POLITECNICO DI TORINO

Master's Degree Course in Mechatronic Engineering

Master's Degree Thesis

UWB Anchors Self-Localization for Indoor Environments

Supervisors
Prof. Marcello Chiaberge
Dott. Ing. Giovanni Fantin

Candidate
Marco Ambrosio

July 2021

Contents

List of Tables

List of Figures

Abstract

Ultra-WideBand (UWB) is a well-known technology in the field of position localization since positioning with an error lower than 10 cm can usually be achieved. UWB is considered a valuable technology for service robotics since it allows the precise tracking of moving objects, e.g. UGVs and UAVs, in GPS denied zones and indoor environments. However, the setup for a UWB localization system is time-consuming and inefficient as each anchor's position must be accurately measured to obtain good performance from the system. This process may be tricky or impossible in harsh environments. Also, it is prone to artificial errors that may compromise the performance of the whole system. In this work, an algorithm for the self-localization of UWB anchors is proposed wherein the positions of the anchors are estimated by freely moving a ground vehicle (GV) equipped with a UWB tag. The proposed method is inspired by a Simultaneous Localization and Mapping (SLAM) technique used by the robotics community. An Extended Kalman Filter (EKF) estimates the anchors' position and improves the GV's odometry. Experimental tests demonstrate that anchors' positions are estimated with an error smaller than 20 cm, also in the presence of noisy distance measurements. Furthermore, EKF hugely improves GV's odometry since the error in the position does not increase with time, allowing precise and reliable localization of the tag.

Chapter 1

Introduction

UWB is a growing technology in the field of position localization. Indeed, it allows the localization of UWB tags with a relatively low expense and a high precision in harsh environments, also where GPS is not available.

The general setup for a UWB localization system is composed of four or more antennas placed in known positions, called anchors, and one or more moving tags which position must be identified. Surveying the position of each anchor is usually done by hand and requires additional equipment, such as laser distance meters. In addition to being a time-consuming and inefficient process, it is also impossible to be performed in places that are dangerous or can not be reached by humans.

This thesis faces the problem of anchors' positioning with the aim of finding a fast and reliable method to assign them coordinates and to set up a localization system. A method inspired by SLAM robotics community is proposed, in which a tag is placed on a moving robot and it is used to estimate the position of the fixed anchors. An EKF is used to estimate the positions of the anchors, the pose of the robot and their uncertainty. The proposed algorithm is implemented in ROS1 and tested in different situations in order to evaluate its performance.

In Chapter 2 an overview of UWB technology can be found. A brief history of the technology is provided alongside with a description of the current regulations. Also, details on its use for localization are provided.

In Chapter 3 an introduction to the EKF is provided. Moreover, its use for solving SLAM problems is analyzed. Finally, the algorithm for anchors' self-localization is proposed and described.

In Chapter 4 the experimental tests conducted for validating the algorithm are described. Instrumentation used during the tests is described. Obtained results are presented and critically analyzed.

Chapter 2

UWB for Localization

2.1 Historical overview

Despite what may be thought, UWB for radio communication is a relatively old technology. The first radio communication in history, realized by Guglielmo Marconi in the late XIX century, exploited the electromagnetic waves produced by a spark gap. Since it is very limited in time, this kind of event creates a wave spread over the frequency spectrum; thus, it can be categorized as UWB.

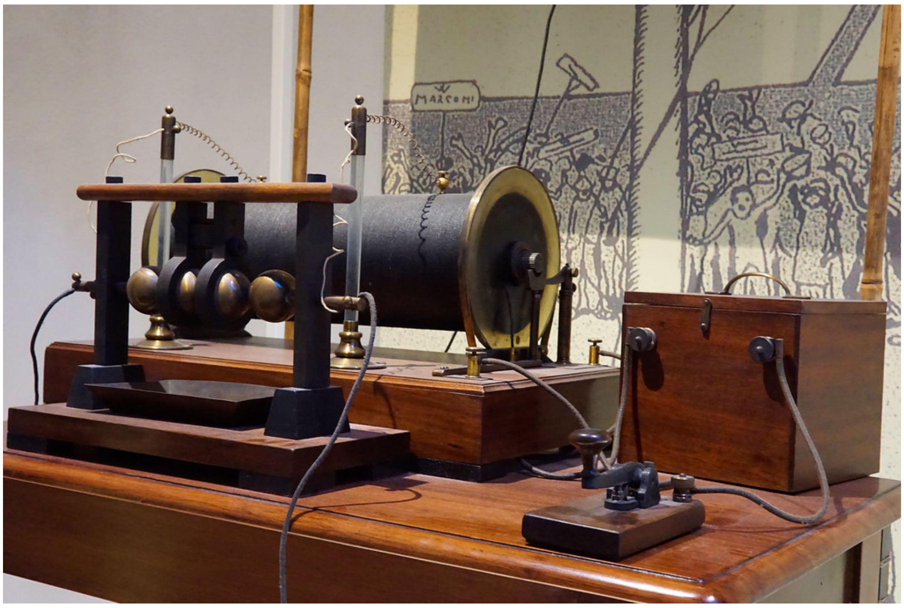


Figure 2.1: Marconi's device for radio signals transmission.

In the following decades, Marconi improved his radio devices, and they become to be quite common. However, the extensive use of these devices arouses the first problems. Spark gap transmitters were very energy-consuming. Moreover, the radio spectrum was largely occupied by spark gap transmissions. Without means of synchronization, two transmitters could not be used in the same geographical area without significant interference.

The development of vacuum tubes and transistors technologies quickly improved, making the continuous wave (CW) radio transmission more convenient. As a result, the spark gap transmitter was abandoned, and UWB technology was not of interest until the late '60s.

Between 1969 and 1984, H. F. Harmuth published papers and books exposing the basic theory behind UWB receivers and transmitters.

Almost in the same period, from 1972 to 1987, Ross and Robbins patented a series of devices capable of exploiting UWB, not only for communications but also for radar and sensing applications.

In 1989, this technology was referred to as "ultra wide-band" by the U.S. Department of Defense for the first time.

In 1994, the first low power application was developed by T. E. McEwan, who built the "Micropower Impulse Radar". A simple 9V battery operated this.

In parallel to the technical development, there was a rush for regulation and standardization of the technology. The first approvals for commercial use came in 2002 by the Federal Communication Commission (FCC). They approved the use of UWB signals in the spectrum between 3.1 and 10.6 GHz, with a power spectral density lower than -41.3 dBm/MHz.

In 2006, a commission from IEEE produced a regulation and a standard for UWB communication. They could not define the "de facto" standard for UWB physical layer since two major groups were opposed (UWB Forum proposes the direct-sequence UWB while WiMedia Alliance proposes MC-UWB). Nonetheless, the effort produced the following document, still used nowadays: "IEEE 802.15.4a UWB – Low-Rate Wireless Personal Area Networks (WPANs), Standard ECMA-368 High Rate Ultra Wideband PHY and MAC Standard, Standard ECMA-369 MAC-PHY Interface for ECMA-368, Standard ISO/IEC 26907:2007, Standard ISO/IEC 26908:2007". [1]

In 2019 Apple announced the first consumer electronic products embedding UWB technology.

2.2 Definition and Regulations

As defined in the IEEE 802.15.4a mentioned above, a UWB signal is either a signal with simultaneous bandwidth B greater or equal to 500 MHz or a signal with fractional (relative) bandwidth B_r larger than 20

$$B > 500\ MHz, \quad \text{or } B_r = \frac{2\ (f_u - f_l)}{f_u + f_l} > 20\% \tag{2.1}$$

where f_u and f_l are the upper and lower frequency at which the signal has a power spectral density 10 dB lower than its maximum.

Regulations for using UWB devices have been released in different countries and regulate several aspects, such as applications, maximum emissions level, allocated frequency ranges, and techniques to mitigate interferences.

As mentioned above, the U.S. FCC was the first authority worldwide that released regulations for UWB in February 2002. The allocated frequency range for indoor application is between 3.1 and 10.6 GHz, with emission levels shown in Figure 2.2.

Europe regulations came later, only in March 2006, and it allocates two different frequency range for indoor applications: 4.2-4.8 and 6-8.5 GHz. Also, in this case, emission levels are fixed and are shown in Figure 2.2.

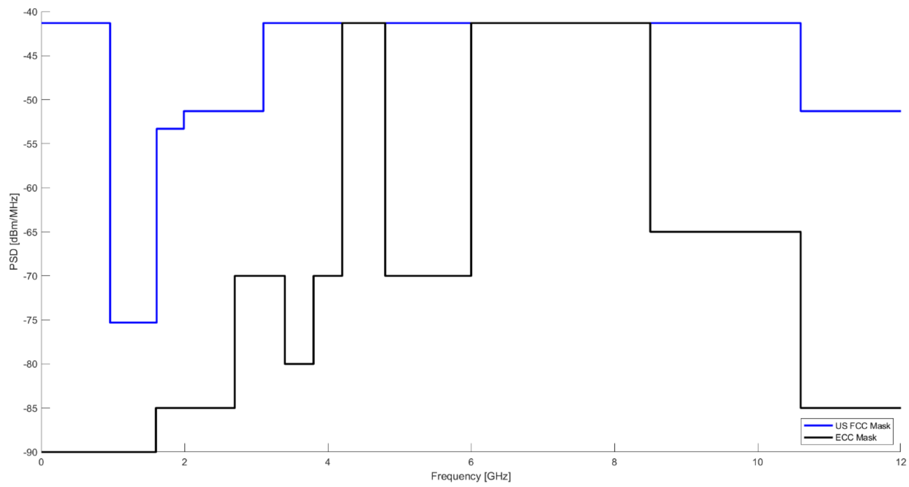


Figure 2.2: UWB spectral masks as defined in US FCC and ECC regulations. Numerical data reported in [2]

.

Other countries like Japan and China released rules regarding the use of UWB with two frequency ranges, similar to what happens in Europe.

In any case, for all the countries, the allowed transmitted power for UWB is minimal. For example, the FCC mask can be taken that allows a power spectral density of -41.3 dB/MHz for the whole mask, resulting in total transmitted power of 0.56 mW. The European regulations are even more strict in these terms. So it is clear that UWB applications can be considered only for short-range applications. [2]

2.3 Advantages

UWB technology offers a series of advantages that offers many possibilities for various applications. The large spectrum that can be used offers excellent flexibility in system design, allowing to adapt the system to specific needs. In fact, a variety of parameters can be changed to adapt data rate, range, power, and quality of service.

UWB is capable of high data rate transmission (> 1Gbps) over short ranges (less than 1 m). Nevertheless, a data rate can be traded-off for an increase in transmission distance. In the same way, data rate and range can be tuned to have a good compromise with power consumption.

Another advantage of UWB is the fine temporal resolution that allows robustness against multipath propagation and an excellent feature for ranging applications.

Since UWB signals span a vast frequency range, they show relatively low material penetration losses, allowing better linking a more comprehensive selection of situations. Moreover, multipath propagation can be distinguished and resolved since also minimal differences in time can be identified.

The other significant implication of good time resolution is the capability for ranging applications. Due to the extreme shot duration of pulses, ranging precision under 10 cm can be easily achieved. Moreover, IR-UWB technology requires small-sized and economical hardware, allowing its use in a wide variety of applications.

Finally, UWB is valuable for its small power usage and robustness to eavesdropping since UWB signals look like noise. [3]

2.4 IR-UWB vs. OFDM

The UWB spectrum mask can be exploited with two principal techniques. One is with very short pulses in the time domain, of the duration of hundreds of picoseconds to some nanoseconds. The pulse is radiated directly from the antenna, and a carrier is not required. This method is known as impulse radio (IR). The second method requires dividing the allocated bandwidth into multiple broadband channels, each one transmitted through a sub-carrier frequency. Each signal is modulated simultaneously to be orthogonal one to the other. In such a way, guard bands between channels are not required because the orthogonality avoids the cross-talk, and the bandwidth can be fully exploited. This second method is known as OFDM transmission. [4]

One significant advantage of the OFDM is that orthogonality between sub-carriers avoids cross-talk; thus, large guard bands can be avoided. This results in high spectral efficiency and optimal exploitation of the mask. Moreover, if some interferences occur, not all data may be lost because only some frequencies could be affected. On the contrary, if some interferences occur with IR, the whole pulse is disrupted.

OFDM could also be affected by the Doppler shift since a very accurate frequency synchronization is required between transmitter and receiver.

FFT and IFFT algorithms should be implemented on the receiver and the transmitter to achieve orthogonality in OFDM, requiring an additional energy cost. This also increases the complexity of the hardware and its cost. [2]

On the other side, IR technology offers good efficiency with simple architectures and analog components. The bandwidth is not exploited with the maximum efficiency, but it may be increased employing pulse shaping circuits. Moreover, it should be considered that the transmitted power allowed for UWB communication is really low, below 0.5 mW. Thus a low-power transmitter can be compatible with the expected applications of UWB. [5]

To summarize, IR-UWB advantages are simple architecture and pulses generated by simple analog components. A disadvantage is the limited bandwidth efficiency. On the other end, OFDM offers excellent bandwidth exploitation but with complex hardware and high power consumption.

2.5 Localization with UWB

Precise localization in indoor environments is quite a mandatory requirement for mobile robots. UWB, with its time resolution below the nanosecond, is capable of reaching ranging precisions up to 10 cm, allowing an accurate localization. [2], [6]–[8] Different techniques can be used to estimate the distance of two UWB antennas and, consequently, a mobile unit's position. Some of the most relevant are reported below.

Received Signal Strength (RSS) is one basic approach for range estimation. As one may imagine, a signal becomes weaker and weaker, going further from the signal's origin. This phenomenon can be exploited to have a rough estimate of the distance between a transmitter and a receiver, knowing the emitted power. Although it is straightforward to implement, this method suffers from poor accuracy, especially in indoor environments, where free space propagation can not be assumed. [1], [2], [7] One method to avoid such problems is to create a map of the environment, called fingerprint. Then the mobile object's position is retrieved by matching the RSS of several antennas to the previously constructed map. However, it is to highlight that this method is highly time-consuming and must be repeated in every new configuration of the localization system. [2]

Time of Arrival (TOA) is one of the most used techniques for UWB ranging applications. [8] A signal's arrival time is measured at the receiver. Then, the Time of Flight (TOF) can be computed since the received message contains the starting time. This method requires almost perfect clock synchronization between the sender and the receiver: even a microsecond misalignment can cause an error up to 300 m.

