



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

DV

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 STOUCHE - 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, JIAJIN, 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

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, ins, localizace, vnitřní prostředí, kalman-filtr, ekf

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

The most awesome abstract

Keywords: ultra-wideband, imu, ins, localization, indoor, kalman-filter, ekf, indoor localization

Contents /

| | |
|---|-----------|
| 1 Introduction | 1 |
| 1.1 Aims and requirements | 1 |
| 1.2 Structure of the thesis | 2 |
| 2 Sensors | 3 |
| 2.1 Localization based on Ultra-wideband | 3 |
| 2.2 Inertial measurement unit | 5 |
| 2.2.1 Accelerometers | 6 |
| 2.2.2 Gyroscopes | 6 |
| 2.2.3 IMU's errors and Allan variance analysis | 7 |
| 2.2.4 Performance of IMUs according to their application | 9 |
| 2.3 Inertial navigation systems . . | 10 |
| 2.4 Odometry | 10 |
| 3 Data fusion principles in pose estimation | 12 |
| 3.1 Kalman and particle filter . . . | 12 |
| 3.2 Algorithms based on Kalman filter | 13 |
| 3.3 Error state Extended Kalman filter | 13 |
| 3.4 Chosen approach for the state estimation | 14 |
| 4 Localization system design and implementation | 16 |
| 4.1 System architecture design . . . | 16 |
| 4.2 System kinematics | 17 |
| 4.2.1 Representation of 3D attitude and rotation in space | 17 |
| 4.2.2 The kinematics equations in continuous time . | 18 |
| 4.2.3 The kinematics equations in discrete time . . . | 20 |
| 4.3 Error state extended Kalman filter implementation . | 20 |
| 4.4 Injection the error state into the navigation state | 22 |
| 4.5 Implementation tools | 23 |
| 5 Experiments | 24 |
| 5.1 Used hardware description . . . | 24 |
| 5.1.1 Sensors specification | 24 |
| 5.1.2 CART2 platform description and sensors placements | 27 |
| 5.2 Experiments in the lab at Datavision s. r. o. | 29 |
| 5.2.1 Experimental lab description | 29 |
| 5.2.2 Description of experiments | 29 |
| 5.2.3 Evaluation of experiments . | 31 |
| 5.3 Experiments at CIIRC | 32 |
| 5.3.1 Experiments at CIIRC description | 32 |
| 5.3.2 Evaluation of experiments at CIIRC | 32 |
| 6 Conclusion and future work | 34 |
| 6.1 Mozne vyuuziti teto fuze v prumyslu | 34 |
| 6.2 Evaluace vysledku z experimentu celkova (discussion) . | 34 |
| 6.3 Future work | 34 |
| References | 35 |
| A Abbreviations and symbols | 39 |
| A.1 A list of abbreviations | 39 |

Tables / Figures

| | | |
|-------------|--|----|
| 5.1 | Technical specifications of Epson M-G365PDF1 | 25 |
| 5.2 | System performance of MDEK1001 | 27 |
| 5.3 | Summary of transforms for experiments with CART2..... | 29 |
| 5.4 | Summary of transforms for experiments setup at Datavision s. r. o. | 29 |
| 5.5 | Velocities during experiments in Datavision s. r. o..... | 32 |
| 2.1 | Two way ranging communication | 4 |
| 2.2 | IMU block diagram..... | 5 |
| 2.3 | Accelerometers technology plotted by bias instability and scale factor stability. | 6 |
| 2.4 | Gyroscopes technology plotted by bias instability and scale factor stability. | 7 |
| 2.5 | The difference between non-overlapping and overlapping sample. | 8 |
| 2.6 | An example of Allan variance plot..... | 9 |
| 2.7 | A performance of IMU per application. | 9 |
| 2.8 | Example of coordinate frames used in INS | 10 |
| 2.9 | Example of body and sensor frame | 10 |
| 2.10 | Schema of strapdown INS..... | 11 |
| 2.11 | A differential drive kinematics scheme..... | 11 |
| 4.1 | The proposed architecture of the localization system. | 17 |
| 5.1 | IMU used in experiments Epson M-G365PDF1 | 24 |
| 5.2 | Experimental setup for static data acquisition of Epson M-G365PDF1 | 24 |
| 5.3 | Overlapping Allan variance plot for Epson M-G365PDF1. . | 25 |
| 5.4 | Moving average filter characteristics for Epson M-G365PDF1..... | 26 |
| 5.5 | DWM1001-DEV development boards. | 26 |
| 5.6 | Positioning of anchors and tags..... | 27 |
| 5.7 | AprilTag used for detection of CART2 position | 27 |
| 5.8 | Camera used for AprilTag detection Niceboy Stream Pro . | 27 |
| 5.9 | A photo of the utilized CART2 platform | 28 |

| | | |
|-------------|---|----|
| 5.10 | A photo of sensors place- ments on CART2..... | 28 |
| 5.11 | Illustration of coordinate sys- tems at CART2 | 28 |
| 5.12 | Experimental setup at Datavision s. r. o..... | 30 |
| 5.13 | Trajectories for experiments at Datavision s. r. o..... | 31 |
| 5.14 | Experimental setup at Intel- ligent and mobile robotics lab (CIIRC) | 33 |

Chapter 1

Introduction

Indoor real-time localization is a crucial component in autonomous mobile robotics, and nowadays, the interest for precise localization is growing due to the fourth industrial revolution influencing all industries. There are existing approaches and technologies to deal with indoor localization in the industry but do not fully meet all the requirements of the fourth industrial revolution.

One of the requirements is to keep as few manual interventions as possible for the technology to work reliably for a long time. An example is autonomous ground vehicles (AGV) by a company Ceit in Škoda factories, which are localized by continuous magnetic tapes physically mounted on the factory's ground. The AGV follow these tapes, which can be damaged by moving AGVs or people and can not be easily replaced or modified.

Another possible requirement is the modularity of the factory environment. Today's trend is to create a factory that consists of modular parts, and the entire production process is assembled according to current needs. Therefore, approaches based on the principle of creating a map based on landmarks or contours from cameras or lidars and subsequent localization in the map created in this way may fail. These approaches must address long-term sustainability, and this topic is not straightforward and can lead to difficulties.

The demand is also for a localization, which will be universally applicable to various robotic platforms, whether ground vehicles of different shapes and constructions or flying drones. This requirement is very well met in the outdoor environment by the Global Navigation Satellite System (GNSS), which unfortunately is not suitable for use in the indoor environment. The use of external beacons seems to be a reasonable solution because it has similar properties as GNSS. These beacons can be based, for example, on ultrasound or radio waves as Ultra wide-band or Wi-Fi.

A single technology cannot meet these requirements, but the appropriate fusion of carefully chosen approaches can.

1.1 Aims and requirements

This thesis is assigned by the Czech company Datavision s. r. o. and is a part of a project called *Guidance and Localization upgrade creating Autonomous Mobile Robots*. The shortcut of this project is **REX**, and it is also used in this thesis.

REX aims to create fleet management of autonomous mobile robots, including their localization, control, navigation and planning. The project is co-financed by the Technology Agency of the Czech Republic (TACR) under the TREND Programm FW03010020 and aims to satisfy the fourth industrial revolution requirements.

This thesis aims to propose an indoor real-time localization system, which includes both position and orientation of AGV. This aim is closely connected with the specification of localization in the REX system but is simplified into 2D. The fundamentals need to be reusable in 3D with a few modifications.

The essential aim is to design a localization system based on ultra wide-band technology and onboard dead reckoning sensors, which should improve the UWB localization itself.

1.2 Structure of the thesis

This thesis is organized into six major parts. The following Chapter 2 describes the fundamentals of used sensors and the concept of inertial navigation systems. Chapter 3 discusses existing data fusion algorithms for pose estimation algorithms and their pros and cons. The proposed localization and its implementation is described in Chapter 4. In Chapter 5, the proposed localization is experimentally evaluated in two environments. The thesis is concluded in Chapter 6, where the usage of this system in the industrial environment is given, followed by a summary of the thesis output and a few proposals for improving and extending the work.

Chapter 2

Sensors

In Chapter 2 the overview of used sensors and their properties is given. The main aim is to describe localization methods based on ultra-wideband technology(UWB), an inertial measurement unit (IMU) followed by inertial navigation systems (INS), and odometry. The section dedicated to UWB localization briefly introduces the UWB signals follows with three most used localization techniques: Two-Way Ranging, Time Difference of Arrival, and Reverse Time Difference of Arrival. 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. This section is closely related to the INS section, where principles of INS are explained, and a strap-down implementation example is given. The last section includes basic information about odometry and its benefits and which odometry is used during experimental verification of the proposed localization system.

2.1 Localization based on Ultra-wideband

Ultra-wideband is an emerging wireless personal area network (PAN) radio technology with wide possibilities of use. The most promising usage in the field of robotics is accurate indoor localization because its benefits include

- high data rates,
- high time resolution,
- low power consumption,
- multipath immunity,
- low costs,
- small size
- and simultaneous ranging communication[1].

The UWB signal is defined as a signal with an absolute bandwidth (B) of at least 500 MHz, defined as

$$B = f_H - f_L, \quad (1)$$

where f_H is the upper frequency and f_L is the lower frequency, or with a fractional bandwidth of larger than 20 % given by

$$B_{frac} = \frac{B}{f_c} = \frac{B}{\frac{f_H+f_L}{2}} [2]. \quad (2)$$

As the bandwidth of the signal is wide, the power spectral density is low. It means, that even that UWB signals share a spectrum with some narrowband signals such as WiFi, it basically behave as environmental noise and does not effect any other narrowband signals much[2]. Also, as the length of each pulse is small, the possibility of overlapping the original pulse is reduced. Thus it should be robust against multipath problem[2].

For precise communication, a direct line of sight should be established between the transmitter and receiver. However, as UWB signal consists of many frequencies, some of them can reflect well from some object, while others can penetrate through them[2].

There exists a few **localization techniques based on UWB** signals, which are exchanged between tag and several reference anchors with known position[2]. The more accurate results in the line of sight environments are based on measuring the signal's time of flight from several devices, namely Two-Way Ranging (TWR), Time Difference of Arrival (TDoA) and Reverse Time Difference of Arrival (RTDoA). The overview of these technologies follows.

Two-Way ranging (TWR) is a simple method, where the tag and anchors exchanges message in both ways. Thus the updated rate of the tag position is limited and decreased with a higher number of tags asking for their positions. The synchronization of the messages is provided with one anchor declared as an initiator of the network. The communication between the tag and anchors is illustrated in Figure 2.1. This method is the most used nowadays, as it is the simplest[2].

