CZECH TECHNICAL
UNIVERSITY
IN PRAGUE

F3

Faculty of Electrical Engineering
Department of Control Engineering

Master's Thesis

Indoor localization system for automated vehicles based on Ultra-Wideband technology

Bc. Jitka Hodná
Cybernetics and robotics

May 2021

Datavision s. r. o.
Supervisor: Ing. Tomáš Novák

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Hodná Jitka** Personal ID number: **439565**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Control Engineering**

Study program: **Cybernetics and Robotics**

Branch of study: **Cybernetics and Robotics**

II. Master’s thesis details

Master’s thesis title in English:

Indoor localization system for automated vehicles based on Ultra-Wideband technology

Master’s thesis title in Czech:

Interiérový lokalizační systém pro autonomní prostředky s využitím technologie Ultra-Wideband

Guidelines:

1. Study the state of the art data fusion principles used for pose estimation. Study principles of Inertial navigation systems (INS)
2. Propose a localization system for autonomous vehicles based on fusion of data from Ultra-Wideband (UWB) positioning system and on-board dead-reckoning sensors such as Inertial measurement unit (IMU)
3. Evaluate proposed localization system for use in industrial environments.

Bibliography / sources:

[1] THRUN, SEBASTIAN, WOLFRAM BURGARD, AND DIETER FOX - PROBABILISTIC ROBOTICS, 2005 - Massachusetts Institute of Technology, USA (2005)
[2] GREWAL, MOHINDER S., ANGUS P. ANDREWS, AND CHRIS G. BARTONE - GLOBAL NAVIGATION SATELLITE SYSTEMS, INERTIAL NAVIGATION, AND INTEGRATION - John Wiley & Sons, 2020
[3] KELLY, ALONZO - MOBILE ROBOTICS: MATHEMATICS, MODELS, AND METHODS - Cambridge University Press, 2013
[4] MOORE, THOMAS, AND DANIEL STOUCH - A GENERALIZED EXTENDED KALMAN FILTER IMPLEMENTATION FOR THE ROBOT OPERATING SYSTEM, Intelligent autonomous systems 13. Springer, Cham, 2016. 335-348
[5] HOL, JEROEN D., et al. - TIGHTLY COUPLED UWB/IMU POSE ESTIMATION, 2009 IEEE international conference on ultra-wideband. IEEE, 2009
[6] LI, JIAXIN, ET AL. - ACCURATE 3D LOCALIZATION FOR MAV SWARMS BY UWB AND IMU FUSION, 2018 IEEE 14th International Conference on Control and Automation (ICCA). IEEE, 2018.

Name and workplace of master’s thesis supervisor:

Ing. Tomáš Novák, DataVision s.r.o., Ukrajinská 2a, Praha 10

Name and workplace of second master’s thesis supervisor or consultant:

Ing. Martin Hlinovský, Ph.D., Department of Control Engineering, FEE

Date of master’s thesis assignment: **15.01.2021** Deadline for master's thesis submission: __________

Assignment valid until:
by the end of summer semester 2021/2022

Ing. Tomáš Novák Supervisor’s signature	prof. Ing. Michael Šebek, DrSc. Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

© ČVUT v Praze, Design: ČVUT v Praze, VIC

Acknowledgement / Declaration

Lorem ipsum sit amet, thanks to TACR (TACR project with the correct name and number)

I hereby declare that I wrote the presented thesis on my own and that I cited all the used information sources in compliance with the Methodical instructions about the ethical principles for writing an academic thesis.

....

Prague, May 21, 2021

Abstrakt / Abstract

Lorem ipsum sit amet

Klíčová slova: ultra-wideband, imu

Překlad titulu: Interiérový lokalizační systém pro autonomní prostředky s využitím technologie Ultra-Wideband

The most awesome abstract

Keywords: ultra-wideband, imu

Contents /

1 Introduction 1
1.1 Section 1 1
2 Indoor localization methods 2
2.1 Section 1 2
3 Sensors 3
3.1 Localization based on Ultra-wideband 3
3.2 Inertial measurement unit 3
  3.2.1 Accelerometers 4
  3.2.2 Gyroscopes 4
  3.2.3 IMU's errors and Allan variance analysis 5
  3.2.4 Performance of IMUs according to their aplication 7
  3.2.5 Navigation (mechanization) equations 7
3.3 Odometry 7
4 State estimations algorithms for localization 10
4.1 Kalman and particle filter . . . 10
4.2 Algorithms based on Kalman filter 10
4.3 Error state Extended Kalman filter 11
4.4 Chosen approach for state estimation 12
5 Localization system design and implementation 14
5.1 System architecture design . . . 14
5.2 System kinematics 15
  5.2.1 Representation of 3D attitude and rotation in space 15
  5.2.2 The kinematics equations in continuous time . . 16
  5.2.3 The kinematics equations in discrete time . . . 18
5.3 Error state extended Kalman filter implementation . 18
5.4 Injection the error state into the navigation state 20
5.5 Implementation tools 21
6 Experiments 22
6.1 Section 1 22
7 Conclusion and future work 23
7.1 Section 1 23
7.2 Section 2 future work 23
References 24
A Abbreviations and symbols 27
A.1 A list of abbreviations 27
A.2 A list of symbols 27

Chapter 1
Introduction

Lorem ipsum sit amet

1.1 Section 1

Lorem ipsum sit amet

Chapter 2

Indoor localization methods

2.1 Section 1

Lorem ipsum sit amet

Chapter 3
Sensors

In chapter 3 the overview of used sensors and their properties is given. The main aim is to describe an localization method based on ultra-wideband technology(UWB), an inertial measurement unit (IMU) and an odometry. The UWB TODO. The IMU section mainly focuses on the unit overview, a short description of gyroscopes and accelerometers used in it, the errors of these sensors, its analysis and outcomes for the localization unit. The odometry TODO.

3.1 Localization based on Ultra-wideband

3.2 Inertial measurement unit

An inertial measurement unit (IMU) is a device that utilizes measurement systems such as gyroscopes and accelerometers to estimate the relative position, velocity and acceleration of a vehicle in motion[1]. The unit is typically integrated with an on-board computational unit and may contain more sensors as a magnetometer or thermometer.

The gyroscopes measure angular velocities and accelerometers specific forces, which can be easily transformed into linear accelerations [1]. The IMU typically contains three orthogonal accelerometers and three orthogonal gyroscopes. Because of that, it can measure angular velocities and specific forces in each axis to maintain a 6-DOF estimate of the pose of the vehicle (position (x, y, z) and orientation $(roll, pitch, yaw)$. The process of the computation can be seen in Figure 3.1.

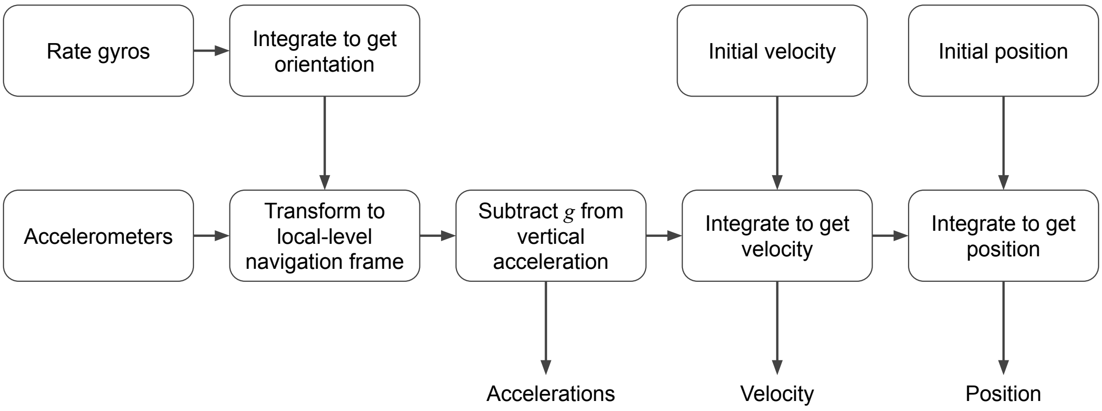
Rate gyros
Integrate to get orientation
Initial velocity
Initial position
Accelerometers
Transform to local-level navigation frame
Subtract g from vertical acceleration
Integrate to get velocity
Integrate to get position
Accelerations
Velocity
Position

Figure 3.1. IMU block diagram [1]

There are two basic ways how to mount the IMU to a vehicle, also called mechanization architectures [1–2].

