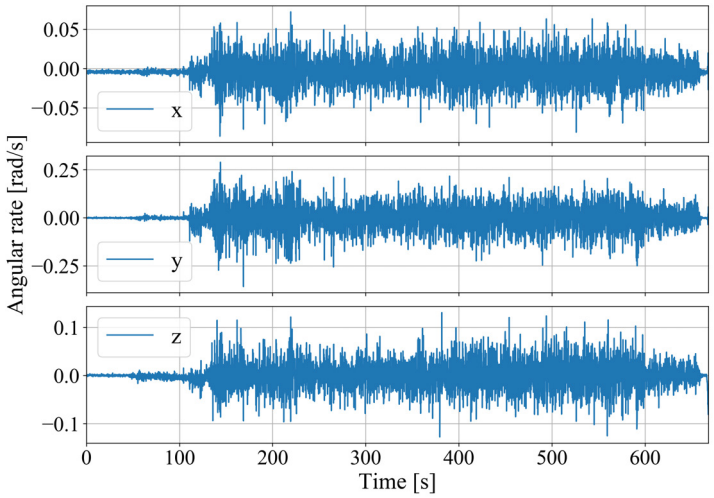


Figure 9.20: Angular rates measured by the XSens MTi-G-710 IMU during the final tests for the second trajectory.

Even though a horizontal positioning accuracy of < 10 cm could not be reached, a heading accuracy of $< 1°$ was achieved (Figure 9.19). This accuracy was sufficient for the automatic steering system to work at the demonstration.

Bridging GNSS Outages

As for the first tests, 30-second GNSS outages were also simulated for the final tests in post-processing. Figure 9.21 shows the 2D and 3D position errors as well as the heading deviation from the reference as time series. The red box indicates the time span where no GNSS observations were used. Compared to the results without GNSS outages (Figure 9.19), it can be seen that a similar accuracy is achieved for the 2D and 3D positions while the heading error increases during the simulated GNSS outages. For the first trajectory, the heading deviation from the reference reaches a maximum of 2.96°, for the second trajectory a maximum of 0.92°. These heading errors are noticeably larger than for the first tests. One reason for this might be that the compost turner followed a more curved path during the final tests. When a tracked vehicle follows a curved path, slip occurs. The slip ratios are estimated using the angular rate of the IMU about its x_3-axis, which is strongly affected by noise (see Figure 9.20) due to the vibrations of the machine. An inaccurate slip estimation leads to an inaccurate heading estimation, since the slip ratios are used in the dynamic model to compute the heading change (see Equation 7.43).

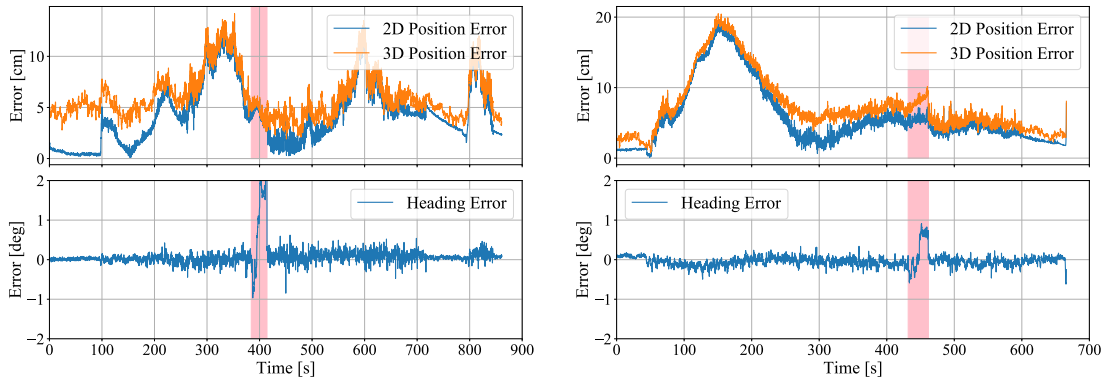


Figure 9.21: Results of the GNSS, IMU, and odometry integration with a 30 second GNSS outage. Deviation of the filtered solution from the reference trajectory as 2D/3D position errors and heading deviation for (a) the first trajectory, and (b) the second trajectory.

9.2.4 Adding Observations from the Stereo Camera

Since there were problems with registering the point cloud of the stereo camera to the pre-recorded point cloud of the composting site during the first tests, the results achievable with the NDT registration algorithm and the stereo camera data of the final tests were only analysed in post-processing.

Figure 9.22 shows the 2D and 3D position errors, as well as the heading deviation from the reference trajectory of the poses estimated with NDT and data from the ZED stereo camera for the first (a) and second (b) trajectories. The filtered trajectory of the GNSS, IMU and odometry fusion was used as an initial estimate of the NDT algorithm. The accuracy decreases towards the end of the trajectories, where fewer features are observable for the stereo camera. At the beginning of the first trajectory, the accuracy is also worse as the compost turner was located so far away from the windrows that the stereo camera was only able to observe a small part of the windrows (see the starting point of the trajectory in Figure 9.13). While the compost turner was driving through the windrows and turning them, the accuracy achievable with NDT was between 25 and 50 cm. The heading deviation from the reference is within $\pm 5°$ for the major part of the time series.

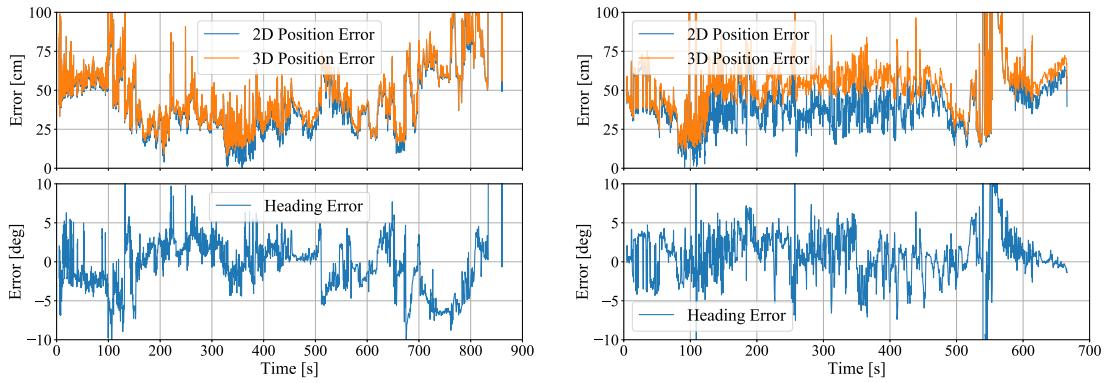


Figure 9.22: Results of the NDT registration with stereo camera data for (a) the first trajectory, and (b) the second trajectory.