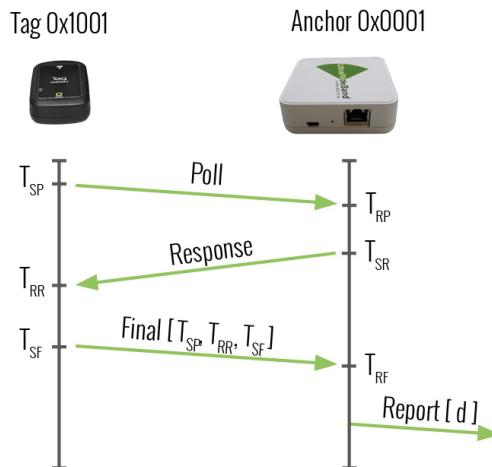


Figure 2.1. Two way ranging communication[3].

In **Time Difference of Arrival** (TDoA) technology, the tag only transmits and anchors only receive data. The tag sends a message to all available anchors. They estimate the time difference of delivered messages, and the tag's position is calculated according to that. With this technology, a higher update rate can be achieved even with more tags than TWR[2].

In **Reverse Time Difference of Arrival** (RTDoA), the tag listens and anchors transmit into the environment. The position is directly computed in the tag. This technology has no upper limit for tags and also promised the highest update rates. Thus it is well suited for flying drones or similar applications, where this high update rate and low latency are in need[2]. Unfortunately, this technology is not yet commonly available on the market.

Despite all mentioned benefits, in practice, the UWB localization faces errors caused by surrounding factors, leading to coordinate jitter or outliers[4].

2.2 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[5]. The unit is typically integrated with an onboard computational system 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 [5]. 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 2.2.

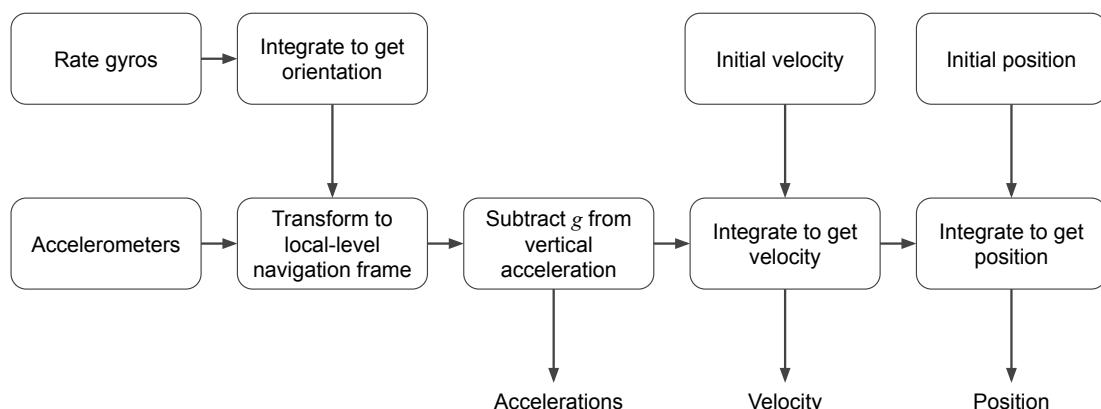


Figure 2.2. IMU block diagram[5].

There are two basic ways to mount the IMU to a vehicle, also called mechanization architectures [5–6].

- 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 transformation of the IMU body frame to the local frame. The conversion is closely related to inertial navigation systems described in Section 2.3.

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 the external environment [6], and it has a high data rate (100 Hz - 200 Hz). That makes the IMU measurement complementary to the UWB localization measurement.

2.2.1 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 [5].

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 the physical properties 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 [7]. MEMS accelerometers are then classified by converting the mechanical displacement of the proof mass to an electrical signal. The most common principles belong to piezoresistive, capacitive sensing, piezoelectric, optical sensing and tunnelling current sensing. Unlike the others, the piezoelectric MEMS sensors can not be used for navigation because their output rate is too low [7].

The current accelerometers used technology according to an application is summarized in Figure 2.3.

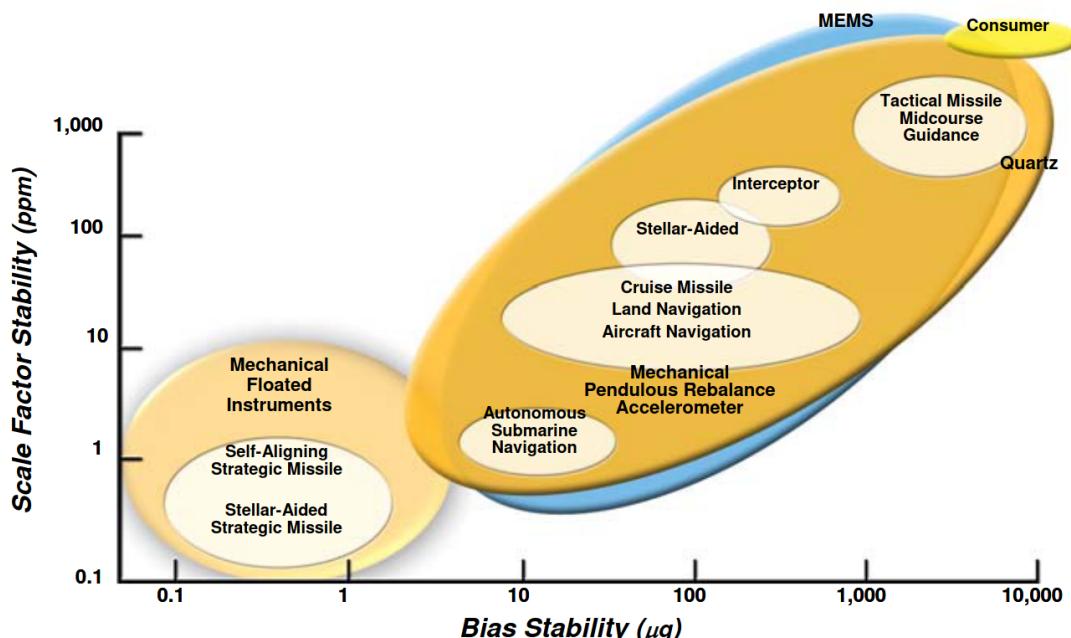


Figure 2.3. Accelerometers technology plotted by bias instability and scale factor stability[8].

2.2.2 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 three main categories of gyroscopes [5].

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, a 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. Typically, vibrating gyroscopes are based on MEMS technology[7], and they play a significant role in robotics because of their simplicity. They are small, cheap, have no rotating parts and have low power consumption.

The performance and application of each technology are demonstrated in Figure 2.4.

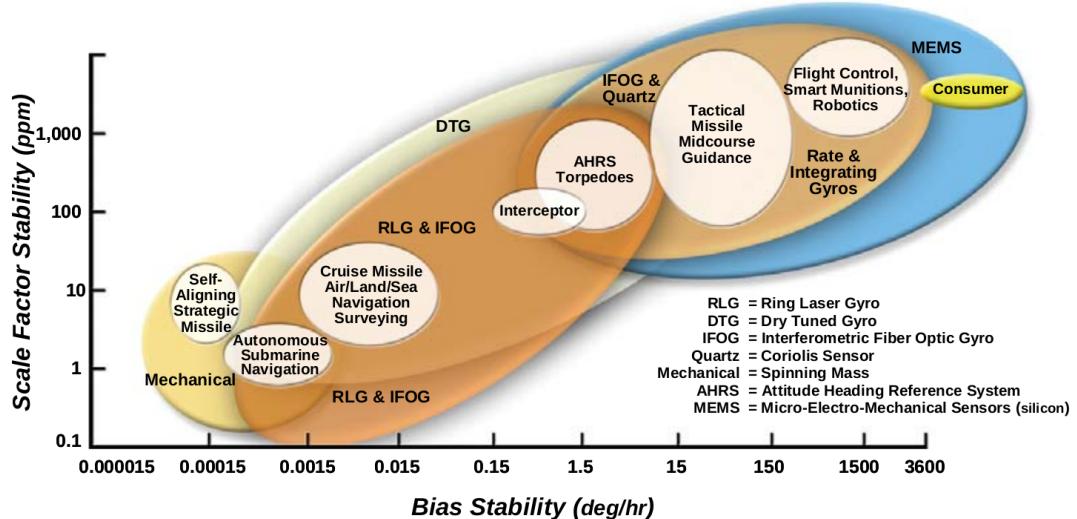


Figure 2.4. Gyroscopes technology plotted by bias instability and scale factor stability[8].

2.2.3 IMU's errors and Allan variance analysis

IMU errors IMUs faces several error sources, which are always related to the specific sample unit and its technology. In this thesis, the main focus is given to MEMS-based IMUs as they are used in experiments. These sensors are typically small and low cost. This section summarises the most significant errors for MEMS sensors and analysis these errors with Allan variance. This analysis is applied to a specific IMU, which is used during experiments.

Errors can be divided into two categories [7]

- stochastical errors, which can be described as random processes,
- and systematic errors are caused by manufacturing imperfections or not ideal handling with IMU. These errors can be corrected by proper calibration.

Nevertheless, errors need to be analyzed and reduced according to application requirements. Next following subset of errors is 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 instability¹, which represents drift of the sensor during a time,
- and initial bias² is a static offset, which can vary during each start-up of the device, but during a run, it is static.

¹ also called in-run bias

² repeatability bias

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 analyze a random error of inertial sensors in the time domain and is the most common time-domain measure of frequency stability. AVAR's brief introduction and important outcomes are given in [9].

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

$$\sigma_A^2(\tau) = \frac{1}{2(N-1)} \sum_{i=1}^{N-1} (\bar{y}_\tau(i+1) - \bar{y}_\tau(i))^2, \quad (3)$$

where N represents the number of clusters in the dataset ($N = \text{floor}(M/n)$), 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, $\bar{y}_\tau(i+1)$ and $\bar{y}_\tau(i)$ are mean values of certain cluster of $i+1$ -th and i -th cluster respectively. [10].