- In **gimbaled systems**, the IMU is attached to a stabilized platform that maintains its inertial orientation as the vehicle manoeuvres.
- In **strap-down systems**, it is rigidly attached to the vehicle.

The mechanization determines the conversion between measurements of IMU and estimation of linear accelerations and angular velocities of the vehicle. It means the transformations IMU body frame to local frame. The conversion is called navigation (mechanization) equations and they are briefly summarized in Section 3.2.5.

IMU's are extremely sensitive to measurement errors given by properties of used gyroscopes, accelerometers and their mounting. As the data are once or twice integrated, any error in measurement causes a linear or quadratic error in the pose estimation. Even with a small measurement error, the IMU's drift becomes significant, and it needs to be externally compensated. The IMU provides a short-term stable solution, which is not affected by external environment [2], and it has a high data rate (100 Hz - 200 Hz). That makes the IMU measurement complementary to the UWB localization measurement.

3.2.1 Accelerometers

Accelerometers can measure external forces acting on the vehicle. They measure a specific force relatively to a non-rotating inertial space in a specific direction. They are sensitive to all forces, including gravity and fictitious forces [1].

Mechanical accelerometers use a spring-mass-damper system. The force acts on the mass, and it causes displacement of the spring. The system is limited by physical properies of real spring.

Microelectromechanical systems (MEMS) based accelerometers are made of at least three components, namely a proof mass, a suspension to hold the mass and a pickoff, which relates an output signal to the induced accelerations [3]. MEMS accelerometers are then classified by the type of converting the mechanical displacement of the proof mass to an electrical signal. In most common principles belong to piezoresistive, capacitive sensing, piezoelectric, optical sensing and tunnelling current sensing. The piezoelectric MEMS sensors can not be used for navigation because their output rate is too low [3].

The current accelerometers used technology according to an application is sumarized in Figure 3.2.

3.2.2 Gyroscopes

Gyroscopes are used for estimating a rotational motion of a body, each gyroscope measures angular rate ω (inertial angular rotation) relatively to a non-rotating inertial space in one axis. There are basically three main categories of gyroscopes [1].

Mechanical gyroscopes have a mass spinning steadily with respect to a free movable axis, they are not used a lot anymore, but they can be found in very old submarines.

Optical gyroscopes are based on the Sagnac effect, which states that frequency/phase shift between two waves counter-propagating in a rotating ring interferometer is proportional to the loop angular velocity. As a light source, laser is typically used. Currently, this technology gives the best performance. Examples can be ring laser gyroscopes (RLG) or fibre optic gyroscopes (FOG).

Vibrating gyroscopes are based on the Coriolis effect that induces a coupling between two resonant modes of a mechanical resonator.

MEMS gyroscopes [3] play significant role in robotics, because of their simplicity. They are small, cheap, have no rotating parts and furthermore have low power consumption.

The performace and application of each technology is demonstrated in Figure 3.3.

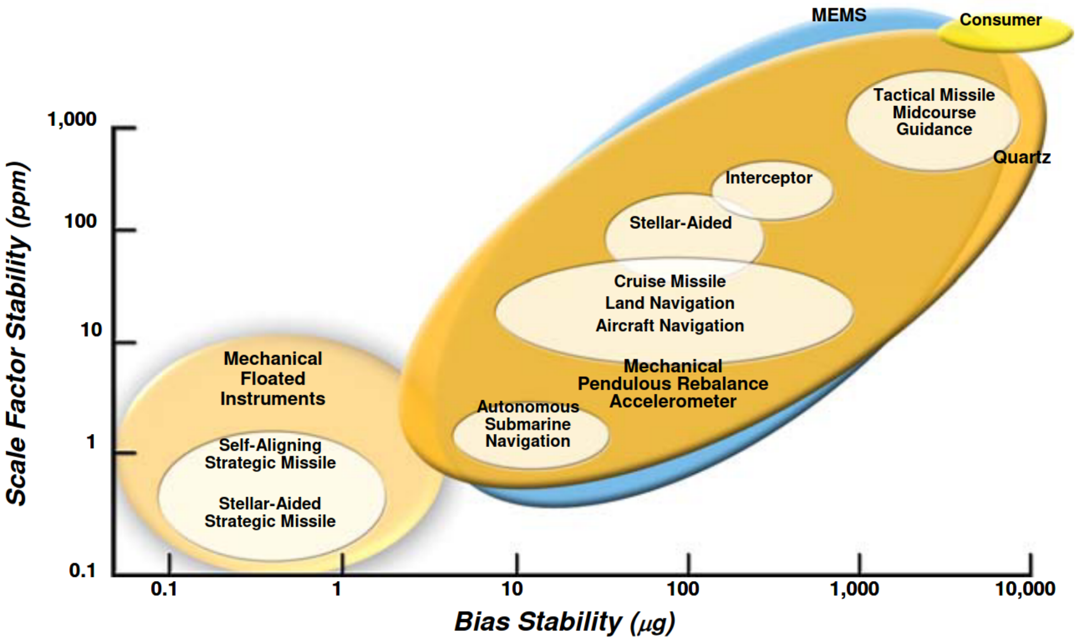

Figure 3.2. Accelerometers technology plotted by bias instability and scale factor stability [4]

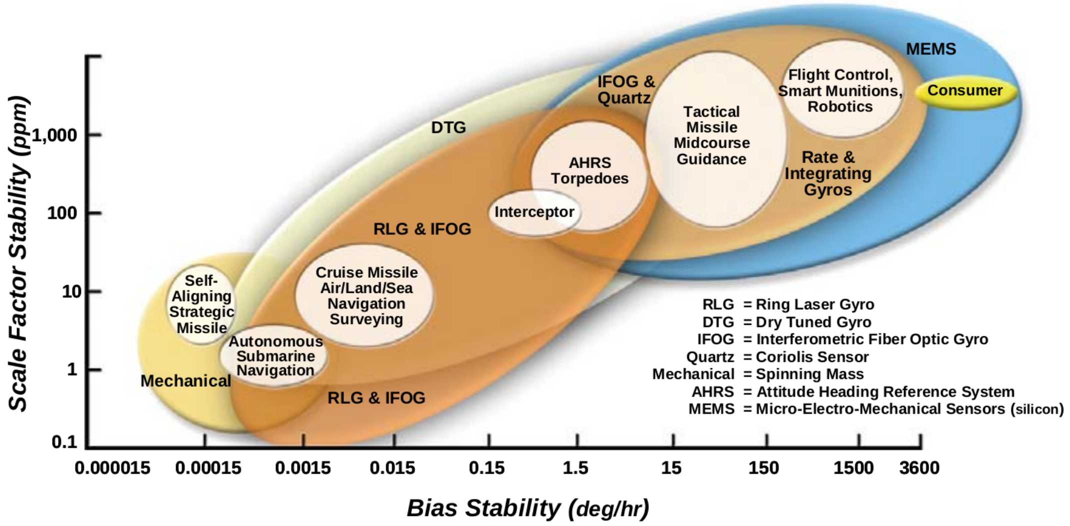

Figure 3.3. Gyroscopes technology plotted by bias instability and scale factor stability [4]

3.2.3 IMU's errors and Allan variance analysis

IMU errors IMUs faces several error sources. In this thesis, the main focus is given to MEMS-based IMU as they are used in experiments. These sensors are typically small and low cost.