TOF measurement protocols have been introduced to avoid the problem of clock synchronization, such as Single-Sided Two Way Ranging (SS-TWR) or Double-Sided TWR (DS-TWR). In these protocols, only time measurement taken on the same device is compared, avoiding the necessity of clock synchronization. For the SS-TWR, the TOF is computed as:

$$T_{tof} = \frac{1}{2}\left(t_{round_A} - t_{reply_B}\right) \tag{2.2}$$

where $t_{round_A} = \tau_{ARx} - \tau_{ATx}$ is the actual round-trip time of a signal measured at Device A and $t_{reply_B} = \tau_{BTx} - \tau_{BRx}$ is the actual reply time of a signal measured at Device B. In the DS-TWR, the same measurements are performed on both sides, requiring a further reply message from the device which started the TWR. Two round-trip durations and two reply times are registered, and the TOF can be computed as:

$$T_{tof} = \frac{1}{4}\left[\left(t_{round_A} - t_{reply_B}\right) + \left(t_{round_B} - t_{reply_A}\right)\right] \tag{2.3}$$

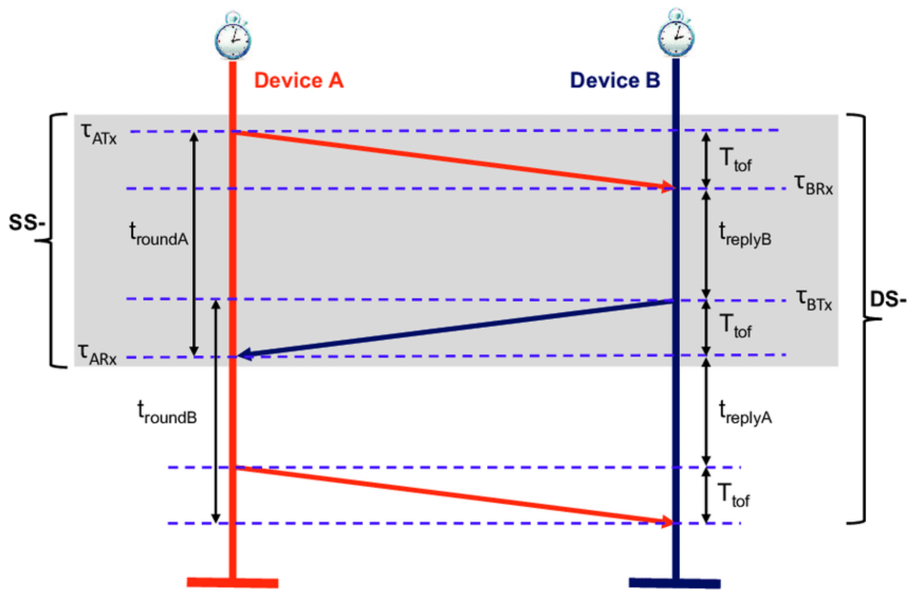


Figure 2.3: Illustration of single-and double-sided Two-Way Ranging (TWR) methods (© 2018 IEEE) Source: [9]

Even if the clock synchronization error is avoided, TWR is prone to errors such as Propagation-Time Delay, Transmission-Time Delay, Receiving-Time Delay, and Preamble Accumulation-Time Delay, which are all analyzed in deep in [9].

A widely used algorithm to retrieve a position given some distances to known points is trilateration. Ranges to at least three fixed anchors are required to uniquely determine the position of a mobile unit in 2D space. Also, the position of the fixed anchors must be known with precision to obtain good results. [1], [2] In Figure 2.4, it is represented a geometrical interpretation of trilateration. Ideally, it consists of finding the intersection of three circles. Actually, due to errors in distance measurements, the applied algorithms search for a point with the minimum error with respect to all the available ranges. [7]

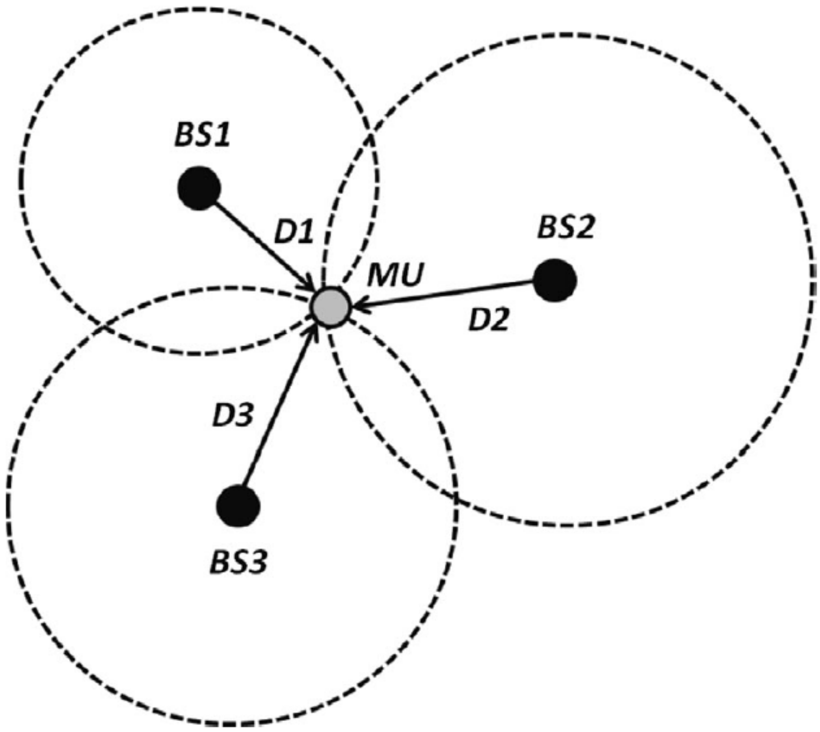


Figure 2.4: Geometrical interpretation of trilateration, given three distances. Source: [2]

Time Difference of Arrival (TDoA) is a variation of ToA that can be used to find the position of a transmitter when the synchronization between the mobile unit and the anchors can not be guaranteed. In this case, ranges are not directly measured, but the ToA of an emitted pulse is measured at different receivers, which position is known. [1], [2], [7] We have that

$$\begin{aligned} t_i &= \tau_i + t_M \\ &= \frac{d_i}{c} + t_M \\ &= \frac{\sqrt{(x_i - x)^2 + (y_i - y)^2}}{c} + t_M \end{aligned} \tag{2.4}$$

where t_i is the TOA at anchor i, t_M is the transmission time, andτ_i is the TOF of the signal.

Now, considering a second anchor, the transmission time of the signal can be excluded from the equation by taking the difference:

$$\begin{aligned} t_i - t_j &= \tau_i + t_M - (\tau_j + t_M) \\ &= \tau_i - \tau_j \\ &= \frac{\sqrt{(x_i - x)^2 + (y_i - y)^2} - \sqrt{(x_j - x)^2 + (y_j - y)^2}}{c} \end{aligned} \tag{2.5}$$

By introducing a third equation, using the TOA at a third anchor, the transmitter's position can be identified by making the intersection of the two hyperbolae.

$$t_i - t_k = \frac{\sqrt{(x_i - x)^2 + (y_i - y)^2} - \sqrt{(x_k - x)^2 + (y_k - y)^2}}{c} \tag{2.6}$$

The intersection can give one or two solutions; thus, a fourth equation is usually required to avoid ambiguities. Notice that the equation involving $t_j - t_k$ can not be used because it is linear dependent on the others.

Angle of Arrival (AoA) is a localization method that does not require distance measurements but only the angle of arrival of a signal. Usually, this kind of measure is more difficult to obtain and requires more complex hardware, such as an antenna array. The advantage of this method is that only two measurements are required to identify a position in a 2D configuration. In Figure 6, a geometrical representation of this method is provided. [1], [2],

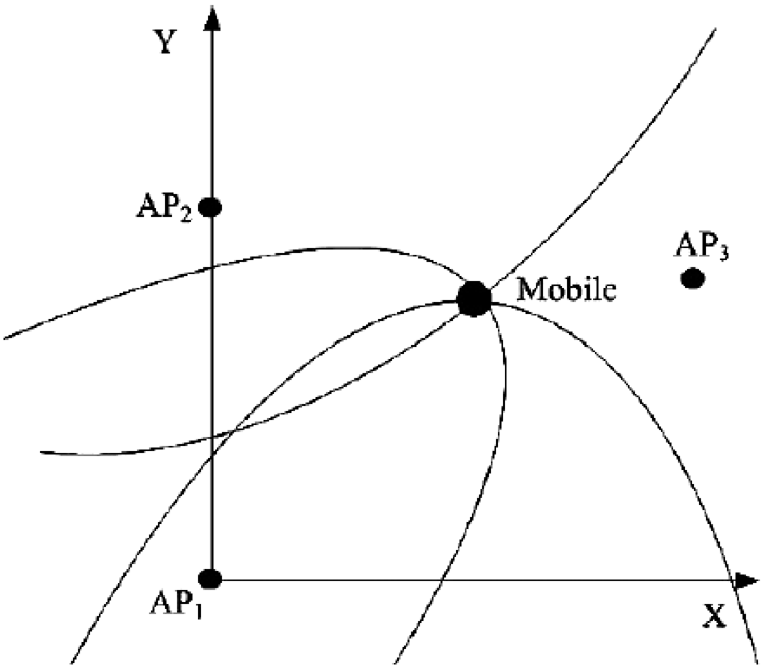


Figure 2.5: Trilateration using TDoA measurements. The position is at the intersection of the hyperbolae. Source: [1]

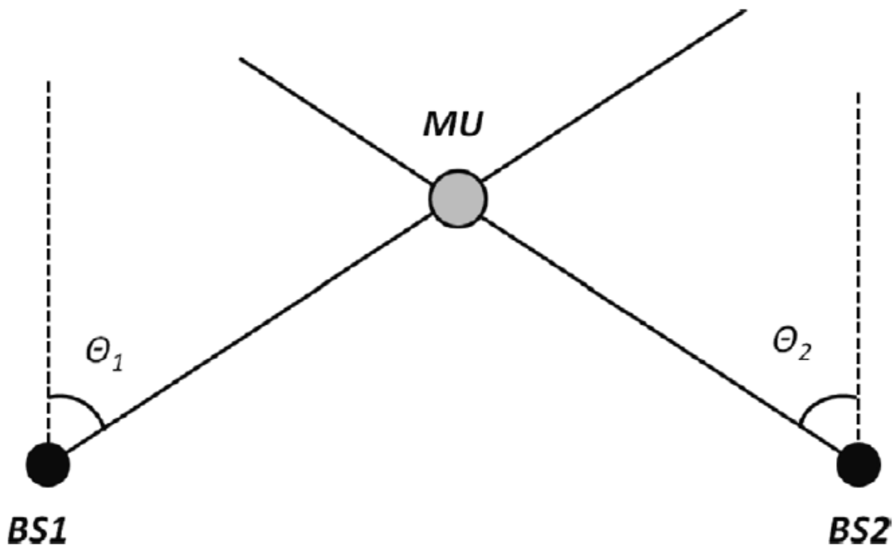


Figure 2.6: Positioning based on AoA measurements. Source: [2]

Chapter 3

Extended Kalman Filter

The Kalman Filter is one of the most well-known and often-used mathematical tools for stochastic estimation from noisy sensor measurements. The Kalman filter is essentially a set of mathematical equations that implement a predictor-corrector type estimator optimal in the sense that it minimizes the estimated error covariance when some presumed conditions are met. It is designed to manage discrete-time controlled processes that are governed by linear stochastic difference equations.

As one may imagine, most real-world phenomena are described by non-linear equations that the original Kalman filter can not manage. In order to adapt this mathematical tool to all these cases, linearization about the current mean and covariance can be performed. This kind of filter is referred to as Extended Kalman Filter (EKF).

3.1 Theoretical Overview

The KF addresses the general problem of trying to estimate the state $x \in R^n$ of a discrete-time controlled process that is governed by linear stochastic difference equations.

The KF consists basically of two operations performed in sequence: the time update step, in which the next state is predicted, and the measurement update step, where the predicted state is corrected based on some measurements.[10]

The general idea of KF can also be applied to problems governed by non-linear equations applying some modifications to the filter equations. The resulting filter is called EKF.

Let us assume that the process of interest is described by the state $x \in R^n$, and that it is governed by the non-linear stochastic difference equation

$$x_k = f\left(x_{k-1}, u_k, w_{k-1}\right) \ , \tag{3.1}$$

with a measurement $z \in R^m$ that is

$$z_k = h\left(x_k, v_k\right) \ , \tag{3.2}$$

where the random variables w_k and v_k represent respectively the process and measurement noise.

The process and measurement noises are assumed to be independent of each other, white, and with normal probability distribution.

$$p\left(w\right) \sim N\left(0, \mathbf{Q}\right) \tag{3.3}$$

$$p\left(v\right) \sim N\left(0, \mathbf{R}\right) \tag{3.4}$$

Q and R represent the process noise covariance matrix and the measurement noise covariance matrix. The non-linear function f relates the state at the previous time step $k-1$ to the state at the current time step k. The function includes as parameters any driving function u_k and the zero-mean process noise w_k. The non-linear function h relates the current state x_k to the measurement z_k. Because one does not know individual values of noise w_k and v_k at each time step, the state and measurement vectors are approximated without them

$$\widetilde{x}_k = f\left(\hat{x}_{k-1},\ u_k,\ 0\right) \tag{3.5}$$

$$\widetilde{z}_k = h\left(\widetilde{x}_k,\ 0\right) \tag{3.6}$$

where $\widetilde{x}_k$ is an a posteriori estimate of the state from a previous time step.