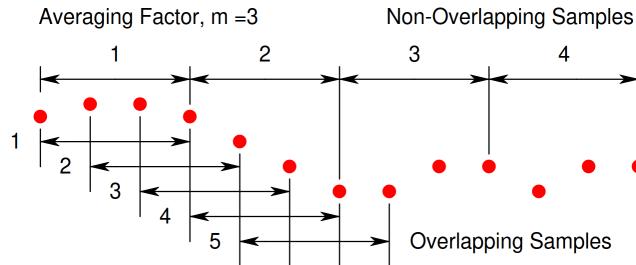


Figure 2.5. The difference between non-overlapping and overlapping sample [11].

The samples in a cluster can be both non-overlapping and overlapping. The difference is illustrated in 2.5. The overlapping samples improve the confidence of the resulting estimate. That is why this method is the most common for measuring time-domain frequency stability in general [9].

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 rates or accelerations.

If the dataset is valid and the AVAR is correctly computed, the plot copies the example plot seen in Figure 2.6. It is typically plotted on a log/log scale. A different slope of the graph describes each noise component by that the chart can be easily divided into specific parts.

The most significant outcome for navigation is when the bias instability is reached (slope is zero). At this time, the sensor model contains only a white (Gaussian) noise [13]. After that period, the external reset needs to be done.

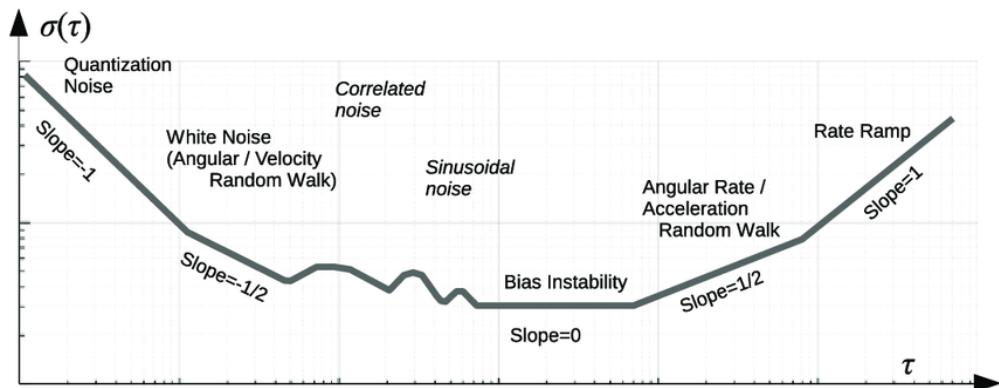


Figure 2.6. An example of Allan variance plot[12].

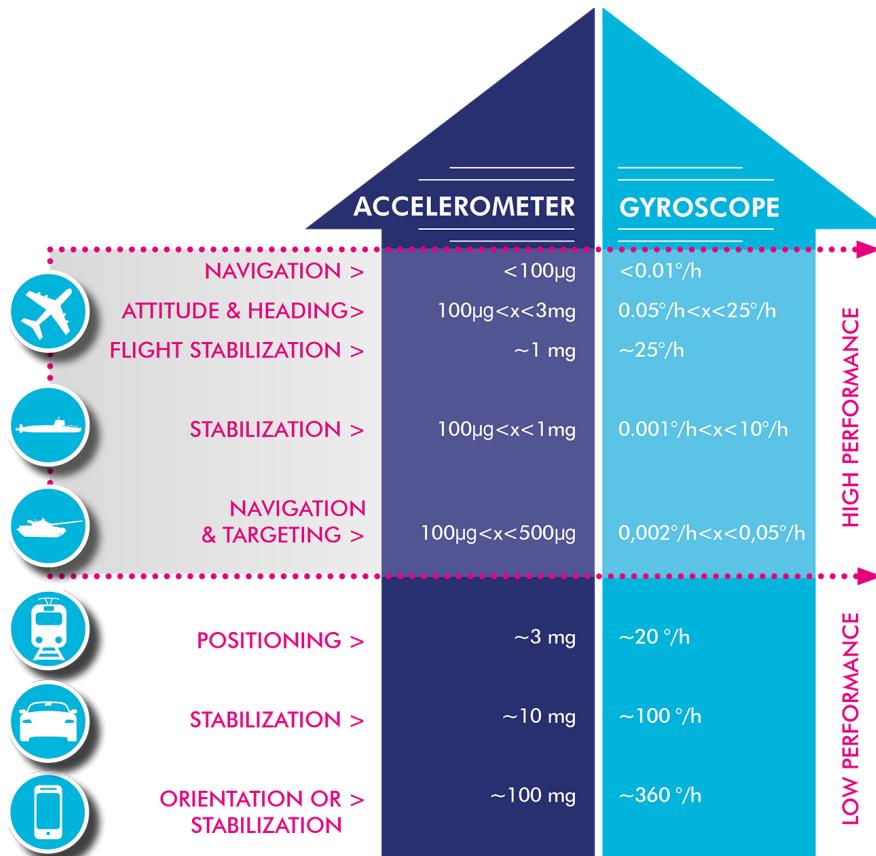


Figure 2.7. A performance of IMU per application[14].

2.2.4 Performance of IMUs according to their application

IMUS can be used in various applications, which differs by IMUs performance. Figure 2.7 summarises the overview of each sensor's precision for a given application is summarized in Figure 2.7.

2.3 Inertial navigation systems

The fundamental idea behind Inertial navigation systems (INS) is integrating a linear acceleration into a position. Because of that, this topic is closely connected with the IMU, as it measures current linear acceleration. The integration of IMU measurement is given by **navigation equations**.

As INS is typically used for navigation of aircraft, the principle of INS is introduced in that example. The plane is moving in a navigation coordinate frame. This frame can be specified as

- a local-level frame (as North-East-Down or East-North-Up),
- as a reference to a specific point at planet Earth
- or an Earth-fixed frame as ECEF [7], and these frames are demonstrated in Figure 2.8.

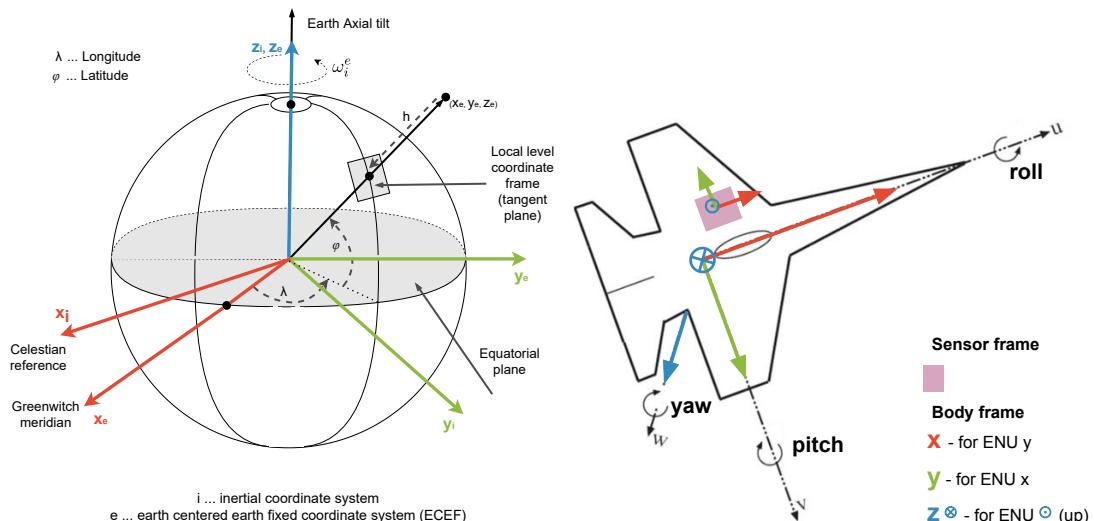


Figure 2.8. Example of coordinate frames used in INS³.

Figure 2.9. Example of body and sensor frame.

IMU sensor is mounted on an aircraft, and its gyroscopes and accelerometers measure in its sensor frame. The aircraft's frame is called body frame, so the measurement needs to be transformed into the body frame (see Figure 2.9). The output of the INS is given in the navigation frame. Thus the last transformation is from the body frame into a navigation frame.

Navigation equations implement the transforms between the sensor frame, the navigation frame and the integration of IMU measurements. For example, navigation signal processing in strap-down INS can be seen in Figure 2.10.

2.4 Odometry

Odometry is an example of a dead reckoning system. It estimates the pose and velocity of a device based on internal relative measurements of its motion. It can be obtained from various sources as IMU, lidars, cameras or wheel encoders[5].

Both IMU and wheel encoders are used in the suggested localization system since they can counter each other's negative characteristics because wheel encoders drift over travelled distance and IMU drift over time[5]. In this thesis, the term odometry

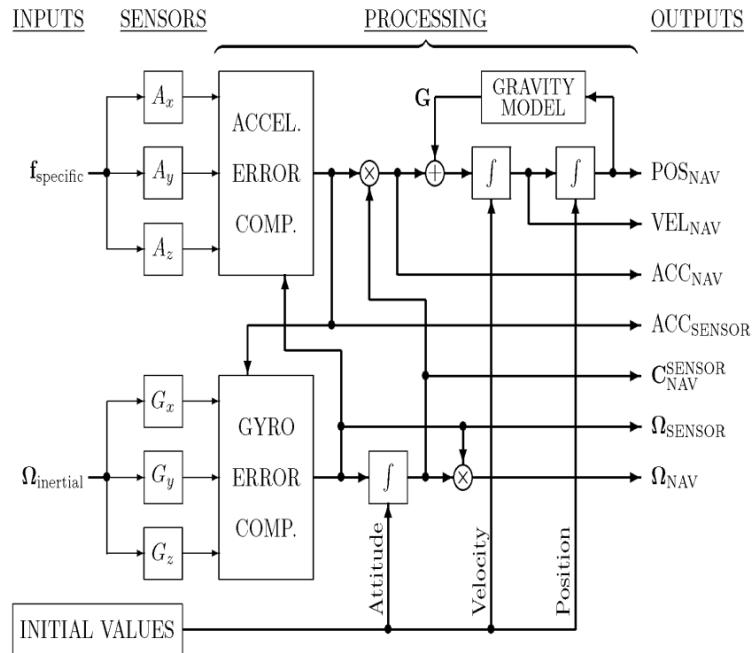


Figure 2.10. Schema of strapdown INS.[15].

is used as the odometry based on measurements from wheel encoders unless stated otherwise.

The details of odometry information vary with vehicle design, and during experiments, the differential type is used (illustrated in Figure 2.11).