These errors can be divided into two categories [3]

- stochastical errors, which can be described as random processes,

- and deterministic errors, also called systematic errors, are basically caused by manufacturing imperfections or not ideal handling with IMU. These errors can be corrected by proper calibration.

Nevertheless, errors need to be analysed and reduced according to application requirements. The following errors are the most significant according to the topic of this thesis.

Biases of accelerometers and gyroscopes used in IMU are examples of systematic errors and can be divided into

- bias instability (or also called in-run bias), which represents drift of the sensor during a time,
- and initial bias (or repeatability bias), which is a static offset, which can be different during each start-up of the device, but during a run, it is static.

Biases are typically represented in $^\circ/hr$ or $^\circ/s$ for gyroscopes and mg for accelerometers.

A scale factor and *a misalignment error*, both systematic errors, could also be significant. The scale factor is connected to imperfection while converting the real measurement input value and output value. The nonorthogonality of all sensors gives the misalignment error in IMU and it is caused during the production.

Angle or *velocity random walks* belong to stochastic errors. The measurement of gyroscopes and accelerometers are subject to white noises (the noise represented by Gaussian distribution). During the estimation of angles and velocities, integration needs to be done. Then the white noise starts to manifest itself by angle or velocity random walk, $(^\circ/s/\sqrt{Hz})$ and $(m^2/s/\sqrt{Hz})$ respectively.

Allan variance(AVAR) is widely used to analyse a random error of inertial sensors in time-domain. The brief introduction and important outcomes from AVAR, the most common time domain measure of frequency stability, is given [5].

The AVAR $\sigma_A^2(\tau)$ is a function of the averaging time τ, computed as

$$\sigma_A^2(\tau) = \frac{1}{2(N-1)} \sum_{i=1}^{N-1} (\overline{y}_\tau(i+1) - \overline{y}_\tau(i))^2, \tag{1}$$

where N represents the number of clusters in the dataset ($N = floor(M/n)$), n is the number of samples in the cluster, M is the total number of samples in dataset, τ is the time length of the cluster ($\tau = m \times T_s$), T_s is the sampling period, $\overline{y}_\tau(i+1)$ and $\overline{y}_\tau(i))$ are mean values of certain cluster of $i+1$-th and i-th cluster respectively. [6].

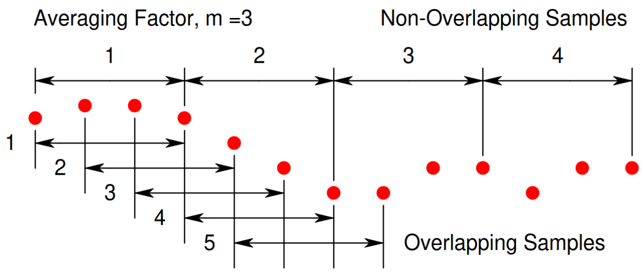

Figure 3.4. The difference between non-overlapping and overlapping sample [5]

The samples in a cluster can be both non-overlapping and overlapping. The difference is illustrated in 3.4. The overlapping samples improve the confidence of the resulting estimate. That is the reason why this method is the most common for a measure of time-domain frequency stability in general [5].

The process of measuring AVAR consist of collecting 24-48 hours long dataset when the inertial sensor is not moving, and it is in not vibrating environments (no trains, subways that would cause vibration). The sampling values are angular rate or accelerations.

If the dataset is valid and the AVAR is correctly computed, the plot copies the example plot seen in Figure 3.5. It is typically plotted on a log/log scale. A different slope of the graph describes each noise component by that the graph can be easily divided into specific parts.

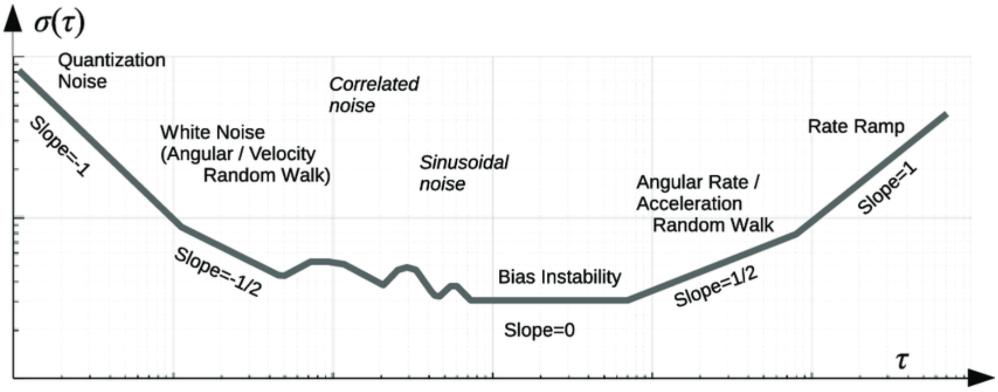

Figure 3.5. An example of Allan variance plot [7]

The most significant outcome for navigation purposes is when the bias instability is reached (slope is zero). At this time, the sensor model contains only a white (Gaussian) noise [8]. After that period, the external reset needs to be done.

3.2.4 Performance of IMUs according to their application

IMUS can be used in various application, which differs by IMUs performance. The overview of each sensor's precision for a given application is nicely summarized in Figure 3.6.

3.2.5 Navigation (mechanization) equations

Both gyroscopes (see section 3.2.2) and accelerometers (see section 3.2.1) measure in IMU inertial frame , typically called body frame. That means they need to be converted to a reference frame. In that frame, the state (positions, orientations, velocities, ...) is estimated and it is the output of the localization method.

Navigation equations implement the transforms between the body frame and the reference frame, either a local-level frame (as North-East-Down or East-North-Up), a reference to a specific point at planet Earth, or an Earth-fixed frame as ECEF [3].

These equations are known and can be found in the book MEMS-based Integrated Navigation [3].

3.3 Odometry

The odometry of a device contains its pose and velocity based on its motion, it can be obtained from various sources as IMU, lidars, cameras or wheel encoders[1], and it is an example of a dead reckoning system.