The actual state and measurement vector can now be approximated using the following linearized equations:

$$x_k \approx \widetilde{x}_k + \mathbf{G}\left(x_{k-1} - \hat{x}_{k-1}\right) + \mathbf{W}w_{k-1} \tag{3.7}$$

$$z_k \approx \widetilde{z}_k + \mathbf{H}\left(x_k - \widetilde{x}_k\right) + \mathbf{V}v_k \tag{3.8}$$

Where

- x_k and z_k are the actual state and measurement vectors,
- $\widetilde{x}_k and \widetilde{z}_k$ are the approximate state and measurement vectors from the equations above,
- $\hat{x}_k$ is the a posteriori estimate of the state at time k
- $\mathbf{G}$ is the Jacobian matrix of f with respect to the state vector x

$$\mathbf{G_{i,j}} = \frac{\partial f_i\left(\hat{x}_{k-1},\ u_k,\ 0\right)}{\partial x_j} \tag{3.9}$$

- $\mathbf{W}$ is the Jacobian matrix of f with respect to the input noise w

$$\mathbf{W_{i,j}} = \frac{\partial f_i\left(\hat{x}_{k-1},\ u_k,\ 0\right)}{\partial w_j} \tag{3.10}$$

- **H** is the Jacobian matrix of h with respect to the state vector x

$$\mathbf{H_{i,j}} = \frac{\partial h_i\left(\widetilde{x}_k,\ 0\right)}{\partial x_j} \tag{3.11}$$

- **V** is the Jacobian matrix of h with respect to the measurement noise v

$$\mathbf{V_{i,j}} = \frac{\partial h_i\left(\widetilde{x}_k,\ 0\right)}{\partial v_j} \tag{3.12}$$

The equations of the EKF are reported in Tables 3.1 and 3.2. For demonstrations on how they are obtained, further details can be found on [10].

3.2 SLAM with EKF

Simultaneous localization and mapping (SLAM) is the problem of concurrently estimating in real-time the structure of the surrounding world (the map), perceived by moving exteroceptive sensors, while simultaneously localizing the agent in it. While it may seem like a chicken-and-egg problem, several solutions have been found, at least in an approximated and probabilistic way. Indeed, EKF is one of the possible approaches that can be used to estimate the map and its uncertainty. [11], [12]

SLAM always involves a moving agent capable of taking measurements of the surrounding environment with exteroceptive sensors (for example, distance measurements to the UWB anchors). Optionally, the moving agent may be equipped with proprioceptive sensors that provide information on its own movement (e.g., wheel encoders, accelerometers, gyroscopes).

A SLAM algorithm's operations are essentially three: moving in the environment, discovering new landmarks, and observing known landmarks.

While the robot moves, its uncertainty in localization increases due to noise and errors. The mathematical model needed to describe the phenomenon is called the motion model. The robot, moving around, can measure both a known or an unknown landmark. The direct observation model is applied in the first case, and the overall uncertainty will be reduced. Otherwise, a new landmark will be added to the map with its initial uncertainty through the inverse observation model if it has not been observed before. [12]

The map is represented through the state vector x of the problem, and it is composed of the robot's states and the position of the landmarks.

$$x = \begin{bmatrix} \mathcal{R} \\ \mathcal{M} \end{bmatrix} = \begin{bmatrix} \mathcal{R} \\ \mathcal{L}_1 \\ \vdots \\ \mathcal{L}_\backslash \end{bmatrix} \tag{3.19}$$

Time Update
1) Project the state ahead in time
$$\bar{x}_k = f\ \left(\hat{x}_{k-1},\ u_k,\ 0\right) \tag{3.13}$$
2) Project the error covariance ahead in time
$$\mathbf{\bar{P}_k = G_k P_{k-1} G_k^T + W_k Q_{k-1} W_k^T} \tag{3.14}$$

Table 3.1: General time update equations of the EKF

Measurement Update
1) Compute Kalman Gain
$$\mathbf{Z_k = H_k \bar{P}_k H_k^T + V_k R_k V_k^T} \tag{3.15}$$ $$\mathbf{K_k = \bar{P}_k H_k^T Z_k^{-1}} \tag{3.16}$$
2) Update estimate with measurement z_k
$$\hat{x}_k = \bar{x}_k + \mathbf{K}\left(z_k - h\left(\bar{x}_k, 0\right)\right) \tag{3.17}$$
3) Update the error covariance
$$\mathbf{P_k = \left(I - K_k H_k\right) \bar{P}_k} \tag{3.18}$$

Table 3.2: General measurement update equation of the EKF

In the EKF, the map is represented by a Gaussian variable through the state vector's mean and the covariance matrix, identified respectively as $\bar{x}$ and $\mathbf{P}$.

$$\bar{x} = \begin{bmatrix} \bar{\mathcal{R}} \\ \bar{\mathcal{M}} \end{bmatrix}, \qquad \mathbf{P} = \begin{bmatrix} \mathbf{P}_{\mathcal{R}\mathcal{R}} & \mathbf{P}_{\mathcal{R}\mathcal{M}} \\ \mathbf{P}_{\mathcal{M}\mathcal{R}} & \mathbf{P}_{\mathcal{M}\mathcal{M}} \end{bmatrix} \tag{3.20}$$

The initial state vector is composed only of the robot's state since any landmark is unknown from the start. Moreover, the initial position of the robot is known with absolutely no uncertainty owing to the fact that it is considered the origin of the map. Therefore,

$$\bar{x}_0 = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}, \quad \mathbf{P_0} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.21}$$

Robot Motion. When the robot performs a movement, the EKF is updated following the time update equations reported in Table 3.1. In a regular EKF, the whole state is updated every time, but in an EKF for SLAM, the only part of the state that changes in time is the robot state $\mathcal{R}$. Hence, the time update operation can be split as follow:

$$\begin{aligned} \mathcal{R} &\leftarrow f\left(x,\ u, w\right) \\ \mathcal{M} &\leftarrow \mathcal{M} \end{aligned} \tag{3.22}$$

where $f\left(x,\ u, w\right)$ is the motion model of the robot. A large part of the state is time-invariant, so as a consequence also the Jacobian matrices $\mathbf{G}$ and $\mathbf{W}$ will be sparse.

$$\mathbf{G} = \begin{bmatrix} \frac{\partial f}{\partial \mathcal{R}} & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} \mathbf{G}_{\mathcal{R}} & 0 \\ 0 & \mathbf{I} \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} \frac{\partial f}{\partial w} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{\mathcal{R}} \\ 0 \end{bmatrix} \tag{3.23}$$

To optimize the computational effort required to run the EKF, all trivial operations, such as multiplication by 1 or 0, are avoided. [11] The resulting time update equations are reported below:

$$\bar{\mathcal{R}} \leftarrow f\left(\bar{\mathcal{R}}, u, 0\right) \tag{3.24}$$

$$\mathbf{P}_{\mathcal{R}\mathcal{R}} \leftarrow \mathbf{G}_{\mathcal{R}} \mathbf{P}_{\mathcal{R}\mathcal{R}} \mathbf{G}_{\mathcal{R}}^{\top} + \mathbf{W}_{\mathcal{R}} \mathbf{Q} \mathbf{W}_{\mathcal{R}}^{\top} \tag{3.25}$$

$$\mathbf{P}_{\mathcal{R}\mathcal{M}} \leftarrow \mathbf{G}_{\mathcal{R}} \mathbf{P}_{\mathcal{R}\mathcal{M}} \tag{3.26}$$

$$\mathbf{P}_{\mathcal{M}\mathcal{R}} \leftarrow \mathbf{P}_{\mathcal{R}\mathcal{M}}^{\top} \tag{3.27}$$

Landmark observation. Whenever measurements from sensors are available, the measurement update step is performed as reported in Table 3.2. However, some changes can be applied to reduce the computational complexity. In particular, the computation of the innovation $\mathbf{Z}$ is sparse because it only involves the observed landmark and the robot state. The general observation function model is represented as:

$$y_i = h_i\left(\bar{\mathcal{R}}, \mathcal{S}, \bar{\mathcal{L}}_i\right) + v \tag{3.28}$$

where $\mathcal{S}$ is the sensor's state and v is the measurement noise.

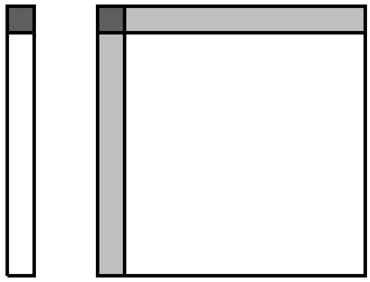


Figure 3.1: Representation of the update part of the map upon robot motion. The bar on the left represents the mean $\bar{x}$, while the square represent the covariance matrix $\mathbf{P}$. The grey part is the only part of the map updated: the robot's state mean $\bar{\mathcal{R}}$, its covariance $\mathbf{P}_{\mathcal{R}\mathcal{R}}$ (in dark grey) and the cross covariances with the landmarks $\mathbf{P}_{\mathcal{R}\mathcal{M}}$ and $\mathbf{P}_{\mathcal{M}\mathcal{R}}$ (in light grey). The rest of the map is time-invariant and is not updated upon robot motion. Source: [11]

Therefore, the Jacobian $\mathbf{H}$ is structured sparsely:

$$\begin{aligned}\mathbf{H} &= \begin{bmatrix}\frac{\partial h_i}{\partial \mathcal{R}} & 0 & \cdots & 0 & \frac{\partial h_i}{\partial \mathcal{L}_i} & \cdots & 0\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{H}_{\mathcal{R}} & 0 & \cdots & 0 & \mathbf{H}_{\mathcal{L}_\mathbf{i}} & \cdots & 0\end{bmatrix}\end{aligned} \tag{3.29}$$

The computation of the innovation (Equation (3.15)) thus can be reduced only to the non-zero elements, resulting in a reduced number of operations. [11] The updated set of equations for SLAM-EKF is the following

$$\mathbf{Z} = \begin{bmatrix}\mathbf{H}_{\mathcal{R}} & \mathbf{H}_{\mathcal{L}_\mathbf{i}}\end{bmatrix}\begin{bmatrix}\mathbf{P}_{\mathcal{R}\mathcal{R}} & \mathbf{P}_{\mathcal{R}\mathcal{L}_\mathbf{i}}\\ \mathbf{P}_{\mathcal{L}_\mathbf{i}\mathcal{R}} & \mathbf{P}_{\mathcal{L}_\mathbf{i}\mathcal{L}_\mathbf{i}}\end{bmatrix}\begin{bmatrix}\mathbf{H}_{\mathcal{R}}^\top\\ \mathbf{H}_{\mathcal{L}_\mathbf{i}}^\top\end{bmatrix} + \mathbf{V} \tag{3.30}$$

$$\mathbf{K} = \begin{bmatrix}\mathbf{P}_{\mathcal{R}\mathcal{R}} & \mathbf{P}_{\mathcal{R}\mathcal{L}_\mathbf{i}}\\ \mathbf{P}_{\mathcal{M}\mathcal{R}} & \mathbf{P}_{\mathcal{M}\mathcal{L}_\mathbf{i}}\end{bmatrix}\begin{bmatrix}\mathbf{H}_{\mathcal{R}}^\top\\ \mathbf{H}_{\mathcal{L}_\mathbf{i}}^\top\end{bmatrix}\mathbf{Z}^{-\mathbf{1}} \tag{3.31}$$

$$\bar{x} \leftarrow \bar{x} + \mathbf{K}\left(y_i - h_i\right) \tag{3.32}$$

$$\mathbf{P} \leftarrow \mathbf{P} - \mathbf{K}\mathbf{Z}\mathbf{K}^\top \tag{3.33}$$

3.2.1 Robot model

The robot model proposed in this section is based on a differential drive robot. It has been chosen to represent the TurtleBot, presented in Section 4.1.1, which has been used during experimental tests. The states that describe the robot's pose in 2D space are its cartesian coordinates,x and y, and its rotation angle about vertical

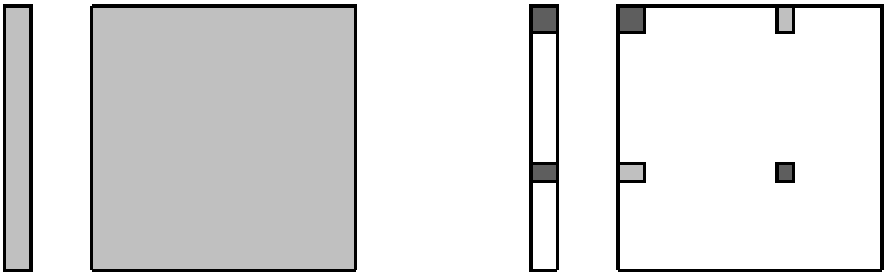


Figure 3.2: *Left*: On each measurement, the whole map is updated because the Kalman gain $\mathbf{K}$ affects the full state. *Right*: However, the computation of the innovation involves only the highlighted parts of the map. The involved terms are the robot state $\bar{\mathcal{R}}$ and the oserved landmark state $\mathcal{L}_i$ with their covariance, $\mathbf{P}_{\mathcal{R}\mathcal{R}}$ and $\mathbf{P}_{\mathcal{L}_\mathrm{i}\mathcal{L}_\mathrm{i}}$ (dark grey), and their cross-variances $\mathbf{P}_{\mathcal{R}\mathcal{R}}$ and $\mathbf{P}_{\mathcal{R}\mathcal{L}_\mathrm{i}}$ (light grey). Source: [11]

axis θ.

$$\mathcal{R}_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} \tag{3.34}$$

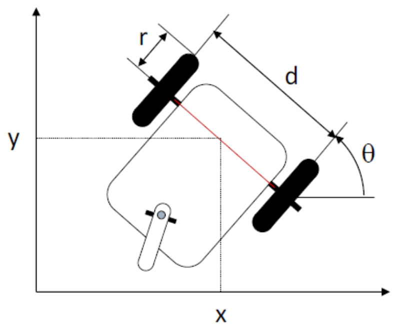


Figure 3.3: Model of a differential drive robot.

The robot motion is modeled as a unicycle with constant velocity inputs v_k and ω_k during the sampling interval $[t_k, t_{k+1}]$. In this interval, the robot follows a circular trajectory of radius $\frac{v_k}{\omega_k}$. A differential drive robot is equivalent to a unicycle as long as the wheels speeds are converted to linear and angular displacements:

$$\Delta s = \frac{r\left(\Delta\Phi_R + \Delta\Phi_L\right)}{2} \tag{3.35}$$

$$\Delta \theta = \frac{r\left(\Delta\Phi_R + \Delta\Phi_L\right)}{d} \tag{3.36}$$

where r is the radius of the wheel and d is the wheelbase.

Given as known the previous state of the robot $\mathcal{R}_k$ and the velocities v_k and ω_k, the next state $\mathcal{R}_{k+1}$ can be computed by integration over the time interval $[t_k, t_{k+1}]$. Different techniques can be used for the discrete integration, considering that the trade-off should be made between precision and computational complexity.