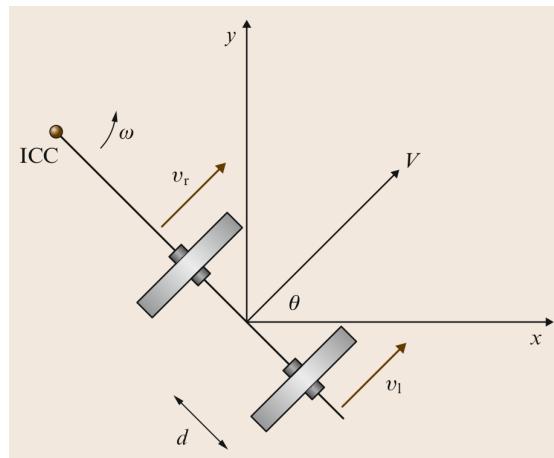


Figure 2.11. A differential drive kinematics scheme[5].

Chapter 3

Data fusion principles in pose estimation

The localization is estimating a robot's coordinates in an external reference frame from sensor data. It is called state. In the state estimation algorithms, more states from different sensors are fused to counter sensors negative characteristics and support the positive ones.

There are many approaches to data fusion. In particular, probabilistic methods are typically more robust in the face of sensor limitations, sensor noise or environment dynamics[16]. Moreover, they often scale much better to complex and unstructured environments, where the ability to handle uncertainty is of even greater importance[16]. Thus, this Chapter is dedicated to probabilistic methods for state estimation based on the Bayes filter[16].

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 introduce the benefits of using it, as it is the core of the localization system of this thesis.

3.1 Kalman and particle filter

Both Kalman and particle filter are the first implementations of Bayes filter in the continuous time[16], 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[16–17].

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[16].

There are a few limitations for both algorithms. For the Kalman filter, the state transitions and measurements must be linear with added Gaussian noise, and the initial state must have normal distribution[16]. There is no such requirement for linearity for the particle filter, and it works fine with nonlinear or multi-modal systems too[16]. 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[16].

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.

3.2 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. The linear system and sensor model assumptions with added Gaussian noise are rarely fulfilled in practice[16]. 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[16]. 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[16], but in practice, it is reliable for the system, which is almost linear in one-time step[18].

The Unscented Kalman filter is a tool that appears superior to the EKF linearization[16, 18]. Carefully selected sample points give the linearization from nonlinear functions. Also, this approach does not assume that the distribution of noise source is Gaussian[18].

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[19–21].

3.3 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[19].

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

$$x_t = x_n \bigoplus \delta x \quad (1)$$

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[20].

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[19–20]

$$\begin{aligned} x_t &= f(x_{t-1}, u_t) \\ P_t &= FP_{t-1}F^T + BQB^T \end{aligned} \quad (2)$$

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[19–20]

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

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[19–20]. The injection of error is given by

$$x_t = x_{t-1} + e_t \quad (4)$$

and the resetting can be illustrated

$$\begin{aligned} e_t &= g(e_t) \\ P_t &= G P_{t-1} G^T \end{aligned} \quad (5)$$

where

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

3.4 Chosen approach for the 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 Inertial navigation system(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 2.2). 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[19–20],
- The unscented Kalman filter or Particle filter algorithms have a high computational load that usually prevents them from being used in real-time system[22].
- The model for estimation of the error is near-linear, as the error is
 - close to the origin,
 - and small, so higher orders can be neglected[19–20],
- numerical stability of the solution[19],
- even if the error estimation has a temporary computer failure, the INS is not affected, and some emergency procedures can come into account[19].

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[19, 23, 21]. This approach is also known as Aided navigation systems and is briefly described in Aided navigation: GPS with high rate sensors[21]. The following Chapter 4 introduces the localization system design and its implementation in detail.

Chapter 4

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 proposes 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 detailed introduction to the proposed localization system with all equations and tools used in the final implementation.

4.1 System architecture design

Various approaches for state estimation were introduced in Chapter 3. The chosen approach is the Error state extended Kalman filter(ES-EKF). According to ES-EKF, the states' error is estimated using a Kalman filter rather than the state itself. The benefits of this approach are briefly summarized in Chapter 3.

The system consists of three crucial steps. The first is the inertial navigation system (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 4.1 and described in the following section in detail.

For navigation purposes, the usual rate of pose estimation is hundreds of Hertz[24]. The INS provides a full state estimate with the IMU update rate, which usually satisfies this requirement. Furthermore, it gives us the state estimation independent of external factors, such as slipping wheels[24].

UWB localization and odometry measurements cannot give us a much higher rate than tens Hertz, and 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 fast and straightforward. On the other hand, the error estimation can be more computationally demanding as the computation of Jacobians needs to be done. Therefore the

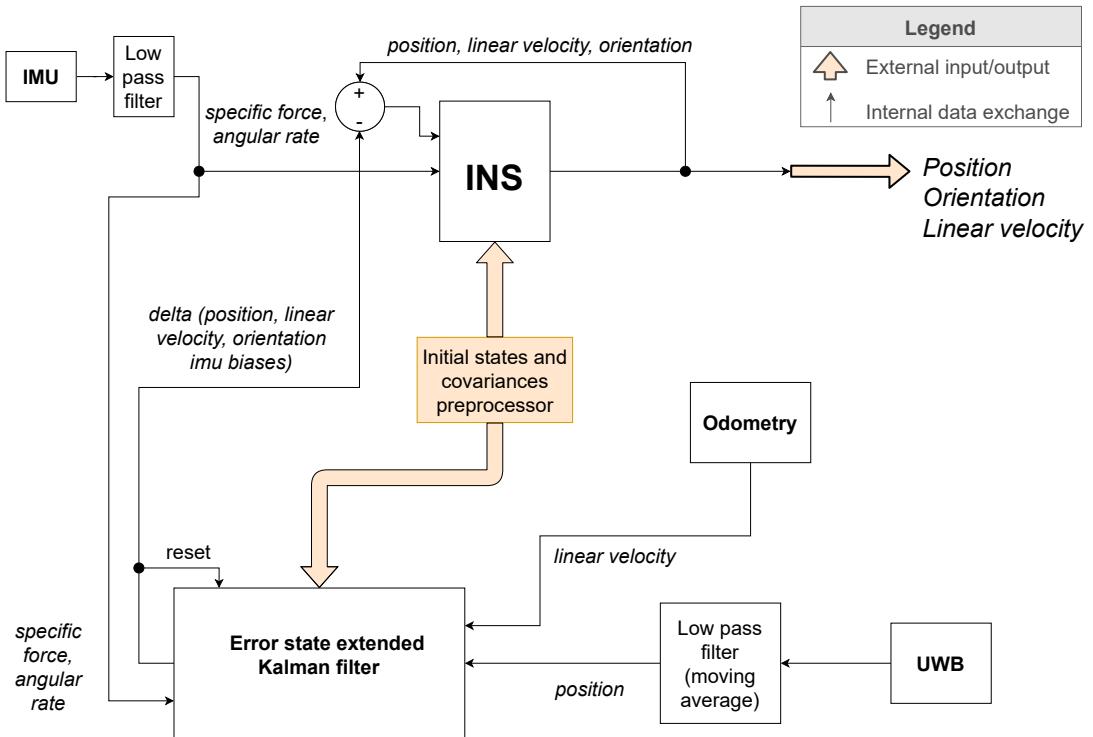


Figure 4.1. The proposed architecture of the localization system.

separation of state 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 must be applied before non-gaussian noise in IMU measurement is significant. That correction compensates 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).

4.2 System kinematics

For a more detailed introduction to system architecture, the system kinematics equations need to be announced. Equations are taken from [20], but instead of using GNSS, it uses UWB localization and linear velocity from odometry.

But before I enter that itself, let me describe an important topic, which represents attitude and rotation in 3D space.

4.2.1 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[5].

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[25–26]. Firstly, quaternions and rotation matrices do not suffer from singularities as Euler and fixed angles do[25]. Secondly, quaternion gives us a compact representation. And finally, these two are the most recommended representation in ROS standard rep-103[26]. As quaternions have many internal models in different libraries (Eigen library in C++[27], geometry messages library[28] or the transform library tf2 in ROS[29]) 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.

4.2.2 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 [20–21, 7, 6]. Therefore, I did not have to derive equations myself and used the one derived in [20] 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 [$m \cdot s^{-2}$],
- $\Omega_t = [(\omega_m - \omega_{bt} - \omega_n) \times] = \begin{bmatrix} 0 & -(\omega_{m3} - \omega_{bt3} - \omega_{n3}) & \omega_{m2} - \omega_{bt2} - \omega_{n2} \\ \omega_{m3} - \omega_{bt3} - \omega_{n3} & 0 & -(\omega_{m1} - \omega_{bt1} - \omega_{n1}) \\ -(\omega_{m2} - \omega_{bt2} - \omega_{n2}) & \omega_{m1} - \omega_{bt1} - \omega_{n1} & 0 \end{bmatrix}$ is true skew-symmetric 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

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

The output of the localization system is **navigation state** (also called nominal), which corresponds to the system kinematics, but does not take into account the noise terms w_t and other possible model imperfections (see equation 237 in [20], hence it is simplified to

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

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 [$m \cdot s^{-2}$],
- $\Omega = [(\omega_m - \omega_b)_{\times}] = \begin{bmatrix} 0 & -(\omega_{m3} - \omega_{b3}) & \omega_{m2} - \omega_{b2} \\ \omega_{m3} - \omega_{b3} & 0 & -(\omega_{m1} - \omega_{b1}) \\ -(\omega_{m2} - \omega_{b2}) & \omega_{m1} - \omega_{b1} & 0 \end{bmatrix}$ is skew-symmetric 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 [20]) of the **error state** are

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

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

$$y = h(x_t) + \rho, \quad (5)$$

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

$$y_1 = p_t + \rho_1, \quad (6)$$

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

$$y_2 = R_t^{-1}v_t + \rho_2 \quad (7)$$

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

4.2.3 The kinematics equations in discrete time

As the equations in continuous time are derived from book [20], 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 [20]. The equation 260c is slightly different since I am using rotation matrix for orientation representation and not quaternion. This equation is changed to

$$R \leftarrow R + (R\Omega)\Delta t, \quad (8)$$

where

- R is the rotation matrix of orientation,
- $\Omega = [(\omega_m - \omega_b)_{\times}] = \begin{bmatrix} 0 & -(\omega_{m3} - \omega_{b3}) & \omega_{m2} - \omega_{b2} \\ \omega_{m3} - \omega_{b3} & 0 & -(\omega_{m1} - \omega_{b1}) \\ -(\omega_{m2} - \omega_{b2}) & \omega_{m1} - \omega_{b1} & 0 \end{bmatrix}$ is skew-symmetric 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 4.1.