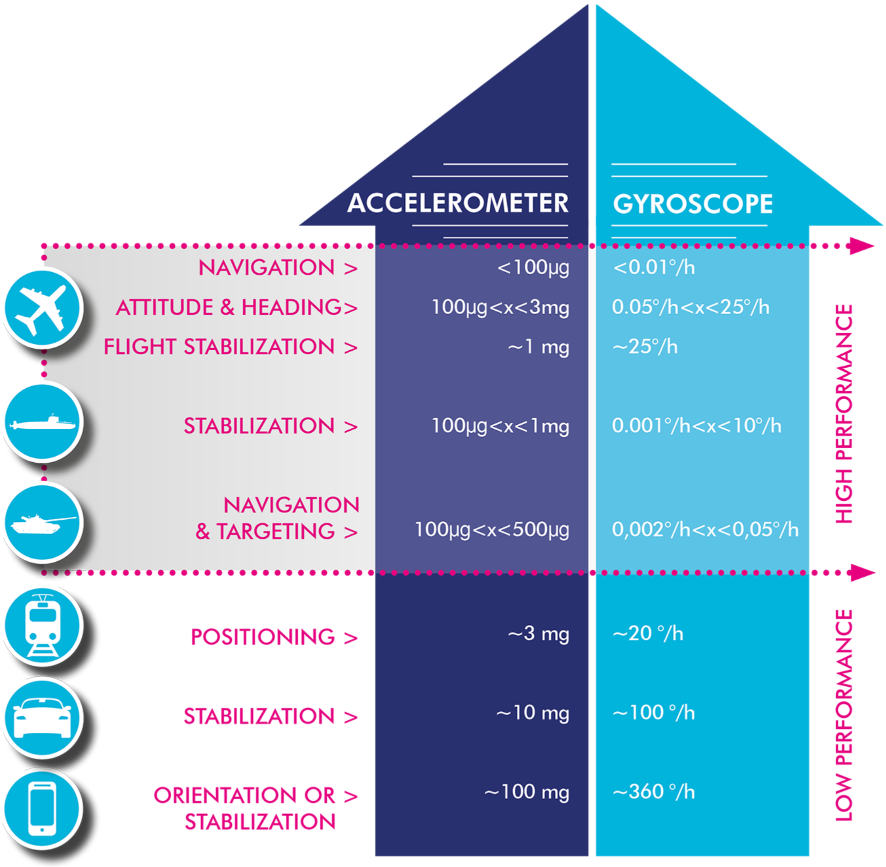
ACCELEROMETER
GYROSCOPE
NAVIGATION >
ATTITUDE & HEADING>
FLIGHT STABILIZATION >
STABILIZATION >
NAVIGATION & TARGETING >
POSITIONING >
STABILIZATION >
ORIENTATION OR > STABILIZATION
<100µg
100µg<x<3mg
~1 mg
100µg<x<1mg
100µg<x<500µg
~3 mg
~10 mg
~100 mg
<0.01°/h
0.05°/h<x<25°/h
~25°/h
0.001°/h<x<10°/h
0,002°/h<x<0,05°/h
~20 °/h
~100 °/h
~360 °/h
HIGH PERFORMANCE
LOW PERFORMANCE

Figure 3.6. A performance of IMU per application [9]

Both IMU and wheel encoders are used in the suggested localisation system since they can counter each other's negative characteristics because wheel encoders drift over travelled distance and IMU drift over time[1].

The details of odometry information varies with vehicle design and during experiments the differential type is used (illustrated in Figure 3.7.

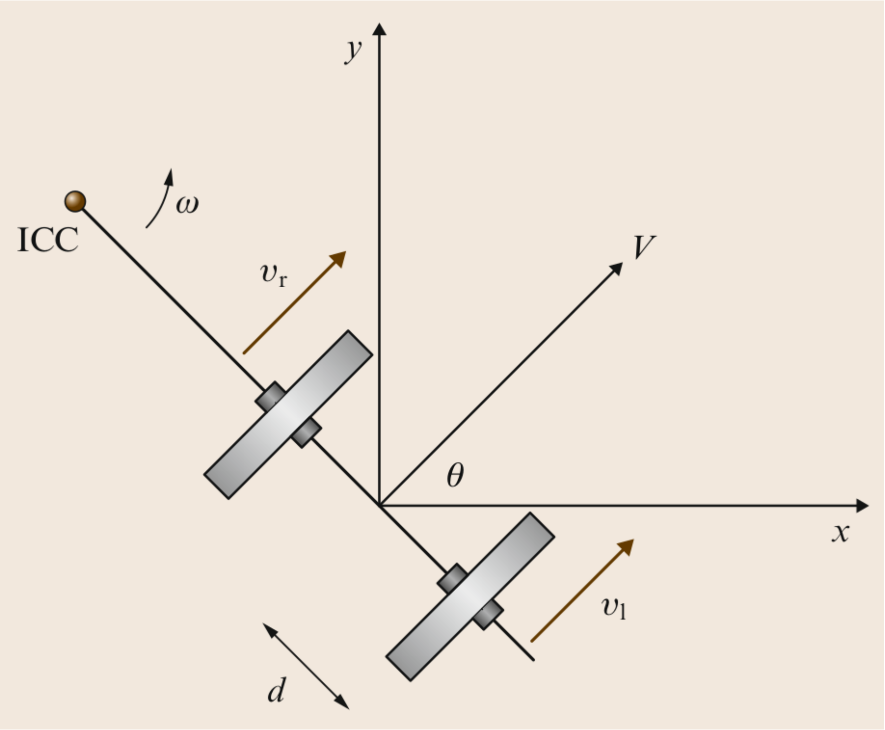
y
ICC
ω
v_r
V
θ
x
v_l
d

Figure 3.7. A differential drive kinematics scheme[1]

Chapter 4

State estimations algorithms for localization

The localisation is the problem of estimating a robot's coordinates in an external reference frame from sensor data. In particular, probabilistic approaches are typically more robust in the face of sensor limitations, sensor noise or environment dynamics[10]. Moreover, they often scale much better to complex and unstructured environments, where the ability to handle uncertainty is of even greater importance[10].

For that reason, only probabilistic methods for state estimation are described in the next chapter. The basis of each technique briefly described in this thesis is Bayes filter[10]. The first section discusses the difference between Kalman and Particle filter for state estimation. The following section introduces advanced concepts derived from the Kalman filter algorithm as extended Kalman filter and unscented Kalman filter. The third and fourth sections go deeply into the Error state extended Kalman filter and introduces the benefits of using it, as it is the core of the localization system of this thesis.

4.1 Kalman and particle filter

Both Kalman and particle filter are the first implementations of Bayes filter in the continuous time[10], and in both filters, the state is represented by belief, which corresponds to a distribution. For the Kalman filter it is a multivariate normal distribution , but for particle filter, the distribution is represented by all particles[10–11].

Both algorithms work with a prediction and correction step, which works with the system and sensor model, respectively. Firstly, it predicts the state based on the internal system model, and secondly, it corrects itself by external measurements and sensor model. Kalman and particle filter algorithms for localization are well described in a referenced Probabilistic robotics[10].

There are a few limitations for both algorithms. For the Kalman filter, the state transitions and measurements need to be linear with added Gaussian noise, and the initial state must have normal distribution[10]. There is no such requirement for linearity for the particle filter, and it works fine with nonlinear or multi-modal systems too[10]. But the algorithm can be more computation demanding as a high number of particles needs to be generated in each sample time for a robust estimation[10].

I decided to use an algorithm based on the Kalman filter for several reasons. First, the localization system will be used for the navigation of vehicles. For these kinds of tasks, the update rate has to be relatively high. Second, the state transition and measurements are approximately linear. And third, these algorithms are typically used in the fusion of IMU and GNSS, as I already mentioned in Chapter 1.

4.2 Algorithms based on Kalman filter

There have been many modifications and extensions of the standard Kalman filter since the 1950s when the filter was invented, because the assumptions of the linear system and sensor model with added Gaussian noise are rarely fulfilled in practice[10]. In these cases, the state transition or sensor model are described by nonlinear functions.

There exist many techniques for linearizing nonlinear functions. The most popular tool, called Extended Kalman filter, use (first-order) Taylor expansion[10]. This approximation has its limitations, which correspond to a degree of nonlinearity of the functions and a degree of uncertainty. The higher these degrees are, the further the approximation deviates from true belief. In general, the Extended Kalman filter has its benefits in simplicity, optimality and robustness[10], but in practice, it is reliable for the system, which is almost linear in one-time step[12].

The Unscented Kalman filter is a tool that appears superior to the EKF linearization[10, 12]. The linearization is given by carefully selected sample points from nonlinear functions. Also, this approach does not assume that the distribution of noise source is Gaussian[12].