Table 9.5 shows the mean deviation of the pose estimated from NDT from the reference trajectory. The horizontal error (RMS_{2D}), the 3D error (RMS_{3D}) as well as the heading deviation from the reference ($\Delta\psi$) are given. The mean 2D error was 49.1 cm for the first and 45.2 cm for the second trajectory.

Table 9.5: Mean deviation of the pose estimated from NDT from the reference trajectory for the final tests.

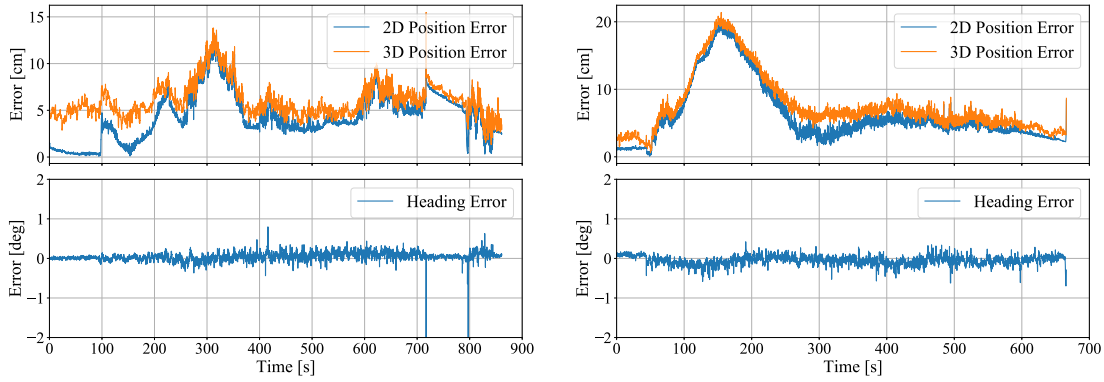
Deviation	**Trajectory 1**	**Trajectory 2**
Horizontal (RMS_{2D})	49.1 ± 37.5 cm	45.2 ± 48.9 cm
3D (RMS_{3D})	65.9 ± 111.4 cm	56.8 ± 54.1 cm
Heading ($\Delta\psi$)	0.82 ± 8.77°	-0.05 ± 13.8°

Figure 9.23 shows the results of a cascaded filter fusing GNSS, IMU, data from rotary encoders with the tailored odometry model, and the pose estimated from NDT with the stereo camera. The tailored odometry model was used in the time update of the filter while the other sensors were used in the measurement update of the filter. Again, the 2D and 3D position errors as well as the heading deviation from the reference heading are shown. Since the pose estimated by NDT, which was used to aid the filter, was not very accurate, only minor improvements in the accuracy were achieved compared to the solution without NDT (Figure 9.19). For the first trajectory (a) the maximum 2D

Table 9.6: Mean deviation of filter fusing GNSS, IMU, NDT, and odometry from the reference trajectory for the final tests.

Deviation	Trajectory 1	Trajectory 2
Horizontal (RMS_{2D})	4.3 ± 2.5 cm	6.1 ± 4.5 cm
3D (RMS_{3D})	6.1 ± 1.8 cm	7.6 ± 4.4 cm
Heading ($\Delta\psi$)	0.05 ± 0.10°	-0.04 ± 0.11°

positioning error is 15.2 cm. For the second trajectory (b) the maximum 2D positioning error is 20.6 cm.



(a) **(b)**

Figure 9.23: Results of the GNSS, IMU, NDT, and odometry integration without a scale factor for odometry. Deviation of the filtered solution from the reference trajectory as 2D/3D position errors and heading deviation for (a) the first trajectory, and (b) the second trajectory.

Table 9.6 shows the mean deviation of the position and heading estimated from the cascaded filter fusing all sensors from the reference trajectory. The mean horizontal error is 4.3 cm for the first and 6.1 cm for the second trajectory. This is slightly lower than for the cascaded filter without NDT, where the mean horizontal error was 6.0 cm for the first and 6.9 cm for the second trajectory.

9.2.5 Discussion of the Results of the Final Tests

For the demonstration of the automatic steering system, a cascaded error-state EKF architecture was chosen. The filter used observations from GNSS and the IMU in the measurement update, and the tailored odometry model for tracked vehicles with information from the rotary encoders in the time update. During the demonstration, the compost turner was steered automatically along two pre-computed routes while turning the windrows with its spiked drum. For the first route, the compost turner was placed at an angle of 1° to the nominal trajectory, for the second route it was placed at an angle of 20.8° from the nominal trajectory. In both cases, the automatic steering system performed in a way comparable to that of a human operator.

To analyse the position and heading accuracy of the filter, the trajectories estimated by the filter were compared to a reference trajectory. The mean horizontal deviation from the reference trajectory was 6.0 cm for the first trajectory and 6.9 cm for the second. The mean heading deviation was 0.05° from the reference heading for the first trajectory and -0.04° for the second.

The results also showed the importance of estimating a scale factor for odometry. When a wrong diameter of the track wheel is used to compute the track speeds, the velocities computed from the rotary encoder readings are affected by a bias. As the Kalman filter assumes that all noise is Gaussian, which is not the case when a systematic bias is present, a scale factor for odometry was modelled as an additional parameter of the state vector in this work. Not taking the scale factor into account would lead to large positioning errors.