4.3 Error state extended Kalman filter implementation

Algorithm and equations for general extended Kalman filter are briefly described in Chapter 3. In this section, these equations are concretized.

The **error state system** is now

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

where

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

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

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

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 4.2.3,

$$F_x = \frac{\partial f(\delta x, u_m)}{\partial \delta x} = \begin{bmatrix} I & I\Delta t & 0 & 0 & 0 \\ 0 & I & -R[a_m - a_b] \times \Delta t & -R\Delta t & 0 \\ 0 & 0 & R\{\omega_m - \omega_b\}^T \Delta t & 0 & -I\Delta t \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (11)$$

F_i is given by

$$F_i = \frac{\partial f}{\partial i} \Big|_{x, u_m} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (12)$$

The covariances matrix is given by random impulses applied to the velocity, orientation and bias estimates, modelled by white Gaussian noise [20]

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

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 2.2.3).

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

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

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

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

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 [20]).

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 [20].

4.4 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

$$x \leftarrow x \bigoplus \delta x, \quad (16)$$

where

\bigoplus appropriate composition of sums or rotation product.

The equations are

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

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. The ES-EKF error reset operation is

$$\begin{aligned} \delta x &\leftarrow 0 \\ P &\leftarrow GPG^T \end{aligned} \quad (18)$$

where G is the Jacobian matrix defined as

$$G = \begin{bmatrix} I_6 & 0 & 0 \\ 0 & I - [\frac{1}{2}\delta\Theta]_x & 0 \\ 0 & 0 & I_9 \end{bmatrix}. \quad (19)$$

4.5 Implementation tools

This section briefly introduces tools used for implementation: the ROS2[30] framework, C++ and Python languages.

ROS2[30] 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, visualization, 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 distribution, such as 20.04 or 18.04.

The newest version of ROS is ROS2, introduced in 2014 at the conference ROSCon 2014 in Chicago[30], but the first distribution was released in May 2019. There are several distributions of ROS2 yet, the localization 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 localization needs to be implemented as a real-time application, it is implemented in C++17. Python3 is used for the visualization 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 5

Experiments

This chapter is dedicated to the experimental evaluation of the proposed localization system.

TODO dopsat pouzite metriky v experimentech a co nas vlastne zajima. - co nas zajima pro evaluaci lokalizace ve 2D prostoru - jake metriky budou pouzite (inspirace u Halodove)

The localization system was implemented on a robotic platform CART2, where an inertial measurement unit and ultra-wideband localization tag were mounted. Specific hardware and the robotic platform are introduced in the first section.

The following two sections are dedicated to experiments at two different experimental environments. The first experiments, described in the second section, were realized in a lab at Datavision s. r. o. company with an external localization system based on an AprilTag detection with the camera. The external localization system was developed in this thesis for the evaluation of the proposed localization system.

Additional experiments took place in an Intelligent and mobile robotics lab at Czech Technical University in Prague – Czech Institute of Informatics, Robotics, and Cybernetics, where the Vicon reference system is used for evaluation. These experiments are summarized in the third section of this chapter.

5.1 Used hardware description

This section presents an overview of used sensors for the proposed localization and a description of the CART2 robotic platform used during experiments.

5.1.1 Sensors specification

The onboard sensors of interest are the Inertial Measurement Unit, ultra-wideband localization tag, April tag and encoders on motors.

Inertial measurement unit used during experiments is Epson M-G365PDF1 (loaner sample). Epson M-G365 is used in various applications ranging from stabilization systems (as camera gimbal) to navigation systems.



Figure 5.1. IMU used in experiments Epson M-G365PDF1.



Figure 5.2. Experimental setup for static data acquisition of Epson M-G365PDF1.

The IMU has six degrees of freedom and measures angular rates and linear accelerations in three axis. It is factory calibrated, and the calibration data are stored in the memory of the unit. Technical specifications of Epson M-G365 can be found at ¹, while summary is included in Table 5.1

| Specification | Value |
|-----------------------------------|-----------------|
| Triple gyroscopes | ± 450 °/sec |
| Gyroscopes bias instability | 1.2 °/hr |
| Gyroscopes initial bias error | 0.1 °/s |
| Angular random walk | 0.08 °/√hr |
| Tri-axis accelerometers | ± 10 G |
| Accelerometers bias instability | 16 μ G |
| Accelerometers initial bias error | 3 mG |
| Velocity random walk | 0.033 (m/s)/√hr |

Table 5.1. Technical specifications of Epson M-G365PDF1[11].

As I already mentioned in Chapter 2.2.3, the AVAR analysis of IMU sensors can give us a brief overview of IMU's specifications. The experimental setup for static data acquisition can be seen in Figure 5.2. The sensor is mounted on two sponges and fixed with cardboard. The static data were recorded for 48 hours at a frequency of 30.0 Hz. The experiment took place at a village without any subway, trams or trains to reduce external vibrations on measurement (to reduce potential outliers) and at standard room temperature (about 23 °C).

For AVAR computation, I used a python library named AllanTools. For purposes of this thesis, the overlapping Allan deviation function is used.

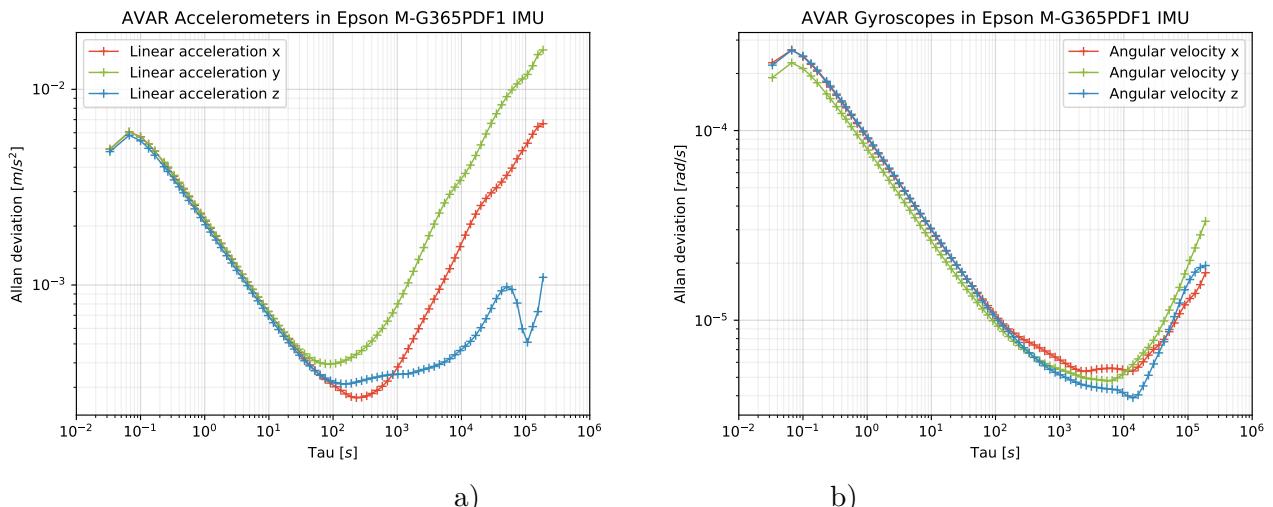


Figure 5.3. Overlapping Allan variance plot for Epson M-G365PDF1.

As shown in Figure 5.3, the external reset of integration of IMU measurement must be performed at 0.01 Hz at the latest. After that period, the non-gaussian noise of accelerometers comes into account. The UWB localization system works at 10 Hz, which should be enough for resetting.

¹ Datasheet of Epson M-G365 https://global.epson.com/products_and_drivers/sensing_system/download_hidden/pdf/m-g365pd_datasheet_e_rev20201007.pdf

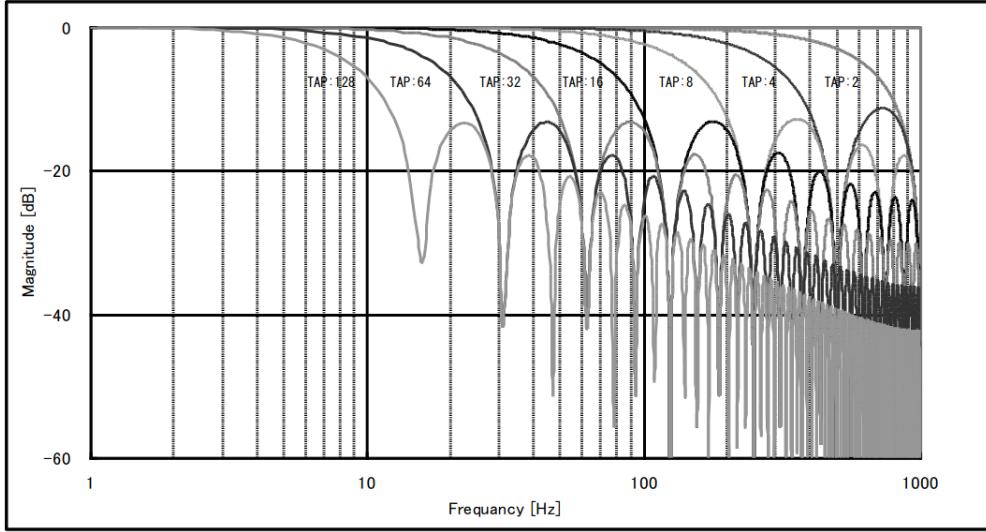


Figure 5.4. Moving average filter characteristics for Epson M-G365PDF1[11].

The IMU is set up for the final experiments to publish delta angle and delta velocity at 100 Hz with a moving average filter with tap 64. Because of that, higher frequencies than 10 Hz are filtered, as shown in Figure 5.4.

UWB localization system is provided by Qorvo's MDEK1001 ultra-wideband development kit. This kit includes 12 DWM1001-DEV development boards in plastic enclosures (see Figure 5.5).



Figure 5.5. DWM1001-DEV development boards[31].

Each can be configured as an anchor, tag or bridge node. The system is installed with six anchors fixedly mounted and one tag which is moving. The anchors are higher than the tag. According to the initiator anchor, the tag's position gives the coordinates frame's origin. The setup of the system can be seen in Figure 5.6.