In conclusion, the degree of nonlinearity of the system is critical for the state estimation by algorithms based on the Kalman filter. Thus, the relatively recent but promising tool Error state Extended Kalman filter was introduced. In this concept, the error of the state is estimated, as it is more likely correctly modelled by a linear function[13–15].

4.3 Error state Extended Kalman filter

Error state Extended Kalman filter belongs to a group of Indirect Kalman filters because it does not estimate the state itself, but the error of the state[13].

The main idea is that the true state, which should be the output of the system, is computed as a suitable composition of nominal state and the error state

$$x_t = x_n \bigoplus \delta x \tag{1}$$

where

- $\bigoplus$ is a suitable composition as linear sum or matrix product,
- x_t is the true state,
- x_n is the nominal state
- and δx is the error state.

The nominal state is considered a large signal, which can be integrated in its nonlinear form and the error state as a small signal, which is a nearly linear function, ideal for Extended Kalman filtering[14].

The algorithm can be illustrated in the following equations in prediction, correction, injection and resetting steps. In **the prediction step** the nominal state and its covariance is estimated by following equation[13–14]

$$\begin{aligned} x_t &= f(x_{t-1}, u_t) \\ P_t &= F P_{t-1} F^T + B Q B^T \end{aligned} \tag{2}$$

where

- $f(x_{t-1}, u_t)$ is the nonlinear function described the current state of the system based on previous state and current input,
- x_t is the nominal state in time t,
- x_{t-1} is the true state in time $t-1$,

- u_t is the input of the system,
- P_t is the state covariance matrix (also called system covariance),
- F is the state transition matrix given by $F = \frac{\delta f(x_{t-1}, u_t)}{\delta x_{t-1}}$,
- B is the input matrix,
- and Q is the input noise covariance matrix.

In **the correction step** the state is corrected based on measurement according to equations[13–14]

$$\begin{aligned} K &= P_{t-1} H^T (HPH^T + R)^{-1} \\ e_t &= K(y_t - h(x_{t-1})) \\ P_t &= (I - KH) P_{t-1} (I - KH)^T + KRK^T \end{aligned} \tag{3}$$

where

- K is Kalman gain,
- H is the measurement transition matrix given by $H = \frac{\delta h(x_t)}{\delta x_t}$,
- R is the covariance of the measurement,
- e_t current estimated error of the state,
- y_t is the measurement in time t,
- and $h(x_{t-1}$ is the measurement model based on previous state x_{t-1}.

In **the injection and resetting step** the state is compensated by error, and the error state and state covariance matrix need to be reseted[13–14]. The injection of error is given by

$$x_t = x_{t-1} + e_t \tag{4}$$

and the resetting can be illustrated

$$\begin{aligned} e_t &= g(e_t) \\ P_t &= G P_t G^T \end{aligned} \tag{5}$$

where

- $g(e_t)$ is reseting function of error state
- and G is the Jacobian matrix defined as $G = \frac{\delta g(e_t)}{\delta e_t}$.

4.4 Chosen approach for state estimation

This thesis aims to design a prototype of the localization system of vehicles based on the fusion of the UWB positioning system and onboard dead-reckoning sensors. As I already mentioned in Chapter 1, the system should be used for the localization of indoor vehicles in an industrial environment. However, each such vehicle (and terrain, where it is moving) can differ in its dynamics. Because of that, I decided to use an approach, which can replace these dynamics. The INS meets these requirements and gives us the state independent of a specific vehicle and specific terrain where it is moving. The estimation is accurate for a short time (see Section 3.2). Thus it needs to be corrected via other measurements as the UWB positioning system and odometry. I decided to estimate the error of the INS in the Error state Extended Kalman filter for the following reasons.

- The correction of INS can be done at a lower rate than the state estimation for localization purposes itself[13–14],
- The model for estimation of the error is linear, as the error is
  - close to the origin,
  - and small, so higher orders can be neglected[13–14].

- numerical stability of the solution[13],
- even if the error estimation has a temporary computer failure, the INS is not affected, and some emergency procedures can come into account[13].

To sum up, the state estimation is given by INS, and the ES-EKF estimates the errors in the state using the difference between the INS and external sources of data, which are odometry and the UWB positioning system. The INS can estimate high-frequency motions of the vehicle accurately, so these dynamics are not in the filter explicitly modelled. The ES-EKF use a model of error propagation in INS, which are at low frequency and very well modelable by linear functions[13, 16, 15]. This approach is also known as Aided navigation systems and is briefly described in Aided navigation: GPS with high rate sensors[15]. The following Chapter 5 introduce the localization system design and implementation in detail.

Chapter 5

Localization system design and implementation

In this chapter, the design of the architecture is described along with the used tools for its solution.

The first part is proposed the system architecture design, the system kinematics equations follow in the second part. In the third part, the equations for the Error state extended Kalman filter are introduced. The fourth part is dedicated to injecting the estimated error into the estimated state and resetting the injected error. In the fifth and final part, the implementation tools are discussed.

To sum up, this chapter gives the reader a detail introduction to the proposed localization system with all equations and tools used in the final implementation.

5.1 System architecture design

Various approaches for state estimation were introduced in Chapter 4. The chosen approach is the Error state extended Kalman filter(ES-EKF). According to ES-EKF is the error in the states estimated using a Kalman filter rather than the state itself. The benefits of this approach are briefly summarized in Chapter 4.

The system consists of three crucial steps. The first is the inertial navigation unit (INS), where the state is estimated based on IMU measurements. This state estimation leads to a dead-reckoning system, where the drift grows with time and needs to be corrected.

The second step is the ES-EKF itself. The error of the state is calculated and is then corrected using measurements from UWB localization and odometry. Measurements from UWB localization and odometry observe the error. The UWB localization gives us the absolute position, which can reduce the drift in step one.

The third part is injecting error into INS estimation and resetting the ES-EKF while the injection is done. Finally, the output of the whole system is given by the INS solution. The system requires initial states with covariances for INS and ES-EKF set up first. The simplified architecture is illustrated in Figure 5.1 and described in the following section in detail[14].

For navigation purposes, the regular rate of pose estimation is hundreds of Hertz[17]. The INS provides a full state estimate with the IMU update rate, which usually satisfies this requirement. Furthermore, it gives us the state estimation utterly independent of external factors, for example, slippering wheels[17].

As UWB localization and odometry measurements cannot give us a much higher rate than tens Hertz, they are used only in the correction step.

In other words, the most dynamic part of the estimation is somehow independent of Kalman filtering. On the one hand, the state estimation in INS is simple and fast. On the other hand, the error estimation can be more computationally demanding as the computation of Jacobians needs to be done. Therefore the separation of state