The initial navigation sensor selection not only included GNSS, IMU, and rotary encoders, but also a stereo camera. The point cloud observed by the stereo camera was registered to the point cloud of the composting site using NDT. The results of the first tests already showed that it was difficult to register the point cloud observed to the point cloud of the composting site, as the shape of the windrows was homogeneous and there were only a few distinct features. Therefore, NDT was only analysed in post-processing for the final tests. Similar problems occurred with the registration: the mean horizontal error of the pose estimated from NDT with stereo camera data was 49.1 cm for the first trajectory and 45.2 cm for the second trajectory. This accuracy is not sufficient for an

automatic steering system. To aid the filter fusing GNSS, IMU, and odometry with a vision-based system, a more accurate sensor and registration algorithm are required. In future investigations, an active sensor such as a LIDAR, which has a wider range and can therefore observe more features, might be used.

The results showed that with a cascaded error-state EKF fusing observations from GNSS, an IMU, and odometry from rotary encoders with a tailored dynamic model, it is possible to estimate the position and attitude of the compost turner in real-time with sufficient accuracy to automatically steer the machine through the windrows of a composting site, even in the case of GNSS outages. In the future, such an automatic steering system might be used at commercial composting sites so that no human operator is exposed to unpleasant working conditions while the windrows are being turned.

Chapter 10

Conclusion and Outlook

10.1 Conclusion

This thesis aimed to develop a multi-sensor positioning system to be used in an automatic steering system for tracked compost turners. Even though automatic steering systems for tractors or harvesters are already commercially available and widely used, no similar system is yet available for tracked compost turners. Since compost turners operate at lower speeds, are subject to strong vibrations, and are tracked vehicles, a tailored automatic steering system had to be developed.

The contributions of this thesis are the selection of suitable navigation sensors, the development of a real-time error-state Extended Kalman Filter (EKF) which fuses information from the sensors selected, and the formulation of a tailored odometry model for skid steering which can be used both in the measurement update and in the time update of an error-state EKF.

In the following, the research questions presented in Chapter 1 will be answered with reference to the structure and the main findings of this thesis.

Which sensors can be used for the precise positioning of compost turners?

Chapter 2 reviewed fundamental aspects of navigation and presented the two main techniques that are used for positioning: position fixing and dead reckoning. For an optimal navigation solution, both techniques have to be combined, as they complement

each other. Chapter 3 presented commonly used navigation sensors and techniques. Among these navigation techniques were positioning with Global Navigation Satellite Systems (GNSS), inertial navigation an Inertial Measurement Unit (IMU), odometry with wheel speed sensors and image-based navigation.

In Chapter 6, a pre-study was described where a set of navigation sensors was tested at a composting site with a real compost turner. Furthermore, a GNSS receiver was tested at the measurement roof of the Geodesy building at Graz University of Technology. To evaluate the accuracy achievable with the navigation sensors, the navigation solutions that were obtained with the sensors were compared to a reference trajectory. Based on the results of the pre-studies, the following navigation sensors were selected: A dual-antenna GNSS receiver with access to a network-RTK service (Trimble BD992), a MEMS IMU (XSens MTi-G-710), a stereo camera (ZED) and the two rotary encoders (2 Channel HallD SDN6.FK10.E05R) of the prototype of the automatically steered compost turner. Since a 3D point cloud of the composting site is available for the automatic steering system, it can also be used as an additional observation.

How can these sensors be combined to optimally estimate the position and attitude of the machine in real-time?

A fundamental task of navigation is positioning, where the state vector of a moving object is determined. The state vector may consist of the coordinates, velocity, and attitude of a moving object. In Chapter 4, Bayes filters were introduced as mathematical tools which allow estimating the state vector of a moving object from navigation sensor data. All Bayes filters rely on the same two computational steps: a prediction step and a measurement update. A distinction was made between Gaussian and non-parametric filters, depending on how the belief, i.e. the conditional probability distribution representing an object's knowledge of its state vector, is modelled. Gaussian filters, such as the Kalman filter, model the belief as multivariate normal distributions, whereas non-parametric filters do not require the belief to follow a fixed functional form.

In Chapter 5, different multi-sensor fusion architectures were presented: centralized, cascaded, and federated integration. In addition, examples of error-state Extended Kalman Filters (EKF) were given.

In Chapter 7, two integration architectures for fusing observations from the sensors selected were chosen to estimate the position and attitude of the compost turner: a modified federated and a cascaded integration architecture. Both integration architectures are based on error-state EKFs to be computationally efficient.

The modified federated integration architecture consists of two local filters. The first local filter fuses observations from GNSS and the IMU. The second local filter fuses observations from the stereo camera, the 3D point cloud of the composting site, and information from rotary encoders. Normal Distributions Transform (NDT) is used to register the point cloud observed by the stereo camera to the point cloud of the composting site in order to estimate the pose (position and attitude) of the compost turner. The estimated pose is used in the measurement update of the second local filter. A tailored odometry model for tracked vehicles, which incorporates measurements from the rotary encoders, is used in the time update of the second local filter. The independently estimated states of both local filters are fused in a snapshot fusion approach.

The cascaded integration architecture fuses pre-processed information from all navigation sensors in one Kalman filter. This allows for better measurement innovation filtering and better outlier detection.

To optimally estimate the position and attitude of the machine in real-time, a cascaded integration architecture fusing GNSS, IMU, and odometry is used.

Which special considerations need to be taken into account for tracked vehicles?

Tracked vehicles are steered differently than wheeled vehicles. They use a technique known as skid-steering, where the speeds of the left and the right track can be controlled independently. When both tracks are steered forward at the same speed, the tracked vehicle follows a straight line. When one track is steered faster than the other one, the vehicle follows a curved path. Tracked vehicles can also rotate around their x_3-axis when one track is steered forward and the other track backward. When tracked vehicles follow a curved path, slip occurs. This has to be taken into consideration when computing odometry information from the track speeds.

In this thesis, a tailored odometry model for tracked vehicles was used, which accounts for slip. The slip ratios for the left and right tracks are estimated from the yaw rate

measured by the IMU, an equation for differential odometry and an additional constraint first derived by Nagatani et al. (2007). The equations for incorporating this model into an error-state EKF were derived in Chapter 7.

How can the positioning performance be evaluated?