The first possibility is Euler integration. Using this method, the angle during the sampling time is considered constant and equal to the angle at the previous step θ_k.x_{k+1} and y_{k+1} are computed only in an approximate, while θ_{k+1} is computed in an exact way.

$$f\left(x_k, u_k, 0\right) = \begin{cases} x_{k+1} = & x_k + v_k T_s \cos\theta_k \\ y_{k+1} = & y_k + v_k T_s \sin\theta_k \\ \theta_{k+1} = & \theta_k + \omega_k T_s \end{cases} \tag{3.37}$$

where $T_s = t_{k+1} - t_k$.

The second way of integrating is the Runge-Kutta method, which approximates the angle between the initial and final point with its mean. Thus, the integration is still not exact but more accurate, introducing just a small overhead in computational complexity.

$$f\left(x_k, u_k, 0\right) = \begin{cases} x_{k+1} = & x_k + v_k T_s \cos\left(\theta_k + \frac{\omega_k T_s}{2}\right) \\ y_{k+1} = & y_k + v_k T_s \sin\left(\theta_k + \frac{\omega_k T_s}{2}\right) \\ \theta_{k+1} = & \theta_k + \omega_k T_s \end{cases} \tag{3.38}$$

Finally, it is possible to integrate in an exact way the kinematic model of the unicycle. Unfortunately, this leads to a set of equations that are not always defined, in particular for $\omega_k = 0$, which requires some attention in a digital implementation.

$$f\left(x_k, u_k, 0\right) = \begin{cases} x_{k+1} = & x_k + \frac{v_k}{\omega_k}\left(\sin\theta_{k+1} - \sin\theta_k\right) \\ y_{k+1} = & y_k + \frac{v_k}{\omega_k}\left(\cos\theta_{k+1} - \cos\theta_k\right) \\ \theta_{k+1} = & \theta_k + \omega_k T_s \end{cases} \tag{3.39}$$

The most suitable choice to implement an EKF is the Runge-Kutta integration because it is precise enough and does not suffer the presence of singularity points. The model is reported in Equation (3.38).

Model Linearization. In order to implement the model in an EKF, it should be linearized. The Jacobian matrices are computed according to Equations (3.9)

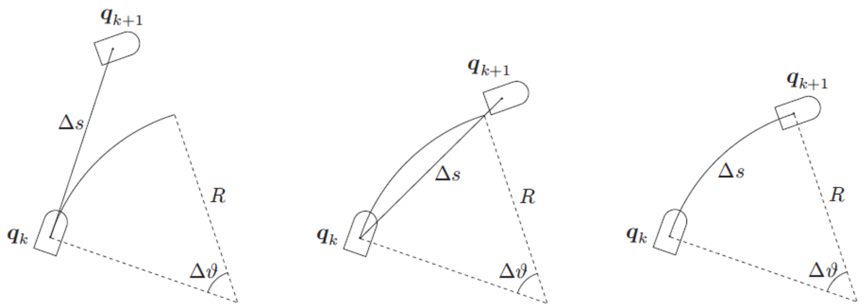


Figure 3.4: Representation of the different methods of integration. *From the left*: Euler, Runge-Kutta, exact. The Euler method approximates the arc of circumference with a straight line in the starting direction. Runge-Kutta also uses a straight line, but the direction is better approximated. As one may notice, the Runge-Kutta method provides a good approximation of the final point.

and (3.10). Here are reported the Jacobians used for the EKF.

$$\begin{aligned}\mathbf{G} &= \begin{bmatrix} \frac{\partial x_{k+1}}{\partial x_k} & \frac{\partial x_{k+1}}{\partial y_k} & \frac{\partial x_{k+1}}{\partial \theta_k} \\ \frac{\partial y_{k+1}}{\partial x_k} & \frac{\partial y_{k+1}}{\partial y_k} & \frac{\partial y_{k+1}}{\partial \theta_k} \\ \frac{\partial \theta_{k+1}}{\partial x_k} & \frac{\partial \theta_{k+1}}{\partial y_k} & \frac{\partial \theta_{k+1}}{\partial \theta_k} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & -v_k T_s \sin\left(\theta_k + \frac{\omega_k T_s}{2}\right) \\ 0 & 1 & v_k T_s \cos\left(\theta_k + \frac{\omega_k T_s}{2}\right) \\ 0 & 0 & 1 \end{bmatrix}\end{aligned} \tag{3.40}$$

$$\begin{aligned}\mathbf{W} &= \begin{bmatrix} \frac{\partial x_{k+1}}{\partial v_k} & \frac{\partial x_{k+1}}{\partial \omega_k} \\ \frac{\partial y_{k+1}}{\partial v_k} & \frac{\partial y_{k+1}}{\partial \omega_k} \\ \frac{\partial \theta_{k+1}}{\partial v_k} & \frac{\partial \theta_{k+1}}{\partial \omega_k} \end{bmatrix} \\ &= \begin{bmatrix} \cos\left(\theta_k + \frac{\omega_k T_s}{2}\right) & -\frac{1}{2} v_k T_s \sin\left(\theta_k + \frac{\omega_k T_s}{2}\right) \\ \sin\left(\theta_k + \frac{\omega_k T_s}{2}\right) & \frac{1}{2} v_k T_s \cos\left(\theta_k + \frac{\omega_k T_s}{2}\right) \\ 0 & 1 \end{bmatrix} T_s\end{aligned} \tag{3.41}$$

3.2.2 Measurement model

In the EKF implementation, two kinds of measures are used: distances from UWB sensors and odometry from wheel encoders.

UWB Sensors. The model used for UWB data is a simple distance model that depends on the estimated robot's pose $\mathcal{R}$ and the estimated anchor's position $\mathcal{L}_i$.

$$h\left(\mathcal{R}, \mathcal{L}_i\right) = \sqrt{\left(x_\mathcal{R} - x_{\mathcal{L}_i}\right)^2 + \left(y_\mathcal{R} - y_{\mathcal{L}_i}\right)^2 + \left(z_\mathcal{R} - z_{\mathcal{L}_i}\right)^2} \tag{3.42}$$

Then, the Jacobian must be computed, according to Equation (3.11), in order to include the measure in the EKF. Here is shown the derivation with respect to $x_\mathcal{R}$ and $x_{\mathcal{L}_i}$.

$$\frac{\partial h\left(\mathcal{R}, \mathcal{L}_i\right)}{\partial x_\mathcal{R}} = \frac{x_{\mathcal{L}_i} - x_\mathcal{R}}{\sqrt{\left(x_\mathcal{R} - x_{\mathcal{L}_i}\right)^2 + \left(y_\mathcal{R} - y_{\mathcal{L}_i}\right)^2 + \left(z_\mathcal{R} - z_{\mathcal{L}_i}\right)^2}} \tag{3.43}$$

$$\frac{\partial h\left(\mathcal{R}, \mathcal{L}_i\right)}{\partial x_{\mathcal{L}_i}} = \frac{x_\mathcal{R} - x_{\mathcal{L}_i}}{\sqrt{\left(x_\mathcal{R} - x_{\mathcal{L}_i}\right)^2 + \left(y_\mathcal{R} - y_{\mathcal{L}_i}\right)^2 + \left(z_\mathcal{R} - z_{\mathcal{L}_i}\right)^2}} \tag{3.44}$$

It may be noticed that the derivatives for the other coordinates are analogous to the one shown in Equations (3.43) and (3.44). Therefore, the Jacobian matrix can be written as follow

$$\mathbf{H} = \begin{bmatrix} \frac{x_{\mathcal{L}_i} - x_\mathcal{R}}{\|h(\mathcal{R}, \mathcal{L}_i)\|} & \frac{y_{\mathcal{L}_i} - y_\mathcal{R}}{\|h(\mathcal{R}, \mathcal{L}_i)\|} & 0 & \cdots & 0 & \frac{x_\mathcal{R} - x_{\mathcal{L}_i}}{\|h(\mathcal{R}, \mathcal{L}_i)\|} & \frac{y_\mathcal{R} - y_{\mathcal{L}_i}}{\|h(\mathcal{R}, \mathcal{L}_i)\|} & 0 & \cdots & 0 \end{bmatrix} \tag{3.45}$$

Odometry. The odometry model is very straightforward because it provides an observation of the robot's pose directly. So, the observation function is

$$h\left(\mathcal{R}\right) = \begin{cases} x_\mathcal{R} \\ y_\mathcal{R} \\ \theta_\mathcal{R} \end{cases} \tag{3.46}$$

Also the Jacobian matrix can be deduced without effort, and it is

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & & 0 \end{bmatrix} \tag{3.47}$$

3.2.3 Anchors' positions initialization

Since the EKF is based on the hypothesis of linearity about the current mean, a reliable initial guess for anchors' positions should be provided. In fact, breaking the linearity hypothesis leads to a distortion of the ideal Gaussian distribution and poor results from the filter. An algorithm relying on the robot's odometry and UWB distances has been employed to provide a good initial guess.

For each anchor, a number of points N is collected while the robot is moving freely inside the environment. For each point, the UWB distance and the odometry reading are collected. Then, the following equation can be written:

$$p_{an} = \underset{p_{an} \in \mathbb{R}^2}{\arg\min} \sum_{k=0}^{N} w_k \left(d_k - \|p_{an} - p_{tag,k}\|\right)^2 \tag{3.48}$$

where

- d_k is the UWB range at time k,
- $p_{tag,k}$ is the position of the robot at time k,
- p_{an} is the position of the anchor, which is the output of the minimization
- w_k is a heuristic weight to take into account the growing inaccuracy of odometry.

A general-purpose least-square solver is used to solve the equation.

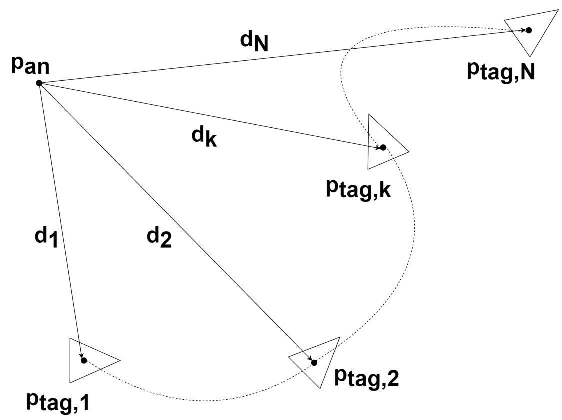


Figure 3.5: Representation of anchor's position initialization algorithm

It is to highlight that the path must be chosen carefully to avoid ambiguity of reflection. In particular, straight lines should be avoided because admissible solutions for the equation are two points symmetric with respect to the line.

3.3 Anchors' self-localization algorithm

In this Section, it is provided a description of the algorithm proposed and tested in this work. The overall structure of the filter is inspired by the works [6], [13].

The system's input is composed of the velocity command provided to the robot, the pose of the robot estimated by odometry, and the UWB distances between the robot's tag and the fixed anchors. The output of the whole process is the 2D position of all the UWB anchors and a corrected position estimate of the robot. The z-coordinate of each anchor is provided as a parameter as it can be easily measured while installing the anchors in place. Instead, the z-coordinate of the robot is assumed to be 0 and constant since the surface of motion is assumed to be flat.

Initially, the EKF only propagates the uncertainty of the robot's pose. While, after the initialization, it propagates the uncertainty both in the robot's pose and in anchors' positions. Initialization of anchors' positions is performed in parallel to the EKF process. In fact, it runs for several seconds until a reliable first guess is obtained and provided to the filter.

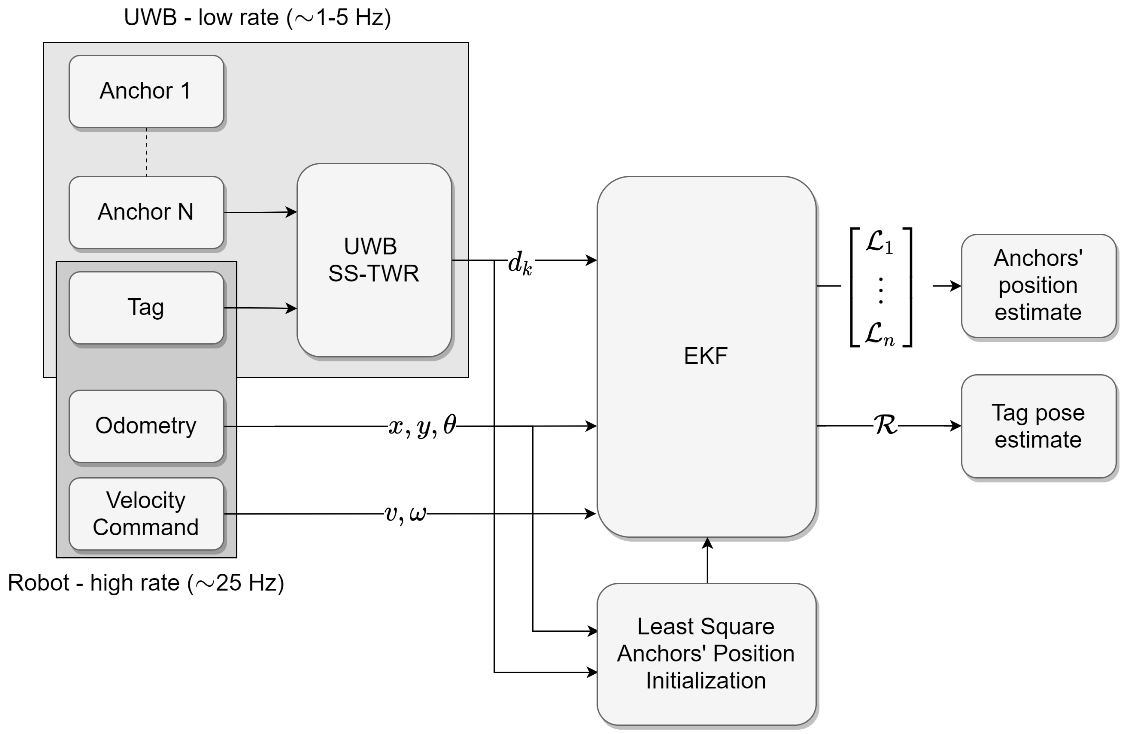


Figure 3.6: Working scheme of the algorithm

Chapter 4

Experimental Work

4.1 Instrumentation

In this Section, instrumentation used during the experimental tests is introduced. They are briefly described, and details on technical specifications are provided.