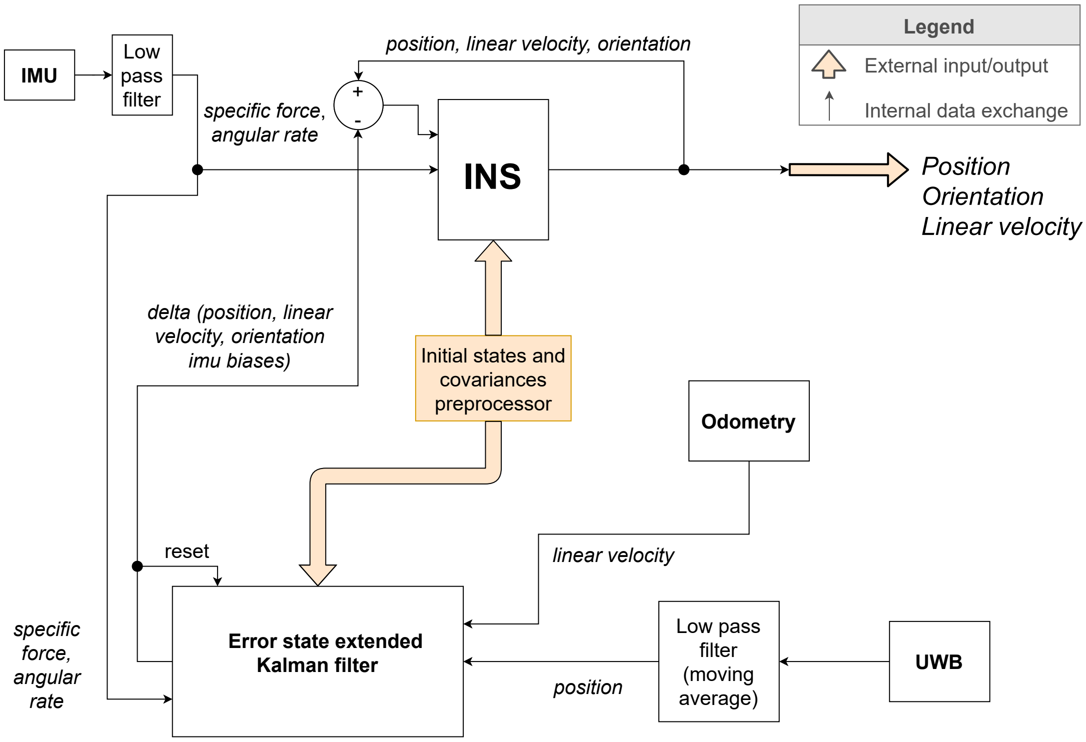
Legend
External input/output
Internal data exchange
IMU
Low pass filter
position, linear velocity, orientation
+
-
specific force, angular rate
INS
Position
Orientation
Linear velocity
delta (position, linear velocity, orientation imu biases)
Initial states and covariances preprocessor
Odometry
linear velocity
reset
specific force, angular rate
Error state extended Kalman filter
position
Low pass filter (moving average)
UWB

Figure 5.1. The proposed architecture of the localization system

estimation and error estimation cause that the calculation of state is sufficiently fast. The error state is estimated separately in ES-EKF and is injected into the state only if another measurement than IMU comes in. The only requirement is that the correction needs to be applied before non-gaussian noise in IMU measurement is significant. That correction reduces the drift of the dead-reckoning system.

In conclusion, the benefit of this architecture is state estimation at a high rate, independent of external events. The state is corrected at a lower rate but faster than the non-gaussian noise becomes significant in state estimation. That brings us the best aspects of all types of sensors, which are used in the architecture.

In the following sections, the output of the system is called the navigation state (i.e., position, linear velocity, and attitude).

5.2 System kinematics

For a more detailed introduction to system architecture, the system kinematics equations need to be announced. But before I enter that itself, let me describe an important topic, which represents attitude and rotation in 3D space.

5.2.1 Representation of 3D attitude and rotation in space

There are many ways how to represent 3D attitude and rotation in space. The most commonly used representations in the field of robotics are

- rotation matrices,

- Euler angles,
- axis-angle
- and quaternions[1].

To not go into much detail, each representation has its pros and cons and applications, where it has its purpose. The rotation matrix is chosen as the internal representation of orientation and the quaternion as an output.

There are several reasons why to pick this representation[18–19]. Firstly, quaternions and rotation matrices do not suffer from singularities as Euler and fixed angles do[18]. Secondly, quaternion gives us a compact representation. And finally, these two are the most recommended representation in ROS standard rep-103[19]. As quaternions have many internal models in different libraries (Eigen library in C++[20], geometry messages library[21] or the transform library tf2 in ROS[22]) and the representation is not easy to imagine, I decided to use the quaternions only as an output and rotation matrix as the internal representation.

5.2.2 The kinematics equations in continuous time

The kinematics formulas in continuous time, that relates the inertial sensor measurements to the **true navigation state**, is well-known [14–15, 3, 2]. Therefore, I did not have to derive equations myself and used the one derived in [quaternion_kinematics] equation 235. The only difference is that orientation is in the rotation matrix and not quaternion. Equations are

$$\begin{aligned}
\dot{p_t} &= v_t \\
\dot{v_t} &= R_t(a_m - a_{bt} - a_n) + g_t \\
\dot{R_t} &= R_t(\Omega_t) \\
\dot{a_{bt}} &= a_w \\
\dot{\omega_{bt}} &= \omega_w \\
\dot{g_t} &= 0,
\end{aligned} \tag{1}$$

where

- p_t is true position in 3D $[m]$,
- v_t is true linear velocity in 3D $[m \cdot s^{-2}]$,
- R_t is true rotation matrix of orientation,
- a_m is specific force given by accelerometers $[m \cdot s^{-2}]$,
- a_{bt} is true accelerometer bias $[m \cdot s^{-2}]$,
- a_n is accelerometers white Gaussian noise $[m \cdot s^{-2}]$,
- a_w is white Gaussian noise accelerometers bias $[m \cdot s^{-2}]$,
- g_t is true gravity vector $[m \cdot s^{-2}]$,
- $\Omega_t = [(\omega_m - \omega_{bt} - \omega_n)_\times] = \begin{bmatrix} 0 & -(\omega_{m_3}-\omega_{bt_3}-\omega_{n_3}) & \omega_{m_2}-\omega_{bt_2}-\omega_{n_2} \\ \omega_{m_3}-\omega_{bt_3}-\omega_{n_3} & 0 & -(\omega_{m_1}-\omega_{bt_1}-\omega_{n_1}) \\ -(\omega_{m_2}-\omega_{bt_2}-\omega_{n_2}) & \omega_{m_1}-\omega_{bt_1}-\omega_{n_1} & 0 \end{bmatrix}$ is true skew-symetric matrix (a tensor of angular velocity) $[\frac{rad}{s}]$,
- ω_m is angular rate given by gyroscopes $[\frac{rad}{s}]$,
- ω_{bt} is true bias of gyroscopes $[\frac{rad}{s}]$,
- ω_n is gyroscopes white Gaussian noise $[\frac{rad}{s}]$,
- and ω_w is white Gaussian noise gyroscopes bias $[\frac{rad}{s}]$.

The state x_t, is governed by IMU noisy reading u_m and perturbed by white Gaussian noise w, defined by

$$\begin{aligned} x_t &= [p_t, v_t, R_t, a_{bt}, \omega_{bt}, g_t]^T \\ u_t &= [a_m - a_n]^T \\ w_t &= [a_w, w_w]^T . \end{aligned} \tag{2}$$

The output of the localization system is **navigation state** (also called nominal), which correspons to the system kinematics, but does not take into account the noise terms w_t and other possible model imperfections (see equation 237 in [14], hence it is simplified to

$$\begin{aligned} \dot{p} &= v \\ \dot{v} &= R(a_m - a_b) + g \\ \dot{R} &= R(\Omega) \\ \dot{a_b} &= 0 \\ \dot{\omega_b} &= 0, \\ \dot{g} &= 0, \end{aligned} \tag{3}$$

where

- p is position in 3D $[m]$,
- v is linear velocity in 3D $[m \cdot s^{-2}]$,
- R is the rotation matrix of orientation,
- a_m is specific force given by accelerometers $[m \cdot s^{-2}]$,
- a_b is accelerometer bias $[m \cdot s^{-2}]$,
- g is gravity vector $[m \cdot s^{-2}]$,
- $\Omega = [(\omega_m - \omega_b)_\times] = \begin{bmatrix} 0 & -(\omega_{m_3} - \omega_{b_3}) & \omega_{m_2} - \omega_{b_2} \\ \omega_{m_3} - \omega_{b_3} & 0 & -(\omega_{m_1} - \omega_{b_1}) \\ -(\omega_{m_2} - \omega_{b_2}) & \omega_{m_1} - \omega_{b_1} & 0 \end{bmatrix}$ is skew-symetric matrix (a tensor of angular velocity) $[\frac{rad}{s}]$,
- ω_m is angular rate given by gyroscopes $[\frac{rad}{s}]$,
- ω_b is bias of gyroscopes $[\frac{rad}{s}]$.