To evaluate the accuracy achievable with the integration architectures developed and the navigation sensors selected, tests were carried out at a composting site. The data acquisition of the tests was covered in Chapter 8. The prototype of the automatically steered compost turner was equipped with an aluminium profile carrying the navigation sensors. Additionally, two 360°-prisms were mounted on the aluminium profile and tracked by two robotic total stations to generate a reference trajectory. Data from the navigation sensors were recorded into Robot Operating System (ROS) bag files so that they could later be replayed to test different filter architectures with the same data. In total, two tests were carried out with the prototype vehicle at a composting site. For the first tests, the compost turner was steered manually and data were recorded without the compost turner turning the windrows since the spiked drum was not working at that time. For the final tests, the compost turner was steered automatically by the automatic steering system developed and turned the windrows with its spiked drum.

In Chapter 9, the results were presented. The positioning performance was evaluated by comparing the estimated positions and headings of the navigation filters to the positions and headings of the reference trajectory.

Data from the first tests were used to evaluate the two different filter architectures presented in Chapter 7. The federated filter architecture allowed analysing the results of the local block filters as well as the overall solution. For the first local filter fusing GNSS and the IMU, the 3D position error was less than 3 cm and the heading deviation from the reference was less than 1°. The second local filter, which fuses NDT and the tailored odometry model, yielded position errors of up to 50 cm and heading errors of up to 5°. This large positioning error was mainly attributed to NDT, as the algorithm had difficulties registering the point cloud observed by the stereo camera to the homogeneous point cloud of the composting site. As the odometry model can only compute position changes from one epoch to the next, it was also affected by drift. The combination of NDT and odometry therefore deteriorated with time. The overall result of the federated

filter is the result of the snapshot fusion algorithm, which fuses information from both local filters. In principle, the snapshot fusion is a weighted least-squares adjustment of the solutions of the two local filters. The resulting position had a horizontal accuracy of less than 10 cm and a heading accuracy of less than 1°. However, the final trajectory was affected by the drift of the second local filter. As the snapshot fusion only fused two observation groups, it was unable to tell which one was erroneous. The cascaded filter architecture fuses information from all sensors in one Kalman filter and can therefore also be used to detect and eliminate outliers. The cascaded filter architecture was therefore not affected by drift. It achieved a 2D and 3D accuracy of less than 10 cm and a maximum heading deviation of 1.34° from the reference trajectory with the data from the first tests. The cascaded filter is therefore better suited for the automatic steering system.

For the final tests and the demonstration of the automatic steering system, a cascaded filter was used which fused observations from GNSS, the IMU, and rotary encoders with the tailored odometry model. During the demonstration of the automatic steering system, the cascaded filter provided the position and attitude of the machine in real-time, which were used to compute manoeuvre instructions to steer the machine along an optimal route. The speed of the compost turner was lower than in the first tests and the vibrations were higher. The mean horizontal positioning error was 6.0 cm for the first and 6.9 cm for the second windrow. The maximum horizontal positioning errors were 16.1 cm for the first and 22.8 cm for the second windrow. Even though a lower accuracy than initially expected was achieved for the final tests, the accuracy was still sufficient to steer the compost turner automatically through the windrows.

NDT was added to the cascaded filter in post-processing. Since the registration accuracy of NDT was low (with a mean error of 49.1 cm for the first and 45.2 cm for the second windrow), only minor improvements were made to the overall positioning accuracy. The cascaded filter fusing information from all sensors had a mean horizontal positioning error of 4.3 cm for the first and 6.1 cm for the second windrow.

What are the advantages of a multi-sensor approach?

Fusing information from multiple sensors to estimate the state vector of the compost turner has several advantages. First, not all components of the state vector can be

estimated using a single sensor. For example, when only a dual-antenna GNSS receiver is used, the position and speed of the vehicle can be computed, but not the full attitude. The full attitude can be determined by adding observations from an IMU. Note that the full attitude could also be determined by adding a third GNSS antenna that is non-collinear with the other two. However, such a setup does not have the advantage of complementary redundancy.

When complementary sensors are fused, the advantages of one sensor compensate for the disadvantages of the other. When a position fixing system, which typically has a high long-term accuracy but a low short-term accuracy, is fused with a dead reckoning system, which typically has a high short-term but a low long-term accuracy, the resulting integrated navigation solution has both a high short-term and a high long-term accuracy.

Moreover, information from position fixing sensors can be used to estimate sensor errors of dead reckoning sensors, such as inertial sensor errors or a scale factor for odometry. This can be done by adding the sensor errors to the state vector and adapting the filter equations accordingly, as presented in Chapter 7 of this thesis.

When redundant information is present to estimate the state vector of a moving object and a suitable navigation filter architecture is chosen, outliers of individual sensors can be detected and eliminated. This was shown in Chapter 9, where the cascaded filter architecture allowed eliminating outliers.

Finally, information from redundant sensors can be used to bridge outages or sensor failures of other sensors. This was shown in Chapter 9, where GNSS outages of up to 30 seconds were bridged with the tailored odometry model. After 30 seconds without GNSS, the horizontal positioning error was still below 10 cm. This result cannot be achieved when MEMS IMUs are used to bridge GNSS outages.

10.2 Outlook

This thesis presented the development of a multi-sensor positioning system used in an automatic steering system for tracked compost turners. Since windrow composting is a

monotonous task and windrows need to be turned regularly, it is a great benefit when compost turners are steered automatically. However, it would be an even greater benefit if windrow composting could be further optimized and the compost turner could operate autonomously.

The automatic steering system developed has several limitations as it still requires a worker to steer the compost turner to the beginning of the windrow. The compost turner can then turn one windrow automatically, but it will stop at the end of the windrow. A worker then has to steer the compost turner to the beginning of the next windrow that has to be turned. Moreover, the automatic steering system requires a 3D point cloud of the composting site as an input to compute the optimal routes. In reality, however, the environment of the composting site might change, because finished windrows will be transported away and new windrows will be piled up.