4.1.1 TurtleBot 3 Burger

TurtleBot is a ROS standard platform robot. TurtleBot3 is a small, affordable, programmable, ROS-based mobile robot for education, research, hobby, and product prototyping. The goal of TurtleBot3 is to dramatically reduce the size of the platform and lower the price without having to sacrifice its functionality and quality while at the same time offering expandability. Furthermore, the TurtleBot3 can be customized in various ways depending on how one reconstructs the mechanical parts and use optional components such as the computer and sensor.

The TurtleBot can run SLAM algorithms to build a map and drive around your room. Also, it can be controlled remotely from a laptop, joypad, or Android-based smartphone. [13]

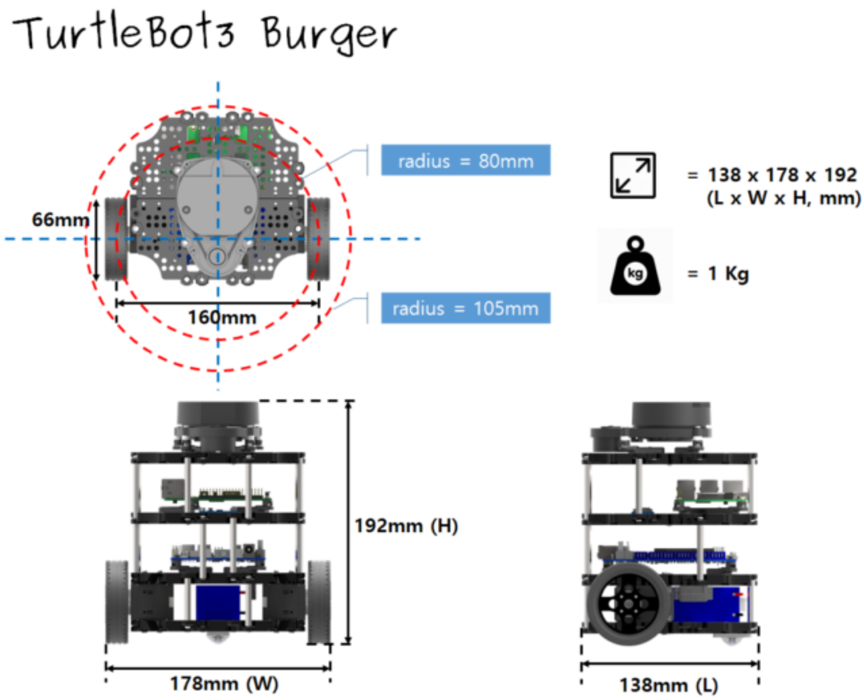


Figure 4.1: TurtleBot 3 Burger dimensional drawing. Source: [13]

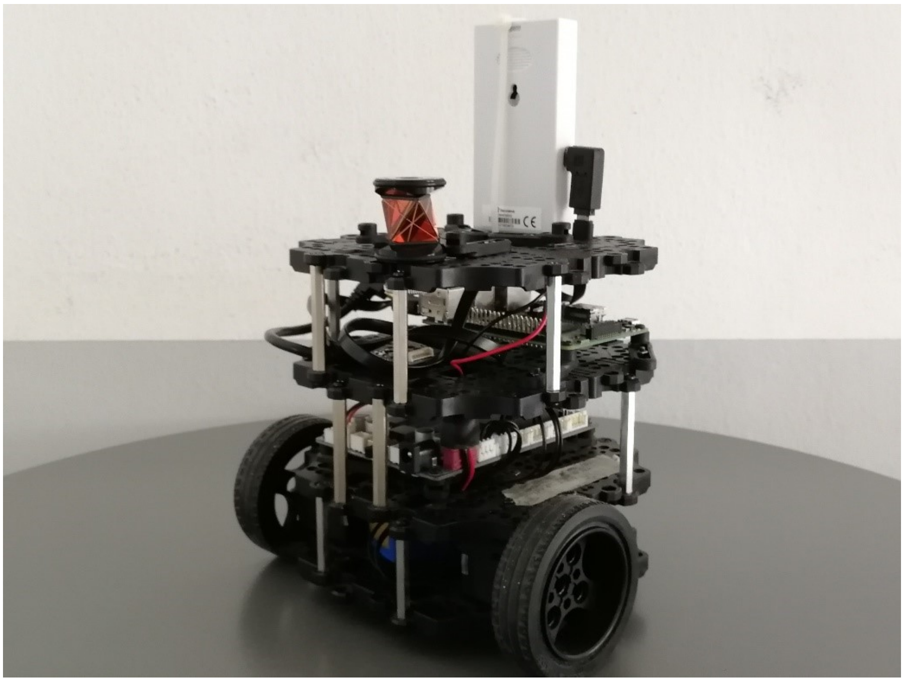


Figure 4.2: TurtleBot configuration used for all the experimental measures

Max Translational Velocity	0.22 m/s
Max Rotational Velocity	2.84 rad/s (162.72 deg/s)
Max Payload	15 kg
Size (LxWxH)	138 mm x 178 mm x 192 mm
Weight (Total)	1 kg
Expected Operating Time	2h 30m
SBC (Single Board Computer)	Rasperry Pi 3 Model B
MCU	32-bit ARM Cortex®-M7 with FPU (216 MHz, 462 DMIPS)
Actuator	XL430-W250
IMU	Gyroscope 3 Axis Accelerometer 3 Axis Magnetometer 3 Axis
Battery	Lithium polymer 11.1V 1800mAh / 19.98Wh 5C

Table 4.1: Technical specifications of TurtleBot 3 Burger

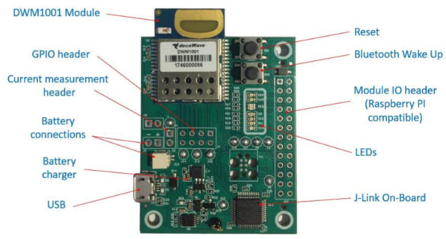
4.1.2 Decawave DWM1001-Dev modules

Decawave DWM1001-Dev module is an evaluation board produced by Decawave equipped out-of-the-box with firmware capable of providing localization features.

Modules incorporate a Decawave DW1000 UWB transceiver capable of sending and receiving UWB signals. The board also includes a Nordic Semiconductor NRF52832 IC. This provides a microcontroller for running the firmware that drives the UWB transceiver and a module for Bluetooth connectivity. [14]

For the application presented here, a different firmware has been used. Firmware is provided by Dynamic Distributed Decentralized Systems Group (D3S), a cross-institutional research group based in Trento, Italy. The FW is based on Contiki-OS, an open-source operating system that runs on constrained embedded systems and provides standardized low-power wireless communication. In particular, the application used is the multi-ranging one, which provides capabilities for multiple tags and anchors ranging in a round-robin, also multiple times per second. [15]

(a)



(b)

Figure 4.3: (a) Decawave DWM1001-Dev board. External packaging on the left, electronic board on the right (b) DWM1001-Dev board with on-board components description

Size (LxWxH)	20 mm x 45 mm x 94 mm
Weight	58 g
Operating Band	UWB Channel 5: 6.5 GHz
Data Rate	6.8 Mbps (IEEE 802.15.4-2011 UWB compliant)
Max Power Spectral Density	-41.3 dBm/MHz
Antenna	DecaWave DW1000 transceiver IC
Firmware	D3S Contiki Multi-Ranging [14]
Maximum Range	up to 60 m in LOS
Ranging Technique	SS-TWR or DS-TWR
Ranging Precision	10 cm (maximum)
Max Ranging Frequency	10 Hz (depend on the number of anchors)
Connections	- On board J-Link debugger - Serial communication via SPI, UART and BLE - 26-pin Raspberry Pi compatible header - On board access to DWM1001 pins

Table 4.2: Technical specifications of DecaWave DWM1001-Dev modules

4.1.3 Leica Absolute Tracker AT403

Leica Absolute Tracker AT403 is a portable laser tracker capable of continuous measurements and reflector tracking features. It is used to obtain a precise and reliable ground truth for anchors' position and the robot's trajectory. Measurements are acquired and saved thanks to SpatialAnalyzer® software, which allows the fast and easy generation of reports and ASCII files to be employed for data analysis. Technical specifications of the laser tracker are reported in Table 4.3.

Figure 4.4: Leica AT430 mounted on a tripod with remote control unit

(a) (b)

Figure 4.5: (a) 0.5" prism used for surveying anchors' positions. (b) Omni-directional prism used for robot tracking.

Size (LxWxH)	290 mm x 221 mm x 188 mm
Weight	7.3 kg
Measurement Angle	Hor. ±360°, Vert ±145°
Accuracy	±15 µm + 6 µm/m
Laser Class	2
Laser Type	635 nm, < 1 mW
Operating Temperature	-15°C to +45°C
Op. Relative Humidity	< 95%

Table 4.3: Leica AT430 technical specifications

4.2 Experimental tests

Several tests have been performed changing anchors placement and robot's trajectory to evaluate the performances of the proposed algorithm. In particular, two different layouts are evaluated for anchors positioning: one with four anchors placed in an even distribution at the vertices of a quadrilateral, the other with a fifth anchor added approximately in the middle of the localization area.

Performances are evaluated on parameters such as error in anchors' positioning, error in odometry and time of convergence. Moreover, a brief analysis is conducted on the UWB ranging precision and on the possible interferences with the reflecting prism.

4.2.1 UWB ranging precision

A first analysis, to obtain a general idea of the precision of the system, is conducted on the UWB ranging error. Also, it is investigated the influence of the reflecting prism, used to measure positions with the laser tracker, near the UWB antenna. Both measurements are performed by placing the robot in a static position, surveying its exact position with the help of the absolute tracker, then registering measurements for at least 30 seconds to have statistical relevance.

Firstly, UWB ranging accuracy has been investigated by acquiring UWB ranges to five anchors in three different tag positions. The position of the tag is measured each time using the laser tracker while the anchors' positions remain fixed. Measured data are reported in Table A.2. As it may be observed, the error from the true measure never overcome 23 cm, with a mean error of 13 cm, which is compatible with the expected precision of the system. Furthermore, it can be noticed that the error on the same UWB anchor is not constant but change with the distance. In conclusion, the overall precision of the ranging method is acceptable for the purpose of the work.

Secondly, it is investigated the influence of the prism near the UWB tag placed on the robot. The robot is fixed in place, and the tag's position is accurately measured using the laser tracker. Anchors' positions were already acquired when placing the localization system. Then, two series of measurements are collected, one with the prism in position and one without, each set containing about 150 range measurements. Measurements are compared according to their mean and standard deviation reported in Table A.1. Both means, with and without prism, are close to each other. Also, the standard deviation is similar in the two cases meaning the measures are distributed in the same way. Thus, it is evident that the prism does not affect in a relevant manner the UWB ranging precision.

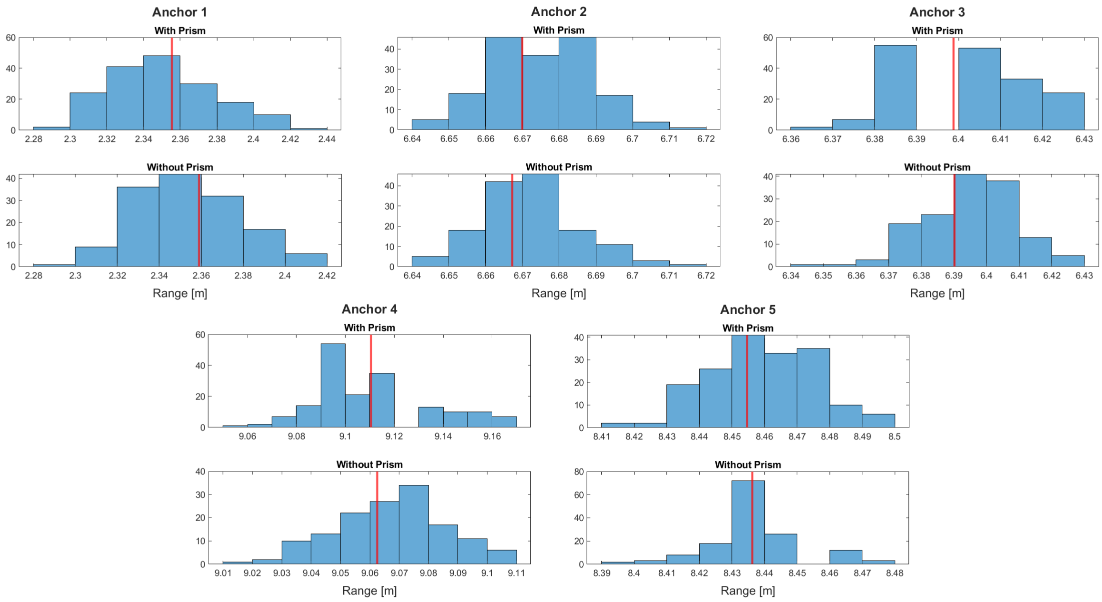


Figure 4.6: UWB ranges distribution with and without prism near the tag.

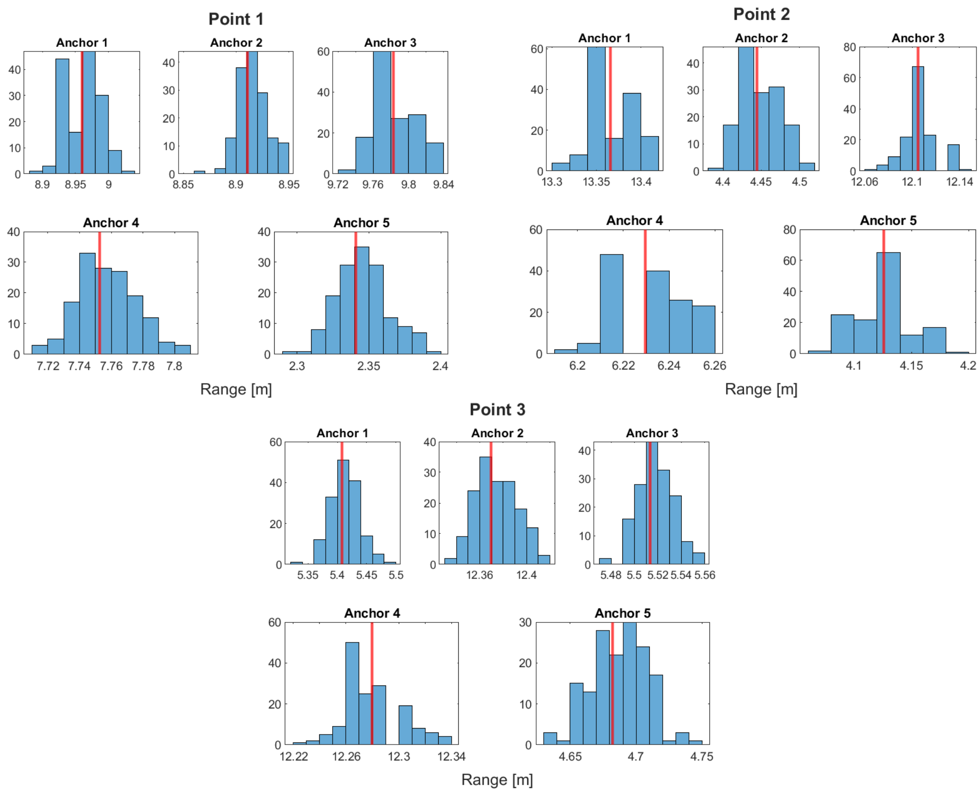


Figure 4.7: UWB ranging measurements to five anchors. Three different tag's positions are reported.