The linearized dynamics (see equation 238 in [14]) of the **error state** are

$$\begin{aligned} \dot{\delta p} &= \delta v \\ \dot{\delta v} &= -R[a_m - a_b]_\times \delta\Theta - R\delta a_b + \delta g - R a_n \\ \dot{\delta\Theta} &= -[\omega_m - \omega_b]_\times \delta\Theta - \delta\omega_b - \omega_n \\ \dot{\delta a_b} &= a_w \\ \dot{\delta \omega_b} &= \omega_w \\ \dot{\delta g} &= 0, \end{aligned} \tag{4}$$

where

- δp is the position error in $[m]$,
- δv is the linear velocity error in $[m \cdot s^{-2}]$,
- $\delta\Theta$ is the orientation error,
- δa_b is acceleration bias error $[m \cdot s^{-2}]$,
- $\delta\omega_b$ is gyroscope bias error $[\frac{rad}{s}]$,
- δg is gravity vector error $[m \cdot s^{-2}]$,
- R is rotation matrix given by nominal state,

- a_m is specific force given by accelerometers $[m \cdot s^{-2}]$,
- a_b is accelerometer bias $[m \cdot s^{-2}]$,
- a_n is accelerometers white Gaussian noise $[m \cdot s^{-2}]$,
- a_w is white Gaussian noise accelerometers bias $[m \cdot s^{-2}]$,
- ω_m is angular rate given by gyroscopes $[\frac{rad}{s}]$,
- ω_b is bias of gyroscopes $[\frac{rad}{s}]$,
- ω_n is gyroscopes white Gaussian noise $[\frac{rad}{s}]$,
- and ω_w is white Gaussian noise gyroscopes bias $[\frac{rad}{s}]$. .

Note that higher orders in linearization are neglected since the error state is small compared to the navigation state.

During **filter correction phase**, measurements from UWB localization and odometry comes into account. Usual, the sensor delivers measurements that depend on the state, such as

$$y = h(x_t) + \rho, \tag{5}$$

where $h(t)$ is a general nonlinear function of the system state (the true navigation state), and ρ is a white Gaussian noise with covariance. For *UWB localization* the function is simple as it is

$$y_1 = p_t + \rho_1, \tag{6}$$

with covariance R_1. But for *odometry* it is a little bit complicated

$$y_2 = R_t^{-1} v_t + \rho_2 \tag{7}$$

with covariance R_1. This difference is important in the computation of Jacobian for the ES-EKF algorithm.

5.2.3 The kinematics equations in discrete time

As the equations in continuous time are derived from book [14], where their representation in discrete time is also presented, I decided to write down only parts, which are different. For more detail see equations 260 in [14]. The equation 260c is slightly different since I am using rotation matrix for orientation representation and not quaternion. This equation is changed to

$$R \leftarrow R(\Omega \Delta t), \tag{8}$$

where

- R is the rotation matrix of orientation,
- $\Omega = [(\omega_m - \omega_b)_\times] = \begin{bmatrix} 0 & -(\omega_{m_3} - \omega_{b_3}) & \omega_{m_2} - \omega_{b_2} \\ \omega_{m_3} - \omega_{b_3} & 0 & -(\omega_{m_1} - \omega_{b_1}) \\ -(\omega_{m_2} - \omega_{b_2}) & \omega_{m_1} - \omega_{b_1} & 0 \end{bmatrix}$ is skew-symetric matrix (a tensor of angular velocity) $[\frac{rad}{s}]$,
- ω_m is angular rate given by gyroscopes $[\frac{rad}{s}]$,
- ω_b is bias of gyroscopes $[\frac{rad}{s}]$.

This integration is happening in the INS box in Figure [23].

5.3 Error state extended Kalman filter implementation

Algorithm and equations for general extended Kalman filter are briefly described in Chapter 4. In this section, these equations are concretized.

The **error state system** is now

$$\delta x \leftarrow f(x, \delta x, u_m, i) = F_x(x, u_m) \cdot \delta x + F_i \cdot i, \tag{9}$$

where

- i is a perturbation vector (usually modelled as white Gaussian noise).

The **Es-EKF prediction part** is given by

$$\begin{aligned} \hat{\delta x} &\leftarrow F_x(x, u_m) \cdot \hat{\delta x} \\ \hat{P} &\leftarrow F_x P F_x^T + F_i Q F_i^T \end{aligned}, \tag{10}$$

where

- P is a process covariance matrix,
- F_x is transition matrix,
- F_i is Jacobian of error state system by impulses,
- Q is covariances of process noise,

The **transition matrix** (also called system matrix) F_x is error state Jacobian and it is simple determined by error state kinematics equations $f(\delta x_t)$ in discrete time in Section 5.2.3,

$$F_x = \frac{\partial f(\delta x, u_m)}{\partial \delta x}$$

$$F_x = \begin{bmatrix} I & I\Delta t & 0 & 0 & 0 & 0 \\ 0 & I & -R[a_m - a_b]_\times \Delta t & -R\Delta t & 0 & I\Delta t \\ 0 & 0 & R\{\omega_m - \omega_b\}^T \Delta t & 0 & -I\Delta t & 0 \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{bmatrix}. \tag{11}$$

F_i is given by

$$F_i = \left.\frac{\partial f}{\partial i}\right|_{x, u_m} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{12}$$

The covariances matrix is given by random impulses applied to the velocity, orientation and bias estimates, modelled by white Gaussian noise [14]

$$Q = \begin{bmatrix} \sigma_{a_n}^2 \Delta t^2 I & 0 & 0 & 0 \\ 0 & \sigma_{\omega_n}^2 \Delta t^2 I & 0 & 0 \\ 0 & 0 & \sigma_{a_w}^2 \Delta t^2 I & 0 \\ 0 & 0 & 0 & \sigma_{\omega_w}^2 \Delta t^2 I \end{bmatrix}, \tag{13}$$

where

- σ_{a_n} is standard deviation of accelerometers $[m \cdot s^{-2}]$,
- σ_{ω_n} is standard deviation of accelerometers $[\frac{rad}{s}]$
- σ_{a_w} is velocity random walk $[\frac{rad}{s\sqrt{s}}]$,
- σ_{ω_w} is angular random walk $[\frac{rad}{s\sqrt{s}}]$.

This information can be obtained from the datasheet or AVAR (see Section 3.2.3).

The **ES-EKF correction part** is given by

$$\begin{aligned} K &\leftarrow PH^T(HPH^T+R)^{-1} \\ \delta x &\leftarrow K(y-h(\widehat{\delta x})) \\ P &\leftarrow (I-KH)\hat{P}(I-KH)^T+KRK^T \end{aligned} \quad , \tag{14}$$

where

- K is Kalman gain,
- H is observation matrix,
- R is covariances of observation noise,
- P is process covariance,
- y is an observation,
- $h(\hat{x})$ is an observation model,
- δx is a error state.

The bf observation matrices differs for *UWB localization* (H_1) and *odometry* (H_2)

$$\begin{aligned} H_1 &= [\, I \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \,] \\ H_2 &= [\, 0 \quad R_t^T \quad -R_t^T[v_t]_\times J_r(\Theta) \quad 0 \quad 0 \quad 0 \,] \,, \end{aligned} \tag{15}$$

where

- R_t is orientation in navigation state,
- v_t is linear velocity in navigation state,
- Θ is orientation R_t in rotation vector form,
- J_r is right jacobian of rotation group $SO(3)$ (see equation 183 in [14]).