An ideal system for windrow composting would not only consist of an autonomous compost turner, but would also include a composting site management database and a smart logistics concept. Ideally, the autonomous compost turner would be able to detect which windrow needs to be turned next, drive there on its own and then turn the windrow automatically. It might even measure temperature, CO2, and methane content as it turns the windrows. It might send these georeferenced data to a composting site management database. While driving around at the composting site, it might automatically map the composting site and update the 3D point cloud. When the compost turner needs recharging, it could automatically drive to the charging station. Some of these aspects are investigated in the follow-up project "ANDREA" (Automated GNSS-based data and process management for composting plants), which is funded by the Austrian Research Promotion Agency (FFG).

Creating this ideal system leads to several challenges for navigation. One aspect is that we have to go from automatic steering to autonomous driving. While the compost turner had clearly defined and straight paths through the windrows to follow for the automatic steering system, it will have to compute and follow more complex paths at the whole composting site for autonomous driving. To do so, a complex route planning module is required. Moreover, the compost turner has to be able to detect and avoid obstacles. To map the environment, a more accurate image-based system is needed. An active

sensor such as LIDAR could be used, which also allows obstacle detection and mapping at night.

Another limitation of the automatic steering system developed for compost turners is that it is designed to work in outdoor environments. However, there are also indoor composting plants. For the automatic steering system to work there, GNSS would have to be replaced by an indoor position fixing sensor. Investigations with UWB or WiFi-RTT could be carried out in the future to determine whether the automatic steering of compost turners is also possible in indoor environments.

Bibliography

Ali, Abdelrahman, Billy Chan, Medhat Omr, and Jacques Georgy (2019). "Coursa Drive: Integrated Navigation Solution for Autonomous Vehicles". In: *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*. ION GNSS+, The International Technical Meeting of the Satellite Division of The Institute of Navigation. Institute of Navigation, pp. 1220–1233. DOI: 10.33012/2019.16870.

Allerton, David J. and Huamin Jia (2005). "A Review of Multisensor Fusion Methodologies for Aircraft Navigation Systems". In: *Journal of Navigation* 58.3, pp. 405–417. ISSN: 0373-4633. DOI: 10.1017/S0373463305003383.

American Society of Agricultural and Biological Engineers (2011). *Agricultural machinery management data ASAE D497. 7 MAR2011 (R2015)*. St. Joseph, MI, USA.

Bancroft, Jared B. (2010). "Multiple Inertial Measurement Unit Integration for Pedestrian Navigation". PhD thesis. Calgary, Alberta: University of Calgary.

Bancroft, Jared B. and Gérard Lachapelle (2011). "Data fusion algorithms for multiple inertial measurement units". In: *Sensors (Basel, Switzerland)* 11.7, pp. 6771–6798. DOI: 10.3390/s110706771.

Bensky, Alan (2016). *Wireless positioning technologies and applications*. Second edition. GNSS technology and applications series. Boston and London: Artech House. ISBN: 978-1-60807-951-3.

Biber, Peter and Wolfgang Strasser (2003). "The normal distributions transform: a new approach to laser scan matching". In: *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, 2743–2748 vol.3. DOI: 10.1109/IROS.2003.1249285.

Briot, Sébastien and Wisama Khalil, eds. (2015). *Dynamics of Parallel Robots*. Mechanisms and Machine Science. Cham: Springer International Publishing. ISBN: 978-3-319-19787-6. DOI: 10.1007/978-3-319-19788-3.

Brown, Robert Grover and Patrick Y. C. Hwang (2012). *Introduction to random signals and applied Kalman filtering: With MATLAB exercises*. 4. ed. Hoboken, NJ: Wiley. ISBN: 978-0-470-60969-9.

Bucy, Richard S. and Peter D. Joseph (2005). *Filtering for Stochastic Processes with Applications to Guidance*. Vol. v.326. AMS Chelsea Publishing Ser. Providence: American Mathematical Society. ISBN: 978-0821837825.

Cambridge Dictionary (2022). *Navigation*. Ed. by Cambridge University Press. URL: https://dictionary.cambridge.org/dictionary/english/navigation (visited on 04/01/2022).

Carlson, Neal A. (1990). "Federated square root filter for decentralized parallel processors". In: *IEEE Transactions on Aerospace and Electronic Systems* 26.3, pp. 517–525. ISSN: 00189251. DOI: 10.1109/7.106130.

Carlson, Neal. A. and Michael P. Berarducci (1994). "Federated Kalman Filter Simulation Results". In: *Navigation* 41.3, pp. 297–322. ISSN: 00281522. DOI: 10.1002/j.2161-4296.1994.tb01882.x.

Collins, Danielle (2015). *What are capacitive encoders and where are they suitable?* URL: https://www.motioncontroltips.com/faq-what-are-capacitive-encoders-and-where-are-they-suitable/ (visited on 07/20/2022).

Destino, Giuseppe, Jani Saloranta, Gonzalo Seco-Granados, and Henk Wymeersch (2018). "Performance Analysis of Hybrid 5G-GNSS Localization". In: *2018 52nd Asilomar Conference on Signals, Systems, and Computers*. IEEE, pp. 8–12. ISBN: 978-1-5386-9218-9. DOI: 10.1109/ACSSC.2018.8645207.

Endo, Daisuke, Yoshito Okada, Keiji Nagatani, and Kazuya Yoshida (2007). "Path following control for tracked vehicles based on slip-compensating odometry". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007*. Piscataway, NJ: IEEE Service Center, pp. 2871–2876. ISBN: 978-1-4244-0911-2. DOI: 10.1109/IROS.2007.4399228.

European Commission (2010). *Communication from the Commission to the Council and the European Parliament on future steps in bio-waste management in the European*

Union. Ed. by European Commission. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52010DC0235 (visited on 03/28/2022).

European Union Agency for the Space Programme (2022). *EUSPA EO and GNSS Market Report*. Ed. by Publications Office of the European Union. Luxembourg. DOI: 10.2878/94903.

Falcone, Marco, Jörg Hahn, and Thomas Burger (2017). "Galileo". In: *Springer Handbook of Global Navigation Satellite Systems*. Ed. by Peter J.G. Teunissen and Oliver Montenbruck. Cham: Springer International Publishing, pp. 247–272. ISBN: 978-3-319-42926-7. DOI: 10.1007/978-3-319-42928-1_9.