The anchors should be mounted at the same height and higher than the operating area of the moving tag. They also should not be mounted close to any metal to get the best accuracy. There is a mobile application for the configuration of the network. The positions of the anchors are estimated manually and set in the network configuration.

Technical specification of MDEK1001 and DWM1001-DEV can be found in documents [31], the summary is listed in Table 5.2.

The odometry is computed according to the measurement of encoders in Maxon EPOS4 positioning controllers for Maxon brushless DC motors.

AprilTag serves as a global reference of the pose of the CART2 platform. The AprilTag detection software computes the precise 3D position, orientation, and iden-

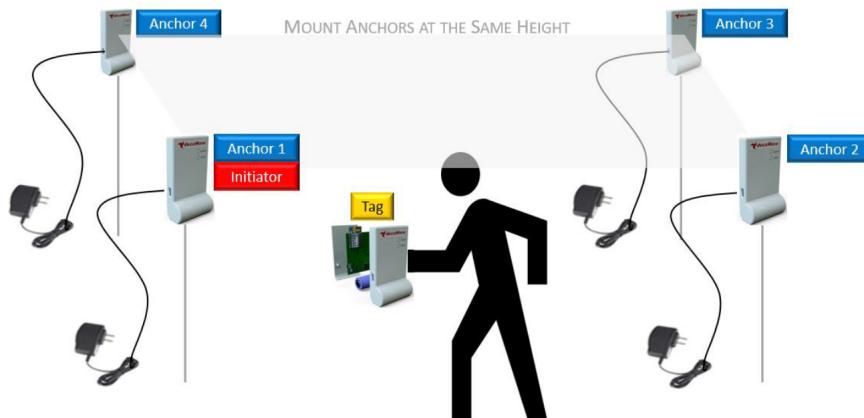


Figure 5.6. Positioning of anchors and tags[31].

| Specification | Value |
|---------------------------|--|
| Localization technology | Two-way ranging |
| Maximum tag location rate | 10 Hz |
| X-Y location accuracy | < 10 cm |
| Point to point range | up to 60 m in line of sight conditions |
| Scheme range | 25 - 30 m between anchors |

Table 5.2. System performance of MDEK1001[31].

ity of the tags relative to the camera[32]. This tag is similar to QR codes (a type of two-dimensional bar code), but it encodes smaller data payloads (between 4-12 bits), and it can be detected more robustly.

Camera used for detection of AprilTag is Niceboy Stream Pro with Full HD (1920 x 1080) resolution, 30 FPS, 90 ° field of vision and f/1.8 lens aperture[33].

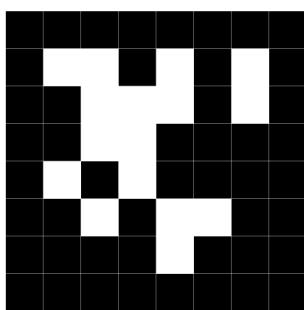


Figure 5.7. AprilTag used for detection of CART2 position at Datavision s. r. o.



Figure 5.8. Camera used for AprilTag detection Niceboy Stream Pro[33].

5.1.2 CART2 platform description and sensors placements

An image of the utilized CART2 platform can be seen in Figure 5.9. The coordinate frame of CART2 called `baselink` is illustrated in all Figures 5.9 and 5.10. The CART2 used for various robotic competitions is a differential drive equipped with the ADlink MXE-210 computer.

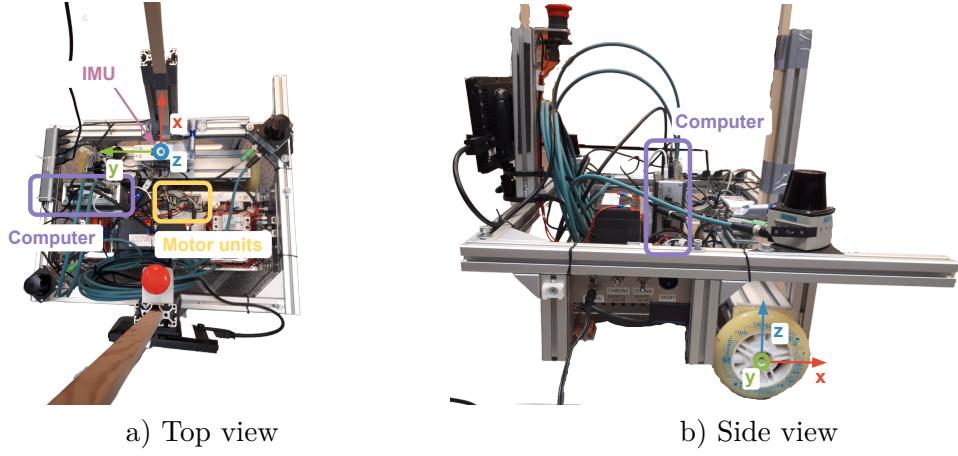


Figure 5.9. A photo of the utilized CART2 platform.

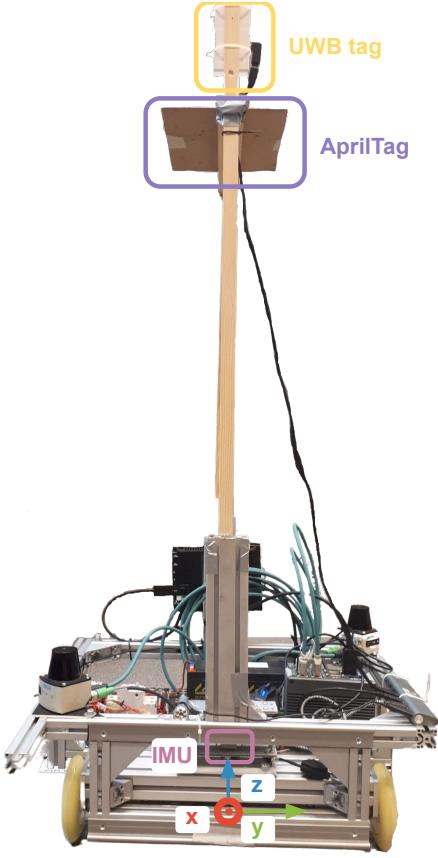


Figure 5.10. A photo of sensors placements on CART2.

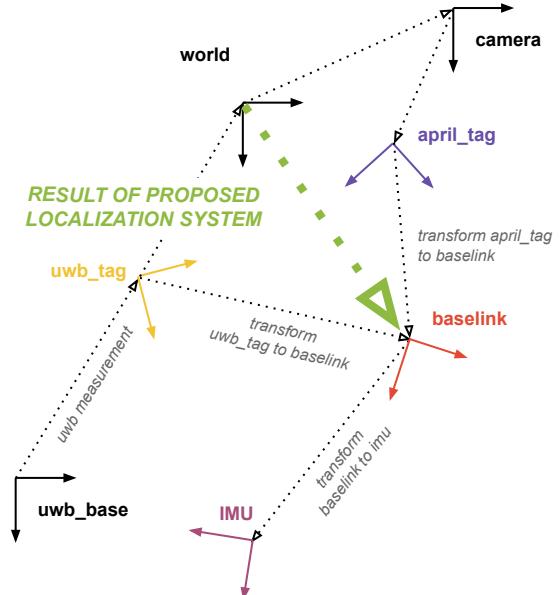


Figure 5.11. Illustration of coordinate systems at CART2.

The onboard sensors placements are illustrated in Figure 5.10. The vehicle uses Maxon brushless DC motor (Maxon EC-imotor) controlled by Maxon EPOS4 control units.

The UWB tag is mounted on a wooden stick approximately one meter upon the CART2 platform to reduce reflections of UWB waves from surfaces and the negative influence of any metal part. The AprilTag is mounted below the UWB tag not to become a barrier in the UWB wave. When the AprilTag was mounted on top of

the CART2 platform, the wooden stick obstructs the camera detection of AprilTag. Thus I decided to mount it as high as possible to reduce these situations.

Coordinate systems of CART2 and sensors placements are illustrated in Figure 5.11 and described in Table 5.3. Note, that world, camera and uwb_base frames are fixed with environment and explained in Section 5.2.

| Transform | Translation [x, y, z] in [m] | Rotation in quaternion [x, y, z, w] |
|----------------------|------------------------------|-------------------------------------|
| uwb_tag to baselink | [0.0, 0.0, 1.192] | [0.0, 0.0, 0.0, 1.0] |
| apriltag to baselink | [0.0, -0.171, 1.071] | [0.707107, -0.707107, 0.0, 0.0] |
| imu to baselink | [0.0, 0.0, 0.05] | [0.0, 0.0, -0.7071068, 0.7071068] |

Table 5.3. summary of transforms for experiments with CART2.

5.2 Experiments in the lab at Datavision s. r. o.

5.2.1 Experimental lab description

For the first experiments, I created an experimental setup at Datavision s. r. o. with global reference given by camera detection of AprilTag mounted at CART2 platform. The address of the building is Ukrajinská 1487/2a, 101 00 Prague 10 - Vršovice. Dimensions of the room for experiments are approximately 4 x 6 [m], and the camera view area is approximately 2.5 x 4.5 [m]. The setup can be seen in Figure 5.12.

Firstly, UWB anchors need to be mounted and measured their poses according to the world coordinate system. These poses need to be set in the mobile application for the configuration of the UWB network. The world frame coincides with the uwb_base frame in rotation and only differs in z coordinate in translation.

| Transform | Translation [x, y, z] in [m] | Rotation in quaternion [x, y, z, w] |
|-------------------|------------------------------|-------------------------------------|
| world to uwb_base | [0.0, 0.0, 2.58] | [0.0, 0.0, 0.0, 1.0] |
| world to camera | [1.089, 2.024, 2.625] | [0.707107, -0.707107, 0.0, 0.0] |

Table 5.4. Summary of transforms for experiments setup at Datavision s. r. o.

Secondly, the camera needs to be mounted and measured its position according to the world coordinate system. The transform between the world and the camera is not trivial. Because of that, I decided to use the transformation mentioned in Table 5.4 and then computed a camera homography according to a few positions measured by camera detection and by hand. The homography defines the transformation between a planar surface (ground) and a camera image plane. The camera homography is then applied to the AprilTag detection pose, which is considered a global reference. These frames are illustrated in 5.11.

5.2.2 Description of experiments

The area for experiments is not big, but I decided that these experiments serve as a proof of concept of the localization idea.