4.2.2 Four anchors layout

The first proposed layout for anchors positioning is one of the most widespread configurations for UWB localization system. It consists of four anchors placed at the edges of the room, as this number is the minimum requirement for a 3D localization system.

Initially, each anchor is placed and its absolute position is measured accurately using the laser tracker. Then, the height of each anchor is manually inserted in a parameter file along with its short address. Finally, the robot is positioned and its location is accurately marked on the ground to create a repeatable starting point. Furthermore, the initial robot placement is used to create the RF transformation in the measuring software to align all the measures to the robot's RF.

Figure 4.8: Experimental setup with four anchors

First, a circular and regular path is performed by the robot, imposing a constant linear and angular velocity. (see Figure 4.9a) The actual trajectory is measured once a second through the laser tracker, which can follow the omni-directional prism mounted on the TurtleBot. Some points are missing due to the shadowing of the prism by the UWB sensor mounted on the robot. (see Figure 4.2)

As a second and more general test, the robot is conducted manually through a dedicated ROS node. The produced trajectory is not regular and aims at reproducing a random path, as shown in Figure 4.12a. However, some limits must be imposed on the robot trajectory, such as a maximum speed, to allow the laser tracker to follow the reflecting prism, and a simple dynamic, which does not imply significant accelerations and decelerations.

Performances of the system are evaluated according to the following parameters:

- Absolute error in the estimated positions of the anchors, computed as the distance between the ground truth and the estimated position:

$$\varepsilon_{abs} = \sqrt{(x_{GT} - x_{EKF})^2 + (y_{GT} - y_{EKF})^2} \tag{4.1}$$

where x_{GT} and y_{GT} are the ground truth coordinates and x_{EKF} and y_{EKF} are the estimated coordinates

- Pose estimation error: absolute error in time and PDF of the error after the first 150 seconds. It is compared with the standard odometry of the TurtleBot.
- Time of convergence of the EKF. It is given as a time after which the output of the filter can be considered qualitatively constant.

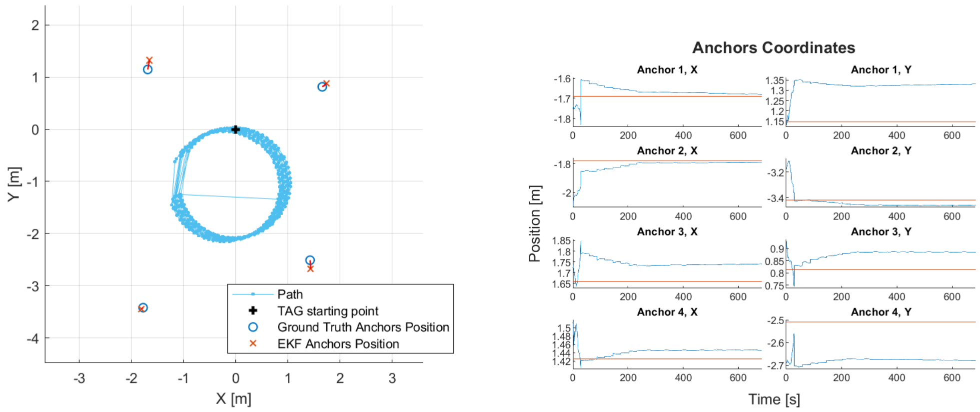
Results show that after a first transition period, during around 200 seconds, the estimated coordinates of the anchors converge to a constant value. During the initial phase, the filter is oscillating and the odometry provided by the EKF is not reliable. When the filter converges to a constant value for the anchors' position also the odometry becomes reliable.

In both runs, the positions of the anchors are estimated with a precision of over 20 cm, except for "Anchor 1" in the second test, as it can be seen in Figures 4.10 and 4.13 and Tables A.3 and A.5. The mean time of convergence of the estimation algorithm is about three minutes. The result is acceptable and within the expectation, considering the mean ranging precision of UWB.

Furthermore, it can be noticed from Figures 4.11 and 4.14 that, after the first transient period of the EKF, the odometry is substantially improved even if the anchors' positions are not exact. This result is highly relevant because it allows the localization of a tag with a precision comparable or greater than the ranging precision, even in the absence of a precise localization of the anchors.

Focusing on the test with the circular path, it can be noticed from Figure 4.11 that the error of the robot's odometry clearly follows an increasing path, as expected. On the contrary, after an initial transient, the robot's estimated pose error tends to remain constant and below the former one. Despite the fact that some spikes can be noticed, they are present at regular intervals and are due to the laser tracker losing focus on the prism while it is shadowed. Plotting the PDF of the error of both the pose estimates after 200 s from the start, it is evident that the proposed algorithm has an improving effect.

Furthermore, the observed result is also more evident in the second test. In fact, the robot's odometry deteriorates faster in the presence of irregular moving paths. Thus the improving effect is also more evident.



(a) (b)

Figure 4.9: Four anchors layout, first test. *Left:* Estimated anchors' positions and robots' trajectory. *Right:* Estimated anchors' coordinates over time. Red line represents the reference value.

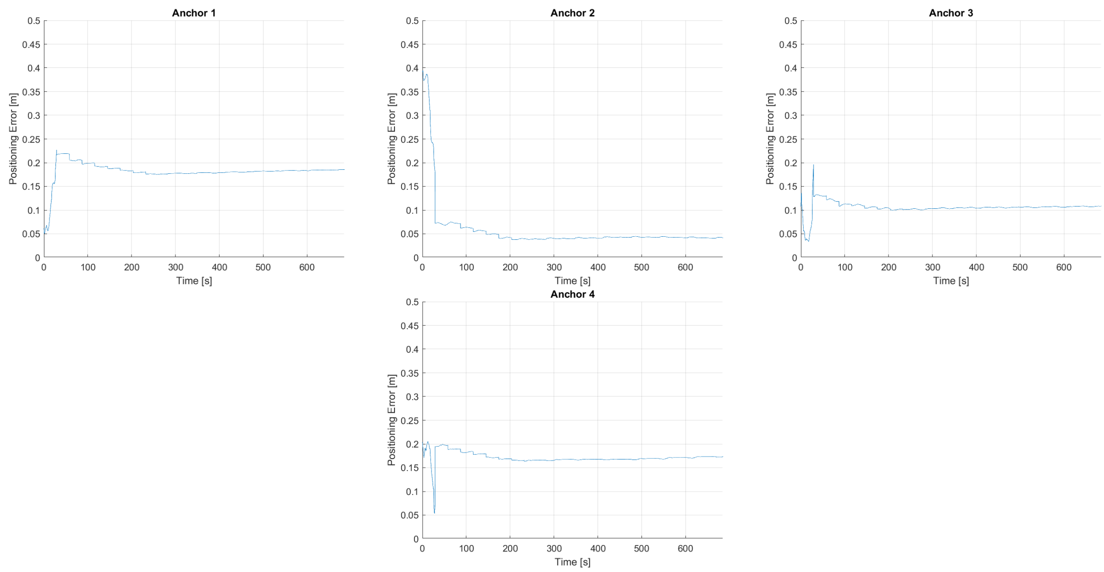


Figure 4.10: Four anchors layout. First test. Anchors' positioning error.

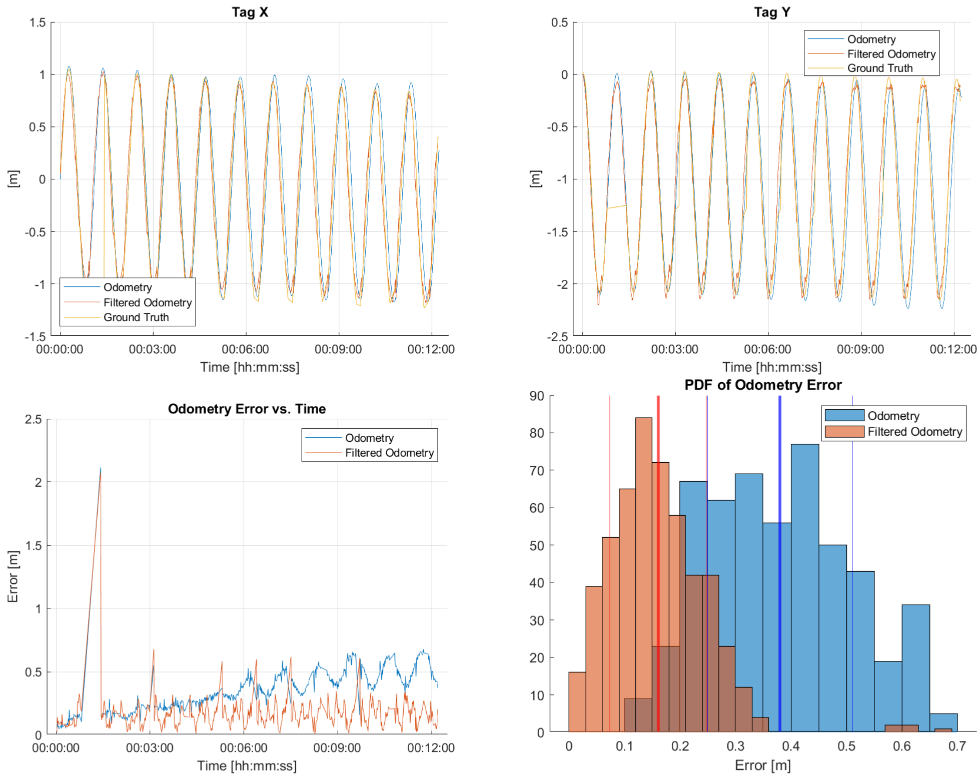
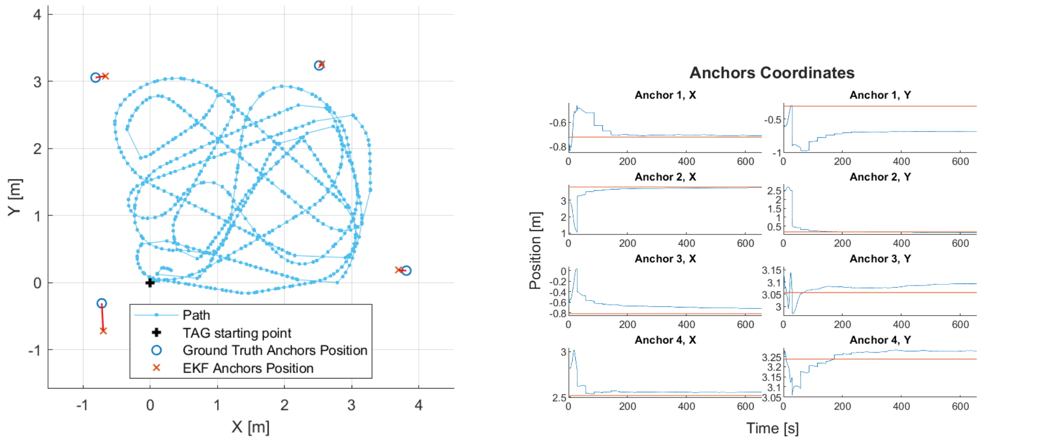


Figure 4.11: Four anchors layout, first test. Odometry. *Top:* x and y coordinates of the robot. *Bottom:* On the left absolute error vs. time, on the right the PDF of the error after the first 150 seconds.



(a) (b)

Figure 4.12: Four anchors layout, second test. *Left:* Estimated anchors' positions and robots' trajectory. *Right:* Estimated anchors' coordinates over time. Red line represents the reference value.

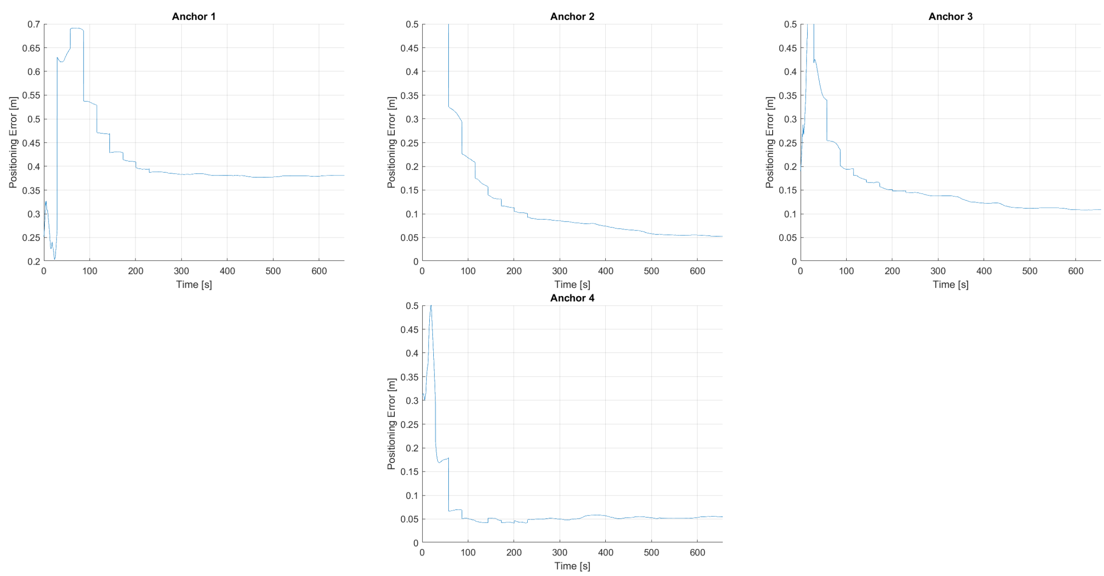


Figure 4.13: Four anchors layout. Second test. Anchors' positioning error.