Farrell, Jay A. and Jan Wendel (2017). "GNSS/INS Integration". In: *Springer Handbook of Global Navigation Satellite Systems*. Ed. by Peter J.G. Teunissen and Oliver Montenbruck. Cham: Springer International Publishing, pp. 811–840. ISBN: 978-3-319-42926-7. DOI: 10.1007/978-3-319-42928-1_28.

Giorgi, Gabriele (2017). "Attitude Determination". In: *Springer Handbook of Global Navigation Satellite Systems*. Ed. by Peter J.G. Teunissen and Oliver Montenbruck. Cham: Springer International Publishing, pp. 781–809. ISBN: 978-3-319-42926-7. DOI: 10.1007/978-3-319-42928-1_27.

GPS World (5.03.2020). "GNSS key to farm of the future". In: *GPS World* 2020. URL: https://www.gpsworld.com/gnss-key-to-farm-of-the-future/ (visited on 03/29/2022).

Grewal, Mohinder S., Lawrence Randolph Weill, and Angus P. Andrews (2007). *Global positioning systems, inertial navigation and integration*. 2. ed. Hoboken, NJ: Wiley. ISBN: 978-0-470-09971-1. DOI: 10.1002/0470099720.

Grisso, Robert, Zane R. Helsel, and Vern Grubinger (2019). *Selecting Engine and Travel Speeds for Optimal Fuel Efficiency*. Ed. by Farm-Energy. URL: https://farm-energy.extension.org/selecting-engine-and-travel-speeds-for-optimal-fuel-efficiency/ (visited on 06/21/2021).

Groves, Paul D. (2013). *Principles of GNSS, inertial, and multisensor integrated navigation systems*. Second edition. Artech House GNSS library. Boston and London: Artech House. ISBN: 9781608070050.

Han, Shufeng, He Yong, and Fang Hui (2018). "Recent development in automatic guidance and autonomous vehicle for agriculture: A Review". In: *Journal of Zhejiang*

University (Agriculture and Life Sciences) 44.4, p. 381. DOI: 10.3785/j.issn.1008-9209.2018.07.231.

Han, X. Z., H. J. Kim, C. W. Jeon, H. C. Moon, and J. H. Kim (2017). "Development of a low-cost GPS/INS integrated system for tractor automatic navigation". In: *International Journal of Agricultural and Biological Enginnering*. Vol. 10 (2), pp. 123–131. DOI: 10.3965/j.ijabe.20171002.3070.

Haug, Anton J. (2012). *Bayesian estimation and tracking: A practical guide*. Hoboken, N.J: Wiley. ISBN: 978-0-470-62170-7.

Hauschild, André (2017). "Basic Observation Equations". In: *Springer Handbook of Global Navigation Satellite Systems*. Ed. by Peter J.G. Teunissen and Oliver Montenbruck. Cham: Springer International Publishing, pp. 561–582. ISBN: 978-3-319-42926-7. DOI: 10.1007/978-3-319-42928-1_19.

Hexagon Agriculture (2022). *HxGN AgrOn Auto Steering*. URL: https://hexagonagriculture.com/solutions/machine-control/auto-steering (visited on 03/30/2022).

Hirt, Martin (2021). *Kostenfreier RTK-Korrekturdatendienst: Hier geht's zu allen Infos und zur Registrierung*. URL: https://www.lko.at/kostenfreier-rtk-korrekturdatendienst-hier-geht-s-zu-allen-infos-und-zur-registrierung+2400+3309904 (visited on 10/22/2021).

Hofmann-Wellenhof, Bernhard, Klaus Legat, and Manfred Wieser (2003). *Navigation: Principles of Positioning and Guidance*. Springer eBook Collection. Vienna and s.l.: Springer Vienna. ISBN: 9783709160787. DOI: 10.1007/978-3-7091-6078-7.

Hofmann-Wellenhof, Bernhard, Herbert Lichtenegger, and Elmar Wasle (2008). *GNSS: Global Navigation Satellite Systems - GPS, Glonass, Galileo, and More*. Wien and NewYork: Springer. ISBN: 978-3-211-73017-1.

Hol, Jeroen D., Thomas B. Schon, and Fredrik Gustafsson (2006). "On Resampling Algorithms for Particle Filters". In: *2006 IEEE Nonlinear Statistical Signal Processing Workshop*. IEEE, pp. 79–82. ISBN: 978-1-4244-0579-4. DOI: 10.1109/NSSPW.2006.4378824.

Jekeli, Christopher (2001). *Inertial navigation systems with geodetic applications*. Berlin: de Gruyter. ISBN: 3-11-015903-1.

Julier, Simon J. and Jeffrey K. Uhlmann (1997). "New extension of the Kalman filter to nonlinear systems". In: *Signal Processing, Sensor Fusion, and Target Recognition VI*. Ed. by Ivan Kadar. SPIE Proceedings. SPIE, p. 182. DOI: 10.1117/12.280797.

Kalman, Rudolf E. (1960). "A New Approach to Linear Filtering and Prediction Problems". In: *Journal of Basic Engineering* 82.1, pp. 35–45. ISSN: 0021-9223. DOI: 10.1115/1.3662552.

Kaplan, Elliott D. and Christopher J. Hegarty, eds. (2006). *Understanding GPS: Principles and applications*. 2nd. ed. Artech House mobile communications series. Boston, Mass.: Artech House. ISBN: 978-1580538947.

Kia, Ghazaleh, Jukka Talvitie, and Laura Ruotsalainen (2021). "RSS-Based Fusion of UWB and WiFi-Based Ranging for Indoor Positioning". In: *WiP Proceedings of the Eleventh International Conference on Indoor Positioning and Indoor Navigation - Work-in-Progress Papers (IPIN-WiP 2021)*. Ed. by Antoni Pérez-Navarro, Raúl Montoliu, and Joaquín Torres-Sospedra.

Kuhlman, Lester R. (1990). "Windrow composting of agricultural and municipal wastes". In: *Resources, Conservation and Recycling* 4.1-2, pp. 151–160. ISSN: 09213449. DOI: 10.1016/0921-3449(90)90039-7.