Experiments can be divided based on six simple trajectories of CART2 into

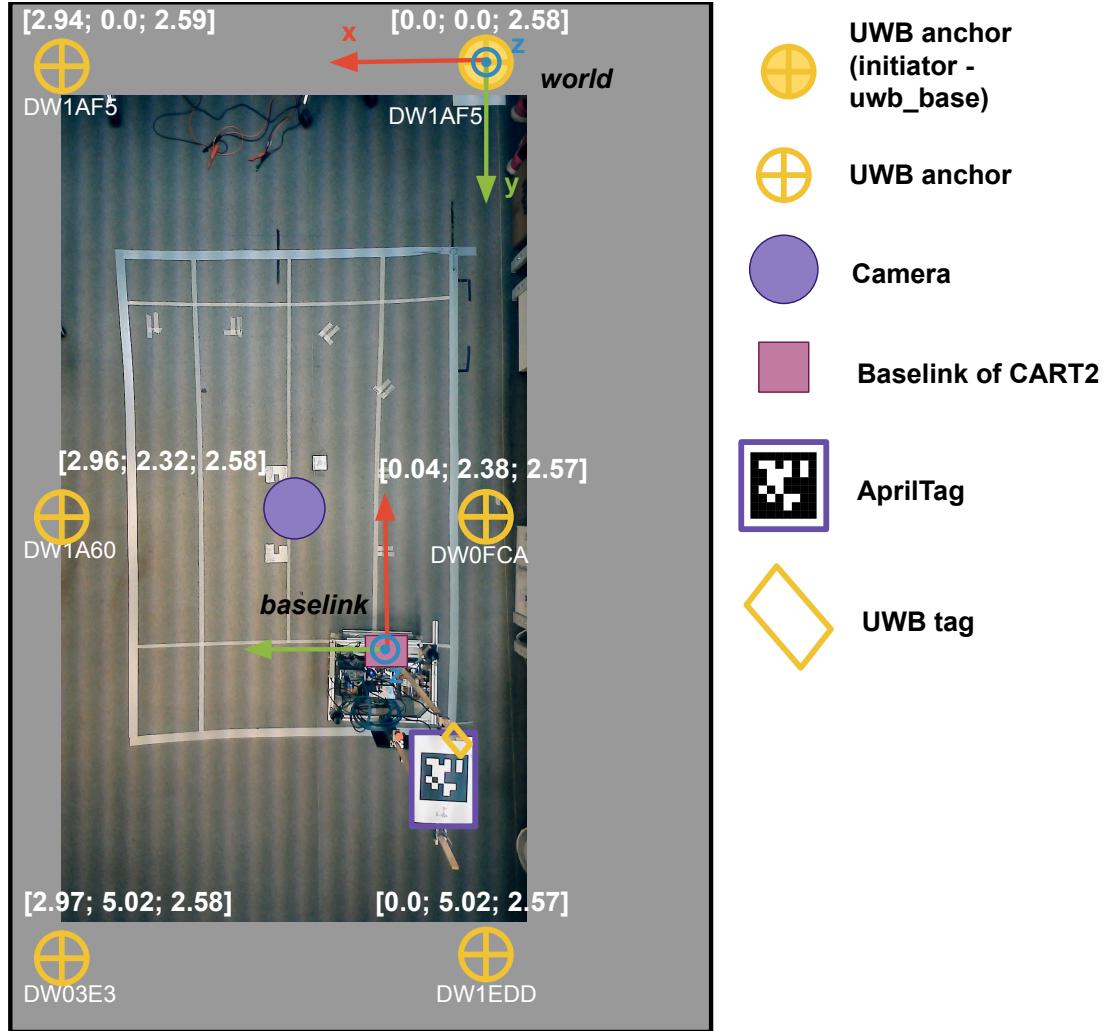


Figure 5.12. Experimental setup at Datavision s. r. o..

- stationary test (stationary test),
- rotation above 360 degrees in one direction at one place test (rotation test),
- moving in x direction (x test),
- moving in y direction (y test),
- moving in rectangle shape (rectangle test)
- and moving in infinity shape (infinity test).

I picked a specific initial pose for each testing trajectory to repeatedly make these tests in very similar conditions. Also, rotation, rectangle, and infinity tests ran several times in a row without start/stop of the system to see how it behaves in a long term.

Trajectories of these tests and the starting and ending positions of CART2 performing the movement are illustrated in Figure 5.13. CART2 platform was controlled via keyboard and joystick during experiments. The controlling via keyboard has the benefit of control velocity in each direction easily. Thus I used it for constant speed during simple moving in single-axis and simple rotation around a single axis. With that, I controlled all tests, except the infinity test, where the movement is complex. During the rectangle test, CART2 drove forward (moving in single-axis) then stopped and turned at one place above 90 degrees (rotating above single axis). I

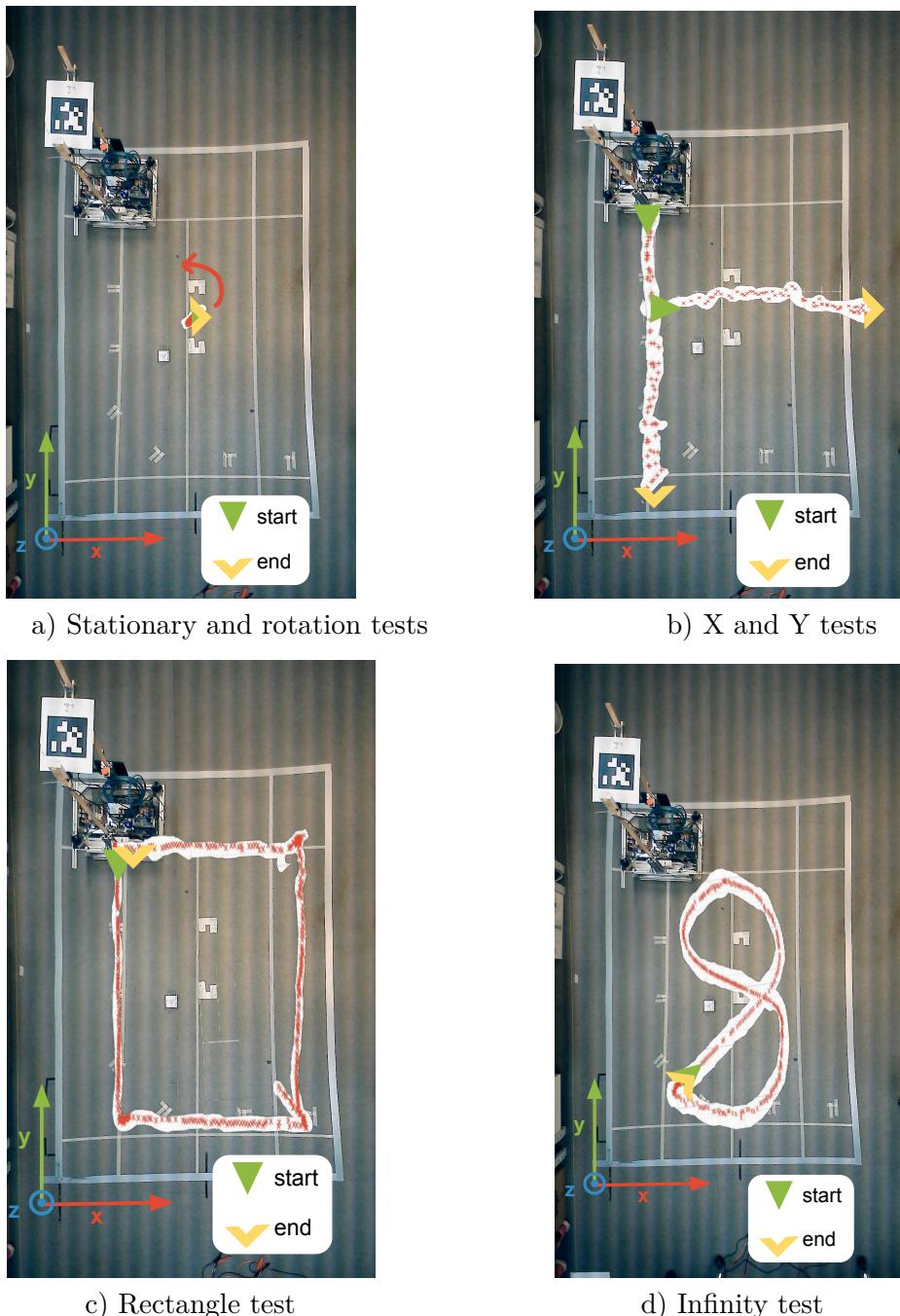


Figure 5.13. Trajectories for experiments at Datavision s. r. o..

controlled the CART2 platform with the joystick in infinity shape movement and tried to move similar velocities as in constant movements. The summary of velocities during experiments is given in Table 5.5.

5.2.3 Evaluation of experiments

TODO

| Test | Description | Speed [m/s] | Turn [rad/s] |
|---------------|-------------------------------------|----------------|----------------|
| stationary | Constant velocities | 0.0 | 0.0 |
| rotation | Constant velocities | 0.0 | 0.1094 |
| x | Constant velocities | 0.0750 | 0.0 |
| y | Constant velocities | 0.0750 | 0.0 |
| rectangle | Constant velocities | 0.0750 or 0.0 | 0.0 or 0.1094 |
| infinity test | non constant (control via joystick) | approx. 0.0750 | approx. 0.1094 |

Table 5.5. Velocities during experiments in Datavision s. r. o..

5.3 Experiments at CIIRC

5.3.1 Experiments at CIIRC description

The next experimental environment was placed at Czech Technical University in Prague – Czech Institute of Informatics, Robotics, and Cybernetics (CIIRC) in the Intelligent and mobile robotics lab. The localization system was evaluated using the VICON external camera localization system [34]. The VICON defines the world frame. UWB localization is installed with six anchors, and their positions are defined in a world frame. The setup is illustrated in Figure 5.14. The benefit of this environment is the size of the laboratory. The minimal distance between two anchors is bigger than 3,5 meters; thus, the accuracy of UWB localization should be increased.

Experiments here are more complex because they are longer both in terms of duration or distance. These experiments promised to evaluate possible drift caused by using two relative localization based on IMU and odometry. As I already mentioned in previous chapters, localization is based on pure IMU drift with time and odometry drift with travelled distance. Performed experiments can be divided based on three simple trajectories of CART2 into

- moving in the rectangle shape (rectangle test),
- moving in the infinity shape (infinity test)
- and moving in bean shape (bean test).