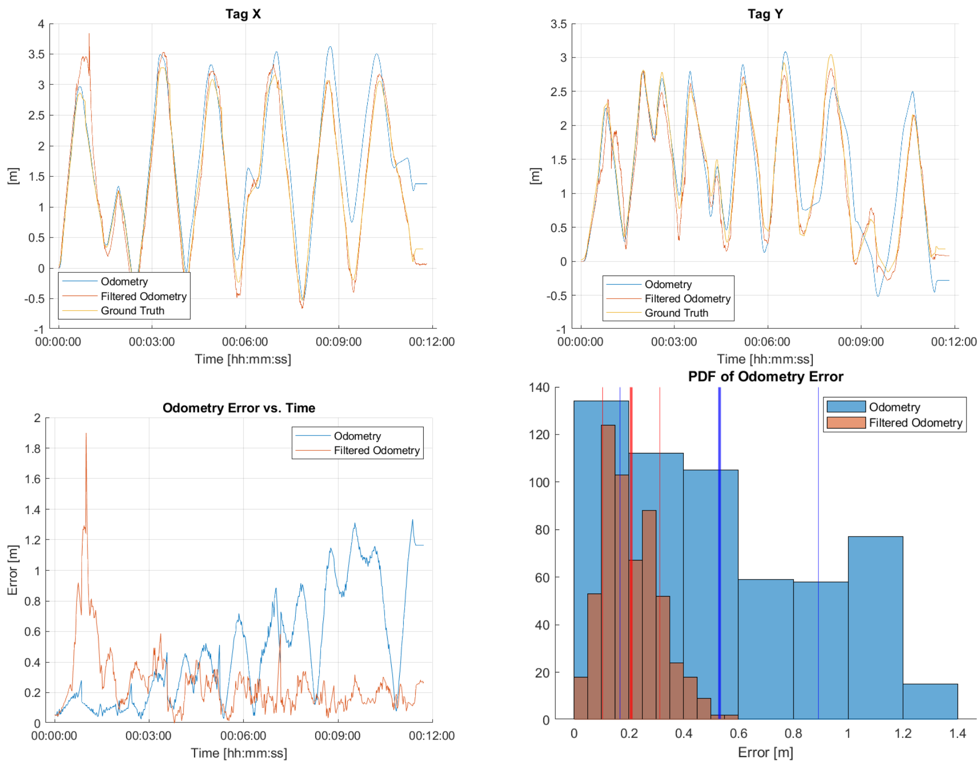


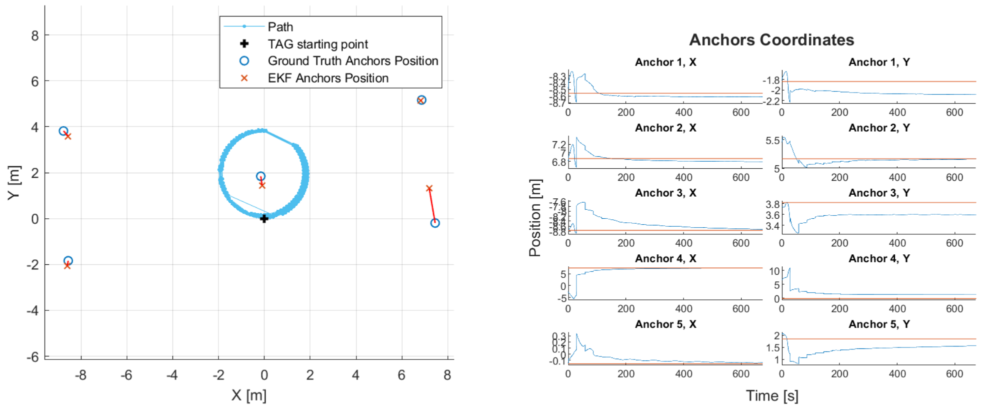
Figure 4.14: Four anchors layout, second test. Odometry. *Top:* x and y coordinates of the robot. *Bottom:* On the left absolute error vs. time, on the right the PDF of the error after the first 150 seconds.

4.2.3 Five anchors layout

A different anchors layout, with an additional anchor, is analyzed for two main reasons. First, it is known from literature that increasing the number of anchors leads to increased precision in localization. Second, the localization area is broader and the distances are greater to try out the system's reliability in different conditions.

The anchors are disposed at the edges of a rectangular room and the fifth anchor is placed almost in the center of it. Also, the robot is placed and the setup process is performed exactly as in Section 4.2.2.

First, a test is performed using a circular pattern to have a control reference comparable with the previous test. (Figure 4.15a) From the obtained data (Figure 4.17), it can be observed that four anchors are placed with an error comparable to the previous test. However, "Anchor 4" is localized with an error greater than one meter. The error is caused by a poor initialization of the position of the anchor, as it can be seen in Figure 4.15b. The path followed by the robot is probably too confined and does not allow a reliable initialization of the landmark. Nevertheless, the estimated pose of the robot, over a long period, is still more reliable than the robot's odometry, even if the absolute error is greater than the one in the previous tests.



(a) (b)

Figure 4.15: Five anchors layout, first test. *Left:* Estimated anchors' positions and robots' trajectory. *Right:* Estimated anchors' coordinates over time. Red line represents the reference value.

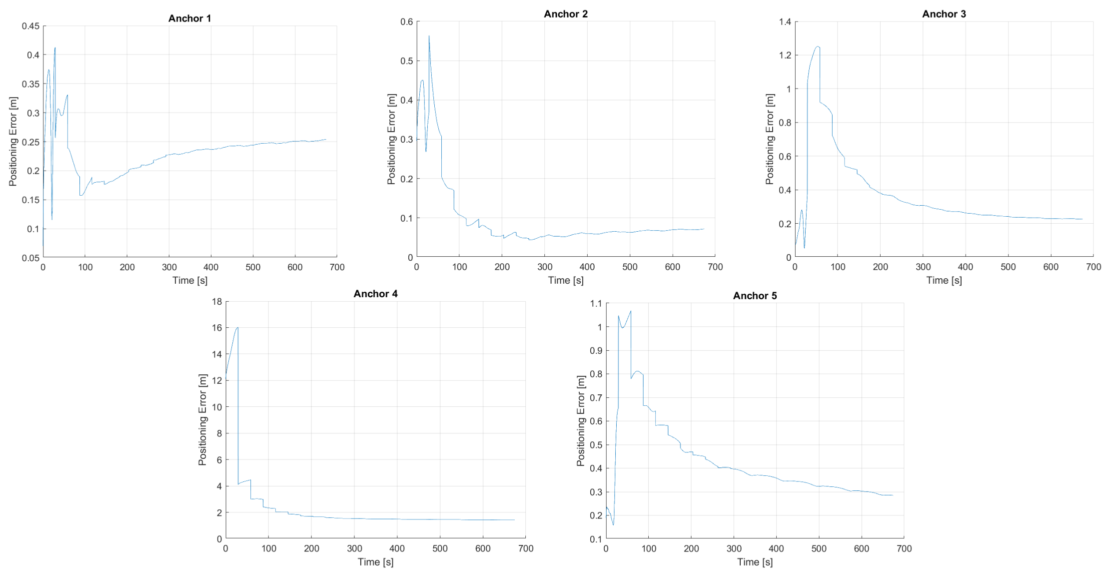


Figure 4.16: Five anchors layout. First test. Anchors' positioning error.

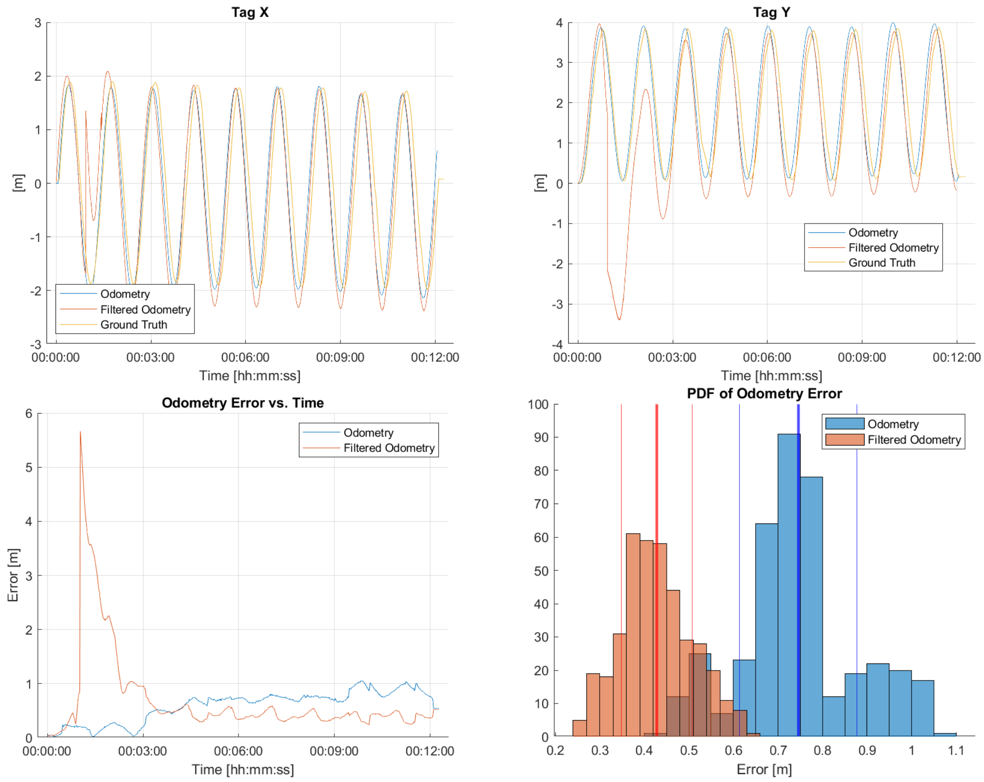
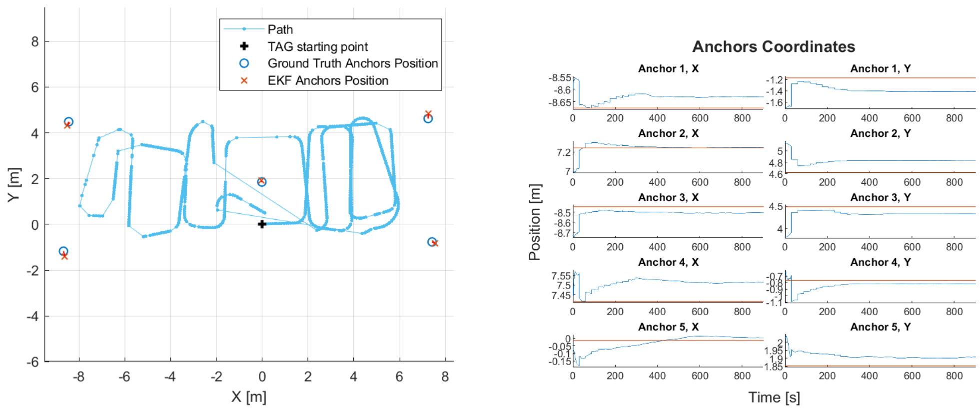


Figure 4.17: Five anchors layout. First test. Odometry. *Top:* x and y coordinates of the robot. *Bottom:* On the left absolute error vs. time, on the right the PDF of the error after the first 150 seconds.

Finally, a more general test is conducted by freely moving the robot to cover almost all the available space, as it is shown in Figure 4.18a. The estimated position of the anchors is close to the actual position since the initialization. Also, in Figure 4.19 it can be observer a small undershoot and then a stabilization around a constant error. The final error is small for all the anchors, with the best anchor within 5 cm of the ground truth and the worse within 25 cm. (see Table A.9) Moreover, the pose estimation is reliable since the initial phase thanks to the excellent initialization of the EKF. As it can be seen in Figure 4.20, the TurtleBot's odometry after a couple of minutes becomes completely wrong and the improvement in the pose estimation can be clearly seen in the PDF graph.



(a) (b)

Figure 4.18: Five anchors layout, second test. *Left:* Estimated anchors' positions and robots' trajectory. *Right:* Estimated anchors' coordinates over time. Red line represents the reference value.

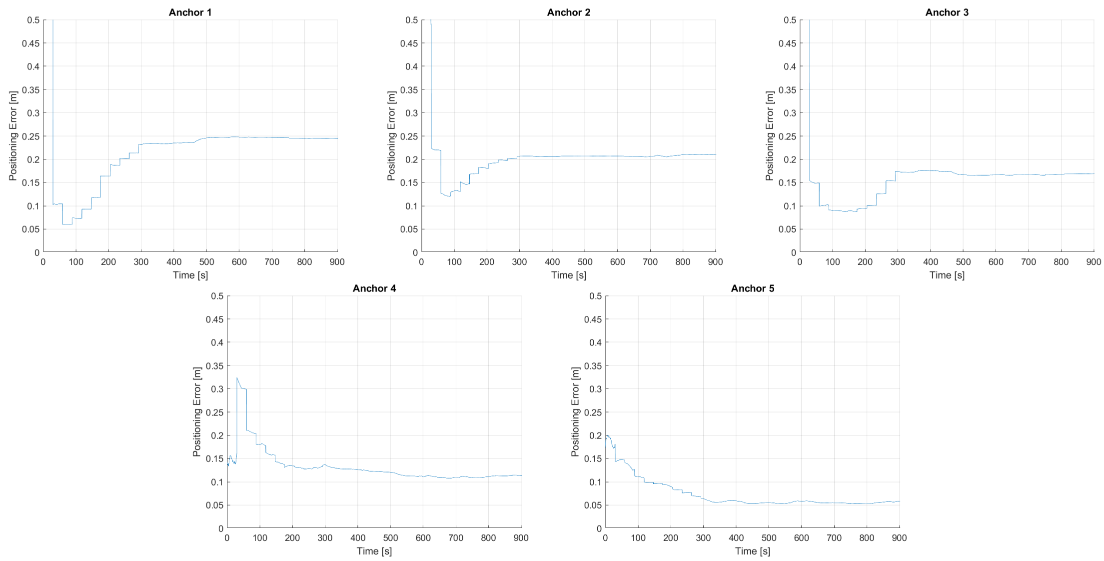


Figure 4.19: Five anchors layout. Second test. Anchors' positioning error.