Lange, Arthur F. and John Peake (2021). "Precision Agriculture". In: *Position, Navigation, and Timing Technologies in the 21st Century*. Ed. by Y. T. Jade Morton, Frank Diggelen, James J. Spilker, Bradford W. Parkinson, Sherman Lo, and Grace Gao. Wiley, pp. 1735–1747. ISBN: 9781119458494. DOI: 10.1002/9781119458555.ch56.

Langley, Richard B., Peter J.G. Teunissen, and Oliver Montenbruck (2017). "Introduction to GNSS". In: *Springer Handbook of Global Navigation Satellite Systems*. Ed. by Peter J.G. Teunissen and Oliver Montenbruck. Cham: Springer International Publishing, pp. 3–23. ISBN: 978-3-319-42926-7. DOI: 10.1007/978-3-319-42928-1_1.

Leica Geosystems AG (2006). *Leica TPS1200+ Technical specifications and system features*. URL: https://totalstations.co/brochure/Leica-TCRP-1201-Edited-1.pdf (visited on 05/28/2022).

– (2018). *Leica RTC360 3D Reality Capture Solution*. URL: https://leica-geosystems.com/en-gb/products/laser-scanners/scanners/leica-rtc360 (visited on 07/01/2022).

– (2020). *Leica Nova MS 60 Data sheet*. URL: https://w3.leica-geosystems.com/downloads123/zz/tps/nova_ms60/brochures-datasheet/Leica%20Nova%20MS60%20DS_en.pdf (visited on 05/21/2022).

Lercher, Thomas (2020). "Automatisierte Routenberechnung für autonom fahrende landwirtschaftliche Maschinen". Bachelor's Thesis. Graz University of Technology.

Magnusson, Martin (2009). *The Three-Dimensional Normal-Distributions Transform: An Efficient Representation for Registration, Surface Analysis, and Loop Detection*. Vol. 36. Örebro Studies in Technology. Örebro: Örebro universitet. ISBN: 978-91-7668-696-6.

Menegaz, Henrique M. T., Joao Y. Ishihara, Geovany A. Borges, and Alessandro N. Vargas (2015). "A Systematization of the Unscented Kalman Filter Theory". In: *IEEE Transactions on Automatic Control* 60.10, pp. 2583–2598. ISSN: 0018-9286. DOI: 10.1109/TAC.2015.2404511.

Nagatani, Keiji, Daisuke Endo, and Kazuya Yoshida (2007). "Improvement of the Odometry Accuracy of a Crawler Vehicle with Consideration of Slippage". In: *IEEE International Conference on Robotics and Automation, 2007*. Piscataway, NJ: IEEE Service Center, pp. 2752–2757. ISBN: 1-4244-0602-1. DOI: 10.1109/ROBOT.2007.363881.

Noureldin, Aboelmagd, Tashfeen B. Karamat, and Jacques Georgy (2013). *Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration*. Berlin and Heidelberg: Springer. ISBN: 978-3-642-30465-1. DOI: 10.1007/978-3-642-30466-8.

Odijk, Dennis and Lambert Wanninger (2017). "Differential Positioning". In: *Springer Handbook of Global Navigation Satellite Systems*. Ed. by Peter J.G. Teunissen and Oliver Montenbruck. Cham: Springer International Publishing, pp. 753–780. ISBN: 978-3-319-42926-7. DOI: 10.1007/978-3-319-42928-1_26.

Perez-Ruiz, Manuel, Jorge Martínez-Guanter, and Shrini K. Upadhyaya (2021). "High-precision GNSS for agricultural operations". In: *GPS and GNSS Technology in Geosciences*. Elsevier, pp. 299–335. ISBN: 9780128186176. DOI: 10.1016/B978-0-12-818617-6.00017-2.

Petovello, Mark (2011). "GNSS Solutions: Absolutely Relative Positioning: How do you compute relative positions with GNSS?" In: *InsideGNSS* May/June 2011, pp. 32–36. URL: https://insidegnss-com.exactdn.com/wp-content/uploads/2018/01/mayjune11-Solutions.pdf (visited on 04/25/2022).

Razza, Francesco, Lorenzo D'Avino, Giovanni L'Abate, and Luca Lazzeri (2018). "The Role of Compost in Bio-waste Management and Circular Economy". In: *Designing Sustainable Technologies, Products and Policies*. Ed. by Enrico Benetto, Kilian Gericke, and Mélanie Guiton. Cham: Springer International Publishing, pp. 133–143. ISBN: 978-3-319-66980-9. DOI: 10.1007/978-3-319-66981-6_16.

Reitbauer, Eva (2017). "Radio Fingerprinting Optimization Tailored to Vehicles in Parking Garages". Master's Thesis. Graz: Graz University of Technology. URL: https://diglib.tugraz.at/radio-fingerprinting-optimization-tailored-to-vehicles-in-parking-garages-2017 (visited on 07/20/2022).

Reitbauer, Eva and Christoph Schmied (2021a). "Bridging GNSS Outages with IMU and Odometry: A Case Study for Agricultural Vehicles". In: *Sensors (Basel, Switzerland)* 21.13. DOI: 10.3390/s21134467.

– (2021b). "Performance Analysis of GNSS/INS/VO/Odometry Sensor Fusion Algorithms for Tracked Agricultural Vehicles". In: *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*. ION GNSS+, The International Technical Meeting of the Satellite Division of The Institute of Navigation. Institute of Navigation, pp. 3250–3262. DOI: 10.33012/2021.18053.

Reitbauer, Eva, Christoph Schmied, and Manfred Wieser (2020a). "Autonomous Navigation Module for Tracked Compost Turners". In: *2020 European Navigation Conference (ENC)*. IEEE, pp. 1–10. ISBN: 978-3-944976-27-3. DOI: 10.23919/ENC48637.2020.9317465.

Reitbauer, Eva, Christoph Schmied, and Michael Schedler (2020b). "Integrated Navigation for Tracked Compost Turners Using GNSS, INS, Odometers, Stereo Camera and 3D Map". In: *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*. ION GNSS+, The International Technical Meeting of the Satellite Division of The Institute of Navigation. Institute of Navigation, pp. 2518–2532. DOI: 10.33012/2020.17700.