I picked a specific initial pose for each testing trajectory to repeatedly make these tests in very similar conditions. Also, rotation, rectangle, and infinity tests ran several times in a row without start/stop of the system to see how it behaves in a long term.

5.3.2 Evaluation of experiments at CIIRC

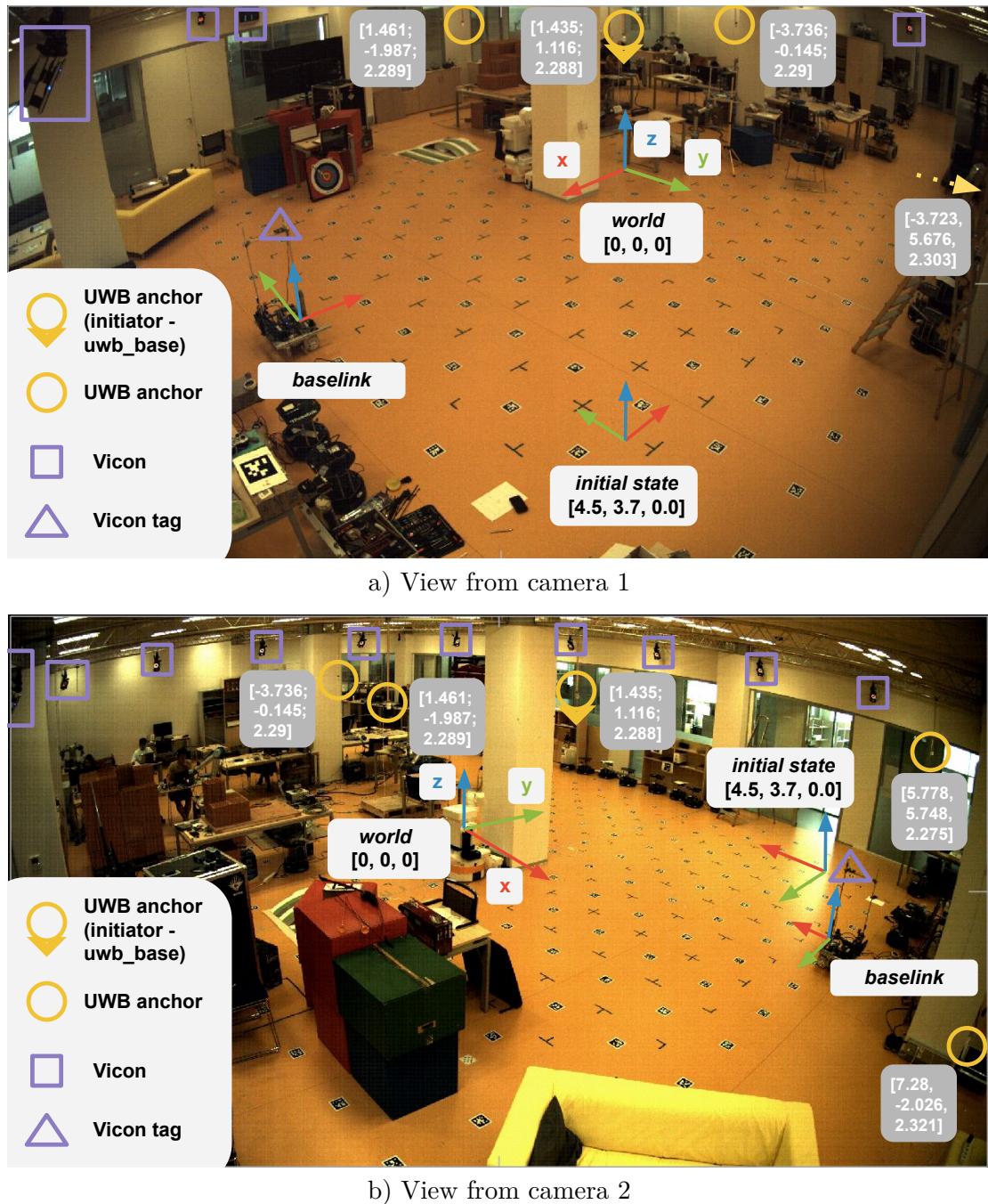


Figure 5.14. Experimental setup at Intelligent and mobile robotics lab (CIIRC).

Chapter 6

Conclusion and future work

Lorem ipsum sit amet

6.1 Mozne vyuziti teto fuze v prumyslu

6.2 Evaluace vysledku z experimentu celkova (discussion)

Lorem ipsum sit amet

6.3 Future work

- kalibrace sensoru poradna - pouziti virtualnich sensoru (robot stoji, robot jede jen rovne...) k tomu, aby estimace byla lepsi - jak to rozsirit pro fungovani na dronech
- lepsi hardware pro realtime aplikace - vyzkouseni lokalizace na vice robotech na jednou - presun do 3D - co se stane, kdyz kolem budou chodit lidi (human body is significant obstacle for UWB waves) - overeni v komplexnim scenariu, kde se na nejakou dobu ztrati signal UWB

References

- [1] V. Lakkundi. Ultra Wideband Communications: History, Evolution and Emergence. *Acta Polytechnica*. 2006, 46 DOI 10.14311/844.
- [2] Zafer Sahinoglu, Sinan Gezici, and Ismail Gvenc. *Ultra-Wideband Positioning Systems: Theoretical Limits, Ranging Algorithms, and Protocols*. USA: Cambridge University Press, 2011. ISBN 0521187834.
- [3] Sewio Networks. *Two way ranging illustration*.
<https://cdn.sewio.net/wp-content/uploads/2016/04/TWR-Scheme.jpg>.
- [4] Qigao Fan, Biwen Sun, Yan Sun, and Xiangpeng Zhuang. Performance Enhancement of MEMS-Based INS/UWB Integration for Indoor Navigation Applications. *IEEE Sensors Journal*. 2017, 17 (10), 3116-3130. DOI 10.1109/JSEN.2017.2689802.
- [5] Bruno Siciliano, and Oussama Khatib. *Springer Handbook of Robotics*. Springer-Verlag, 2007.
- [6] Peter Teunissen, and Oliver Montenbruck. *Springer handbook of global navigation satellite systems*. Springer, 2017.
- [7] Priyanka Aggarwal. *MEMS-based integrated navigation*. Artech House, 2010.
- [8] N. Barbour, and G. Schmidt. Inertial sensor technology trends. *IEEE Sensors Journal*. 2001, 1 (4), 332-339. DOI 10.1109/7361.983473.
- [9] William Riley, and David Howe. *Handbook of Frequency Stability Analysis*. 2008.
https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=50505.
- [10] Shuvra S Bhattacharyya, Ed F Deprettere, Rainer Leupers, and Jarmo Takala. *Handbook of signal processing systems*. Springer, 2018.
- [11] Seiko Epson Corporation. *Inertial Measurement Unit (IMU) : M-G365*.
https://global.epson.com/products_and_drivers/sensing_system imu/g365/.
- [12] Agnieszka Szczęsna, Przemysław Skurowski, Ewa Lach, Przemysław Pruszowski, Damian Pęszor, Marcin Paszkuta, Janusz Ślupik, Kamil Lebek, Mateusz Janiak, Andrzej Polanski, and Konrad Wojciechowski. Inertial Motion Capture Costume Design Study. *Sensors*. 2017, 17 612. DOI 10.3390/s17030612.
- [13] Wikipedia®. *Wikipedia - white noise definition*.
https://en.wikipedia.org/wiki/White_noise.
- [14] © 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.
- [15] *Inertial Navigation Systems*. In: *Global Positioning Systems, Inertial Navigation, and Integration*. John Wiley and Sons, Ltd, 2007. 9. ISBN 9780470099728.
<https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470099728.ch9>.

- [16] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. Cambridge, Mass.: MIT Press, 2005 . ISBN 0262201623 9780262201629.
- [17] 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.
- [18] 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>.
- [19] 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.
- [20] Joan Solà. Quaternion kinematics for the error-state KF. 2015,
- [21] Jay Farrell. *Aided navigation: GPS with high rate sensors*. McGraw-Hill, Inc., 2008.
- [22] Alessandro Benini, Adriano Mancini, and Sauro Longhi. An IMU/UWB/Vision-based Extended Kalman Filter for Mini-UAV Localization in Indoor Environment using 802.15.4a Wireless Sensor Network. *Journal of Intelligent and Robotic Systems*. 2013, 70 DOI 10.1007/s10846-012-9742-1.
- [23] 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.
- [24] 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.
- [25] M. D. Shuster. Survey of attitude representations. *Journal of the Astronautical Sciences*. 1993, 41 (4), 439-517.
- [26] Mike Purvis ROS, Tully Foote. *Standard Units of Measure and Coordinate Conventions*.
<https://www.ros.org/reps/rep-0103.html>.
- [27] 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.
- [28] Michel Hidalgo ROS, Tully Foote. *ROS geometry messages package*.
http://wiki.ros.org/geometry_msgs.
- [29] Tully Foote. *tf: The transform library*. Open-Source Software workshop. 2013.
- [30] 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>.
- [31] Inc © 2021 Qorvo. *Qorvo's MDEK1001 ultra-wideband (UWB) development kit*.
<https://www.qorvo.com/products/p/MDEK1001>.
- [32] Edwin Olson. *AprilTag: A robust and flexible visual fiducial system*. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011. 3400-3407.

-
- [33] Niceboy®. *niceboy ® STREAM PRO camera*.
<https://niceboy.eu/en/products/stream-pro>.
 - [34] Pierre Merriaux, Yohan Dupuis, Rémi Boutteau, Pascal Vasseur, and Xavier Savatier. A Study of Vicon System Positioning Performance. *Sensors*. 2017, 17 (7), DOI 10.3390/s17071591.

Appendix A

Abbreviations and symbols

A.1 A list of abbreviations

All abbreviations used in this thesis are listed below.

- AGV Autonomous ground vehicles.
- AVAR Allan variance.
- CIIRC Czech Institute of Informatics, Robotics and Cybernetics.
- CTU Czech Technical University in Prague.
- DOF Degrees of freedom.
- ES-EKF Error state Extended Kalman filter.
- GNSS Global Navigation Satellite System.
- IMU Inertial measurement unit.
- INS Inertial navigation system.
- MEMS Microelectromechanical systems.
- ROS2 Robot Operating System 2.
- TACR Technology Agency of the Czech Republic.
- TWR Two-way ranging.
- TDoA Time Difference of Arrival.
- RTDoA Reverse Time Difference of Arrival.
- UWB Ultra-wideband.