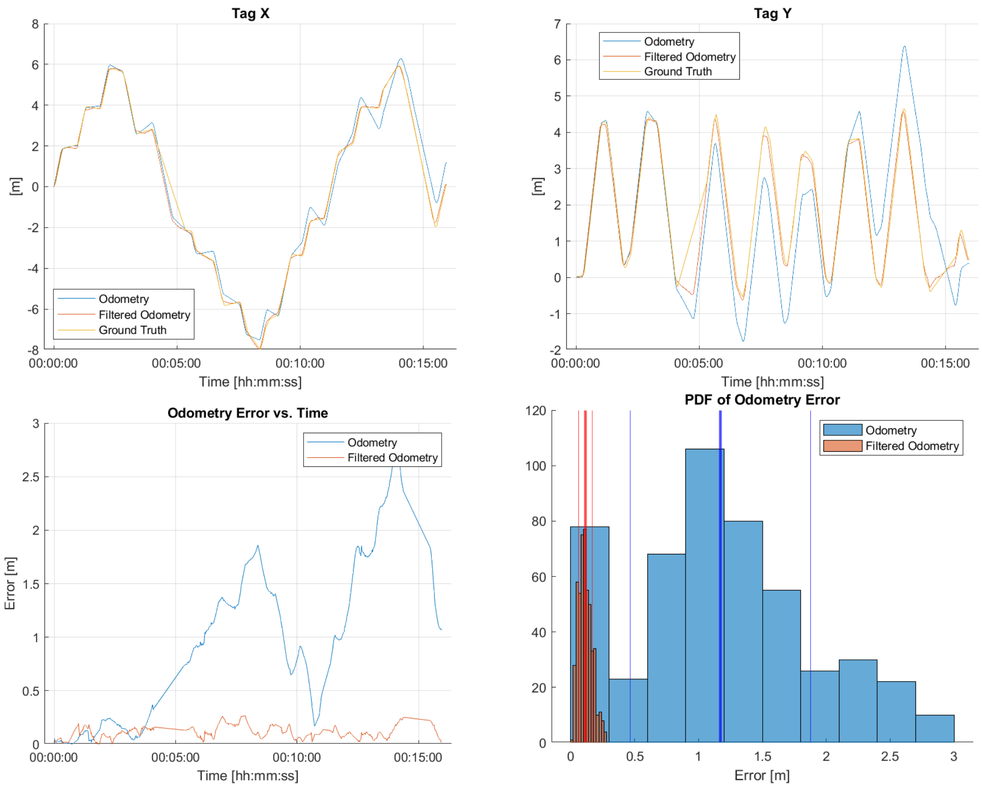


Figure 4.20: Five anchors test. Second test. Odometry. *Top:* x and y coordinates of the robot. *Bottom:* On the left absolute error vs. time, on the right the PDF of the error after the first 150 seconds.

Chapter 5

Conclusions

The work presented in this thesis aims at finding an efficient and reliable way to set up a UWB localization system. The proposed solution makes use of a wheeled ground vehicle equipped with a UWB tag, which is capable of providing information on its position through odometry and distances measurements to fixed UWB anchors. These data are then fused together through an EKF which estimates the positions of the fixed anchors and the pose of the robot.

The tests performed in indoor environments show promising results for the algorithm. In all the proposed layouts the algorithm is able to estimate with high precision the 2D positions of at least three anchors while the remaining, one or two, are estimated with an increased but acceptable error. Moreover, a further result is obtained which was not in the initial intent of the project. Indeed, the pose estimation of the moving tag is strongly improved with respect to the odometry provided by the rover itself. The combination of the two results can lead to interesting considerations. In fact, the proposed algorithm not only provides a system for a fast deployment of a localization system but also guarantees that the error in the positioning of the tag is improved.

Some improvements can still be made to the algorithm, leaving space for future works. For example, the extension of the method to a 3D environment could lead to consistent use of the system in outdoor environments, where the terrain is often not even and there are significant differences in height. Furthermore, for remote zones, where human beings can not be present, a self-deploying localization system could be developed using UGVs or UAVs, enabling localization almost in every place.

References

[1] H. Nikookar and R. Prasad, *Introduction to Ultra Wideband for Wireless Communications*, Springer, Ed. 2009, p. 188, ISBN: 9781402066320.

[2] T. Zwick, W. Wiesbeck, J. Timmermann, and G. Adamiuk, *Ultra-wideband RF System Engineering*, T. Zwick, W. Wiesbeck, J. Timmermann, and G. Adamiuk, Eds., 9. Cambridge University Press, 2013, vol. 53, ISBN: 9781107015555. arXiv: `arXiv:1011.1669v3`.

[3] H. Arslan, N. C. Zhi, and M.-G. Di Benedetto, *Ultra Wideband Wireless Communication.* 2006, p. 500, ISBN: 9786468600.

[4] H. Rohling, *OFDM: Concepts for Future Communication Systems.* 2011.

[5] T. Kurner, M. Jacob, R. Piesiewicz, and J. Schobel, "An integrated simulation environment for the investigation of future THz communication systems", *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, 2007.

[6] Q. Shi, S. Zhao, X. Cui, M. Lu, and M. Jia, "Anchor self-localization algorithm based on UWB ranging and inertial measurements", *Tsinghua Science and Technology*, vol. 24, no. 6, pp. 728–737, 2019, ISSN: 18787606. DOI: `10.26599/TST.2018.9010102`.

[7] S. A. Zekavat and M. Buehrer, *Handbook of Position Location: Theory, Practice, and Advances.* 2011, ISBN: 9780470943427. DOI: `10.1002/9781118104750`.

[8] A. Alarifi, A. Al-Salman, M. Alsaleh, *et al.*, "Ultra wideband indoor positioning technologies: Analysis and recent advances", *Sensors (Switzerland)*, vol. 16, no. 5, pp. 1–36, 2016, ISSN: 14248220. DOI: `10.3390/s16050707`.

[9] C. L. Sang, M. Adams, T. Hörmann, M. Hesse, M. Porrmann, and U. Rückert, "Numerical and experimental evaluation of error estimation for two-way ranging methods", *Sensors (Switzerland)*, vol. 19, no. 3, 2019, ISSN: 14248220. DOI: `10.3390/s19030616`.

[10] G. Welch and G. Bishop, "An Introduction to the Kalman Filter", *SIGGRAPH 2001*, 2001.

[11] J. Solà, "Simulataneous localization and mapping with the extended Kalman filter ' A very quick guide ... with Matlab code !'", pp. 1–35, 2012.

[12] C. Stachniss, J. J. Leonard, and S. Thrun, "Simultaneous localization and mapping", in *Springer Handbook of Robotics*, Springer International Publishing, 2016, pp. 1153–1175, ISBN: 9783319325521. DOI: 10.1007/978-3-319-32552-1_46.

[13] A. M. Santana, A. A. Sousa, R. S. Britto, P. J. Alsina, and A. A. Medeiros, "Localization of a mobile robot based in odometry and natural landmarks using Extended Kalman Filter", *ICINCO 2008 - Proceedings of the 5th International Conference on Informatics in Control, Automation and Robotics*, vol. 2 RA, no. May 2014, pp. 187–193, 2008. DOI: 10.13140/2.1.2081.8882.

[14] P. Corbalán, T. Istomin, and G. P. Picco, "Poster: Enabling Contiki on Ultra-wideband Radios", in *Proceedings of the International Conference on Embedded Wireless Systems and Networks*, ser. EWSN'18, 2018.

Appendix A

Experimental Data

	Ref	Prism			No Prism		
	Ref	**Mean**	**Std Dev**	**Error**	**Mean**	**Std Dev**	**Error**
	[m]	[m]	[cm]	[cm]	[m]	[cm]	[cm]
An. 1	2.100	2.355	28.7	+25.5	2.359	25.5	+25.9
An. 2	6.674	6.670	13.8	-0.40	6.667	13.3	-0.70
An. 3	6.095	6.400	14.1	+30.5	6.390	14.9	+21.4
An. 4	8.960	9.110	23.7	+15.0	9.062	19.1	+10.2
An. 5	8.238	8.454	16.7	+21.6	8.436	15.7	+20.4

Table A.1: UWB ranges measured with and without prism reflector on the robot

		Ref [m]	UWB Range [m]	Error [cm]
1	**An. 1**	8.900	8.960	+6.00
	An. 2	8.800	8.911	+11.1
	An. 3	9.620	9.783	+16.3
	An. 4	7.664	7.753	+8.90
	An. 5	2.111	2.341	+23.0
2	**An. 1**	13.149	13.366	+21.7
	An. 2	4.252	4.444	+19.2
	An. 3	11.948	12.105	+15.7
	An. 4	6.176	6.230	+5.40
	An. 5	3.980	4.126	+14.6
3	**An. 1**	5.248	5.408	+16.0
	An. 2	12.281	12.285	+0.40
	An. 3	5.294	5.513	+21.9
	An. 4	12.160	12.280	+12.0
	An. 5	4.605	4.682	+7.70

Table A.2: UWB ranging error for three different positions of the tag.

	Ref		**0 s**			**100 s**			**300 s**			**500 s**		
	x[m]	**y**[m]	**x**[m]	**y**[m]	ε[cm]	**x**[m]	**y**[m]	ε[cm]	**x**[m]	**y**[m]	ε[cm]	**x**[m]	**y**[m]	ε[cm]
An. 1	-1.69	1.14	-1.74	1.14	6.2	-1.64	1.34	19.9	-1.67	1.32	17.7	-1.68	1.33	18.2
An. 2	-1.78	-3.42	-2.07	-3.17	39.5	-1.84	-3.43	6.4	-1.79	-3.46	4.0	-1.79	-3.46	4.3
An. 3	1.66	0.81	1.72	0.90	10.8	1.76	0.86	11.3	1.73	0.88	10.3	1.74	0.89	10.6
An. 4	1.42	-2.51	1.51	-2.69	20.4	1.43	-2.69	18.2	1.44	-2.67	16.6	1.45	-2.68	16.9

Table A.3: Four anchors layout. First test. Anchors positioning.

	Error [cm]	
	Mean	**Std Dev**
Odometry	38.0	13.08
Filtered	16.07	8.69

Table A.4: Four anchors layout. First test. Odometry.

	Ref		0 s			100 s			300 s			500 s		
	x[m]	**y**[m]	**x**[m]	**y**[m]	ε[cm]	**x**[m]	**y**[m]	ε[cm]	**x**[m]	**y**[m]	ε[cm]	**x**[m]	**y**[m]	ε[cm]
An. 1	-0.72	-0.31	-0.72	-0.56	25.1	-0.63	-0.84	53.5	-0.71	-0.69	38.3	-0.71	-0.69	37.8
An. 2	3.82	0.18	2.88	2.38	238	3.61	0.25	21.8	3.73	0.17	8.5	3.76	0.15	5.8
An. 3	-0.82	3.06	-0.65	3.15	19.0	-0.62	3.07	19.4	-0.68	3.08	13.7	-0.71	3.09	11.1
An. 4	2.52	3.24	2.83	3.24	31.0	2.55	3.20	5.09	2.55	3.27	5.00	2.55	3.28	5.26

Table A.5: Four anchors layout. Second test. Anchors positioning.

	Error [cm]	
	Mean	**Std Dev**
Odometry	53.01	36.28
Filtered	20.92	10.46

Table A.6: Four anchors layout. Second test. Odometry.

	Ref		0 s			100 s			300 s			500 s		
	x[m]	**y**[m]	**x**[m]	**y**[m]	ε[cm]	**x**[m]	**y**[m]	ε[cm]	**x**[m]	**y**[m]	ε[cm]	**x**[m]	**y**[m]	ε[cm]
An. 1	-8.56	-1.84	-8.49	-1.81	7.0	-8.54	-2.00	16.6	-8.61	-2.06	22.7	-8.61	-2.08	24.4
An. 2	6.86	5.17	6.81	5.46	29.6	6.92	5.08	10.7	6.82	5.14	5.3	6.80	5.16	6.3
An. 3	-8.77	3.82	-8.77	3.74	7.6	-8.19	3.51	64.5	-8.56	3.59	30.7	-8.67	3.59	24.2
An. 4	7.45	-0.19	-2.66	6.49	1212	6.48	1.93	234	7.19	1.32	153	7.37	1.25	145
An. 5	-0.16	1.84	-0.15	2.03	18.9	-0.03	1.2	65.7	-0.10	1.45	39.7	-0.13	1.52	32.3

Table A.7: Five anchors layout. First test. Anchors positioning.

	Error [cm]	
	Mean	**Std Dev**
Odometry	74.58	13.16
Filtered	42.80	8.00

Table A.8: Five anchors layout. First test. Odometry.

	Ref		0 s			100 s			300 s			500 s		
	x[m]	**y**[m]	**x**[m]	**y**[m]	ε[cm]	**x**[m]	**y**[m]	ε[cm]	**x**[m]	**y**[m]	ε[cm]	**x**[m]	**y**[m]	ε[cm]
An. 1	-8.68	-1.17	-8.60	-1.71	54.1	-8.67	-1.24	7.37	-8.62	-1.40	23.2	-8.63	-1.41	24.6
An. 2	7.24	4.62	6.99	5.13	56.8	7.30	4.74	13.2	7.25	4.83	20.6	7.25	4.83	20.6
An. 3	-8.45	4.48	-8.73	3.81	72.9	-8.49	4.41	9.0	-8.50	4.32	17.3	-8.50	4.33	16.6
An. 4	7.41	-0.77	7.50	-0.65	14.2	7.48	-0.94	18.1	7.54	-0.83	13.7	7.52	-0.82	12.0
An. 5	-0.01	1.85	-0.04	2.02	17.4	-0.09	1.93	11.1	-0.05	1.91	6.4	-0.00	1.91	5.5

Table A.9: Five anchors layout. Second test. Anchors positioning.

	Error [cm]	
	Mean	**Std Dev**
Odometry	117.1	70.51
Filtered	11.49	5.37

Table A.10: Five anchors layout. Second test. Odometry.