To obtain H_2 from Equation (7), a reader should notice a Jacobian with respect to the rotation vector in section 4.3.4 and equation 188 in [14].

5.4 Injection the error state into the navigation state

While the correction phase is done, the estimated error state comes into account in the navigation state

$$x \leftarrow x \bigoplus \delta x, \tag{16}$$

where

$\bigotimes$ appropriate composition of sums or rotation product.

The equations are

$$\begin{aligned} p &\leftarrow p+\delta p \\ v &\leftarrow v+\delta v \\ R &\leftarrow R^* R\{\delta\Theta\} \\ a_b &\leftarrow a_b+\delta a_b \\ \omega_b &\leftarrow \omega_b+\delta\omega_b \\ g &\leftarrow g+\delta g \end{aligned} \tag{17}$$

where

$R\{\delta\Theta\}$ orientation error in rotation matrix.

The injection of the error state is essential, but the resetting of the error state must also be done.

5.5 Implementation tools

This section briefly introduces tools used for implementation: the ROS2[24] framework, C++ and Python languages.

ROS2[24] is a set of software libraries and tools for building robot applications. It is open-source, and it consists of drivers for hardware, state-of-the-art algorithms, tools for debugging, visualisation, simulation, communications overall processes. All applications created in ROS2 are easy to share and used in the community. It supports all most known and most used programming languages like C++, Python, Java, Lua or Lisp. ROS2 distributions are released to work on operating systems like Ubuntu, MacOs or Windows. Nevertheless, as it is open-source, users usually use it with one Ubuntu distributions, such as 20.04 or 18.04.

The newest version of ROS is ROS2, introduced in 2014 at the conference ROSCon 2014 in Chicago[24], but the first distribution was released in May 2019. There are several distributions of ROS2 yet, the localisation system and experiments are implemented using Foxy Fitzroy1 which was released in June 20202.

ROS2 has defined code style and languages version which are recommended to use. The implementation sticks to these rules and uses C++17 and Python3. As the localisation needs to be implemented as a real-time application, it is implemented in C++17. Python3 is used for the visualisation of experiments results and supporting scripts.

---

1 The documentation to ROS2 Foxy Fitzroy `https://docs.ros.org/en/foxy/index.html`

2 The list of all distributions of ROS2 `https://docs.ros.org/en/galactic/Releases.html`

Chapter 6

Experiments

Lorem ipsum sit amet

6.1 Section 1

Lorem ipsum sit amet

Chapter 7

Conclusion and future work

Lorem ipsum sit amet

7.1 Section 1

Lorem ipsum sit amet

7.2 Section 2 future work

- kalibrace sensoru poradna - pouziti virtualnich sensoru (robot stoji, robot jede jen rovne...) k tomu, aby estimace byla lepsi

References

[1] Bruno Siciliano, and Oussama Khatib. *Springer Handbook of Robotics*. Springer-Verlag, 2007.

[2] Peter Teunissen, and Oliver Montenbruck. *Springer handbook of global navigation satellite systems*. Springer, 2017.

[3] Priyanka Aggarwal. *MEMS-based integrated navigation*. Artech House, 2010.

[4] N. Barbour, and G. Schmidt. Inertial sensor technology trends. *IEEE Sensors Journal*. 2001, 1 (4), 332-339. DOI 10.1109/7361.983473.

[5] William Riley, and David Howe. *Handbook of Frequency Stability Analysis*. 2008. `https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=50505`.

[6] Shuvra S Bhattacharyya, Ed F Deprettere, Rainer Leupers, and Jarmo Takala. *Handbook of signal processing systems*. Springer, 2018.

[7] Agnieszka Szczęsna, Przemysław Skurowski, Ewa Lach, Przemyslaw Pruszowski, Damian Pęszor, Marcin Paszkuta, Janusz Słupik, Kamil Lebek, Mateusz Janiak, Andrzej Polanski, and Konrad Wojciechowski. Inertial Motion Capture Costume Design Study. *Sensors*. 2017, 17 612. DOI 10.3390/s17030612.

[8] Wikipedia®. *Wikipedia - white noise definition*. `https://en.wikipedia.org/wiki/White_noise`.

[9] © Thales group. *Performance of IMU per application*. `https://www.thalesgroup.com/sites/default/files/database/d7/assets/images/thales_topaxyz_imu_infographie_copyright_thales_light_0.png`.

[10] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. Cambridge, Mass.: MIT Press, 2005 . ISBN 0262201623 9780262201629.

[11] Nak Yong Ko, and Tae Gyun Kim. *Comparison of Kalman filter and particle filter used for localization of an underwater vehicle*. In: *2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. 2012. 350-352.

[12] Simon J. Julier, and Jeffrey K. Uhlmann. *New extension of the Kalman filter to nonlinear systems*. In: Ivan Kadar, eds. *Signal Processing, Sensor Fusion, and Target Recognition VI*. SPIE, 1997. 182 – 193. `https://doi.org/10.1117/12.280797`.

[13] S.I. Roumeliotis, G.S. Sukhatme, and G.A. Bekey. *Circumventing dynamic modeling: evaluation of the error-state Kalman filter applied to mobile robot localization*. In: *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*. 1999. 1656-1663 vol.2.

[14] Joan Solà. Quaternion kinematics for the error-state KF. 2015,

[15] Jay Farrell. *Aided navigation: GPS with high rate sensors*. McGraw-Hill, Inc., 2008.

[16] Venkatesh Madyastha, Vishal Ravindra, Srinath Mallikarjunan, and Anup Goyal. *Extended Kalman Filter vs. Error State Kalman Filter for Aircraft Attitude Estimation.* In: 2011. ISBN 978-1-60086-952-5.

[17] Jay A. Farrell, and Paul F. Roysdon. Advanced Vehicle State Estimation: A Tutorial and Comparative Study. *IFAC-PapersOnLine.* 2017, 50 (1), 15971-15976. DOI https://doi.org/10.1016/j.ifacol.2017.08.1751. 20th IFAC World Congress.

[18] M. D. Shuster. Survey of attitude representations. *Journal of the Astronautical Sciences.* 1993, 41 (4), 439-517.

[19] Mike Purvis ROS, Tully Foote. *Standard Units of Measure and Coordinate Conventions.*
`https://www.ros.org/reps/rep-0103.html`.

[20] Gaël Guennebaud, Benoît Jacob, and others. *"Eigen", a C++ template library for linear algebra: matrices, vectors, numerical solvers, and related algorithms.* http://eigen.tuxfamily.org. 2010.

[21] Michel Hidalgo ROS, Tully Foote. *ROS geometry messages package.*
`http://wiki.ros.org/geometry_msgs`.

[22] Tully Foote. *tf: The transform library.* Open-Source Software workshop. 2013.

[23]

[24] Dirk Thomas, William Woodall, and Esteve Fernandez. *Next-generation ROS: Building on DDS.* In: *ROSCon Chicago 2014.* Mountain View, CA: Open Robotics, 2014.
`https://vimeo.com/106992622`.

Appendix A

Abbreviations and symbols

A.1 A list of abbreviations

All abbreviations used in this thesis are listed below.

AVAR	Allan variance.
DOF	Degrees of freedom.
ES-EKF	Error state extended Kalman filter.
IMU	Inertial measurement unit.
INS	Inertial navigation system.
MEMS	Microelectromechanical systems.
UWB	Ultra-wideband.

A.2 A list of symbols

π	Lorem ipsum sit amet.