Rovira-Más, Francisco, Ishani Chatterjee, and Verónica Sáiz-Rubio (2015). "The role of GNSS in the navigation strategies of cost-effective agricultural robots". In: *Computers and Electronics in Agriculture* 112, pp. 172–183. ISSN: 01681699. DOI: 10.1016/j.compag.2014.12.017.

Sanz Subirana, Jaume, José Miguel Juan Zorzona, and Manuel Hernandez-Pajares (2013). *GNSS Data Processing, Vol. I: Fundamentals and Algorithms*. Noordwijk: ESA Communications. ISBN: 978-92-9221-886-7.

Schmied, Christoph (2020). "Determination of Relative Orientation based on Distance Measurements". Master's Thesis. Graz: Graz University of Technology. URL: https:

//diglib.tugraz.at/determination-of-relative-orientation-based-on-distance-measurements-2020 (visited on 05/31/2022).

Siebert, Stefanie (2017). *Bio-Waste Recycling in Europe Against the Backdrop of the Circular Economy Package: 161024 ECN Biowaste Recycling in Europe*. Ed. by European Compost Network. URL: https://www.compostnetwork.info/download/bio-waste-recycling-europe-backdrop-circulareconomy- (visited on 03/28/2022).

Simon, Dan (2006). *Optimal State Estimation*. Hoboken, NJ, USA: John Wiley & Sons, Inc. ISBN: 9780470045343. DOI: 10.1002/0470045345.

Skaloud, Jan and Klaus Legat (2006). *Navigation Techniques: Lecture Notes Summer Term 2006*. EPFL Lausanne.

Smoot, Jeff (2022). *Capacitive, Magnetic, and Optical Encoders – Comparing the Technologies*. URL: https://www.cuidevices.com/blog/capacitive-magnetic-and-optical-encoders-comparing-the-technologies (visited on 07/20/2022).

Stanisak, Mirko, Karen H.M. von Hünerbein, Ulf Bestmann, and Werner Lange (2016). "Measured GNSS Jamming Events at German Motorways". In: *Proceedings of POSNAV ITS*. Ed. by Deutsche Gesellschaft für Ortung und Navigation (DGON) e.V.

Sweeten, John M. (2008). *Composting Manure and Sludge*. URL: http://hdl.handle.net/1969.1/87650 (visited on 12/12/2019).

Teunissen, Peter J. G. (1995). "The least-squares ambiguity decorrelation adjustment: a method for fast GPS integer ambiguity estimation". In: *Journal of Geodesy* 70.1-2, pp. 65–82. ISSN: 0949-7714. DOI: 10.1007/BF00863419.

Teunissen, Peter J.G. and Oliver Montenbruck, eds. (2017). *Springer Handbook of Global Navigation Satellite Systems*. Cham: Springer International Publishing. ISBN: 978-3-319-42926-7. DOI: 10.1007/978-3-319-42928-1.

Theurl, Fabian (2022). "Pose Estimation by Comparing Stereo Camera Data with Synthetic Depth Images Generated from Preexisting Point Clouds". Master's Thesis. Graz: Graz University of Technology.

Thrun, Sebastian, Wolfram Burgard, and Dieter Fox (2006). *Probabilistic robotics*. Intelligent robotics and autonomous agents. Cambridge, Mass. and London: MIT Press. ISBN: 978-0262201629.

Trimble Inc. (2022). *Autopilot™ Automated Steering System*. URL: https://agriculture.trimble.com/product/autopilot-automated-steering-system/ (visited on 03/30/2022).

van der Linden, Ann and Almut Reichel (2020). *Bio-waste in Europe — turning challenges into opportunities*. Ed. by Publications Office of the European Union. DOI: 10.2800/630938. (Visited on 03/28/2022).

Wan, Eric A. and Rudolph van der Merwe (2000). "The unscented Kalman filter for nonlinear estimation". In: *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*. IEEE, pp. 153–158. ISBN: 0-7803-5800-7. DOI: 10.1109/ASSPCC.2000.882463.

Ward, Nick (2009). *"Integrated Navigation System": International Dictionary of Marine Aids to Navigation*. URL: https://www.iala-aism.org/wiki/dictionary/index.php/Integrated_navigation_system (visited on 04/05/2022).

Wendel, Jan (2011). *Integrierte Navigationssysteme: Sensordatenfusion, GPS und inertiale Navigation*. 2., überarbeitete Auflage. München: Oldenbourg. ISBN: 978-3-486-70439-6. DOI: 10.1524/9783486705720.

Wertz, James R., ed. (1978). *Spacecraft Attitude Determination and Control*. Vol. 73. Astrophysics and Space Science Library. Dordrecht: Springer. ISBN: 978-94-009-9907-7. DOI: 10.1007/978-94-009-9907-7.

Yamauchi, Genki, Daiki Suzuki, and Keiji Nagatani (2016). "Online slip parameter estimation for tracked vehicle odometry on loose slope". In: *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, pp. 227–232. ISBN: 978-1-5090-4349-1. DOI: 10.1109/SSRR.2016.7784303.

Yamauchi, Genki, Keiji Nagatani, Takeshi Hashimoto, and Kenichi Fujino (2017). "Slip-compensated odometry for tracked vehicle on loose and weak slope". In: *ROBOMECH Journal* 4.1. DOI: 10.1186/s40648-017-0095-1.

Yousif, Khalid, Alireza Bab-Hadiashar, and Reza Hoseinnezhad (2015). "An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics". In: *Intelligent Industrial Systems* 1.4, pp. 289–311. ISSN: 2363-6912. DOI: 10.1007/s40903-015-0032-7.

Zhang, Quan, Qijin Chen, Zhengpeng Xu, Tisheng Zhang, and Xiaoji Niu (2021). "Evaluating the navigation performance of multi-information integration based on low-end inertial sensors for precision agriculture". In: *Precision Agriculture* 22.3, pp. 627–646. ISSN: 1385-2256. DOI: 10.1007/s11119-020-09747-x.