On-board Micro Quadrotor State Estimation Using Range Measurements

A Moving Horizon Approach

S. Stroobants

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

On-board Micro Quadrotor State Estimation Using Range Measurements

A Moving Horizon Approach

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

S. Stroobants

October 15, 2019

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

On-board Micro Quadrotor State Estimation Using Range Measurements

by

S. Stroobants

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: October 15, 2019

Supervisor(s):

Dr.ir. T. Keviczky

Dr.ir. G.C.H.E De Croon

Reader(s):

S. Li MSc

ir. C. De Wagter

Abstract

Accurate indoor localization is essential for autonomous robotic agents to perform tasks ranging from warehouse management to remote sensing in greenhouses. Recently Ultra Wideband (UWB) distance measurements have been used to estimate position and velocity indoors. These UWB-measurements are known to be corrupted by a varying bias. Besides, current estimation methods are not suitable for large areas with a low beacon coverage. The goal of this thesis was therefore twofold. First, a simple bias model was proposed to reduce the influence of the UWB bias while still being implementable on a micro-processor. This model was shown to reduce the measurement error with 50% on validation data. Using this model, UWB-localization in a static beacon-configuration can be quickly improved. Second, an adaptation of the standard Moving Horizon Estimation (MHE) method was proposed that uses a time-window of range measurements to increase the robustness to outliers and is still real-time implementable on a micro-processor. This Moving Horizon Model Parametrization (MH-MP) does not estimate every state in the complete time-window, but only estimates an offset of the initial state in the window. An analysis of simulation data and data gathered in flight has shown that the proposed MH-MP outperforms the Extended Kalman Filter (EKF) in both the position and velocity estimate and has a comparable computation time. Further research is necessary to investigate the possibility of estimating the UWB-bias model parameters online.

Table of Contents

List of Figures

List of Tables

Preface

Since quadrotor research was one of the main reasons for me to start studying Systems and Control, being able to do my final research in this field was a dream come true. I would like to thank Tamás Keviczky for making this possible by introducing me to the MAVLAB. I would also like to thank Guido de Croon for the bi-weekly fruitful discussions that helped me push on with my research. The research would not have been the same without all the coffee-breaks with my friends and fellow students from the TU Delft. During these moments, we often seized the opportunity to have useful discussions that kept me focused. Last, but certainly not least, I want to thank Shushuai Li for supporting me with all the technical difficulties that arose during the practical part of my research. Without his help and understanding, the work would have been a lot harder.

"The beautiful thing about learning is nobody can take it away from you"
— *B. B. King*

Chapter 1

Introduction

Over the past decade, automation has increased the amount of autonomous robotic agents operating in indoor environments. These agents often need to known their location to perform their tasks. Examples are robots that are used to speed up order fulfillment in large warehouses [6], quadrotors that sense vital environmental variables to increase efficiency in greenhouses [7] and search and rescue missions in for humans inaccessible areas [8]. For these indoor robot applications, a highly accurate state estimate is necessary in order to perform certain tasks. For quadrotors, this can involve landing on a charging platform which requires millimeter precision to be able to activate the charging station. Besides this, these applications might require multiple agents operating simultaneously in the same space. To enable autonomous operation, agents need to be able to calculate their own location without the use of a central computer. Having a cheap, online, decentralized, accurate indoors method of localization can have a high impact on applications for mobile robotics.

1-1 Positioning Technology

A common solution for position estimation is currently the use of a Global Positioning System (GPS) [9, 10]. As long as the GPS-receiver has access to the signal of four satelites, it is able obtain a position-lock. With a normal commercial GPS receiver, like the U-Blox NEO-M8, an accuracy of 2.5 meters can be obtained [11]. For highly dynamic maneuvers in small areas this is not accurate enough. A higher accuracy can be provided (in the order of centimeters) with the use of a base station [10]. Since most of the GPS errors are due to multipathing or atmospheric effects, these can be cancelled out by using a base-station with a known location in the neighbourhood of the receiver.
This GPS signal cannot be used reliably inside or in other places where the GPS signal is obstructed [12]. Also the frequency of GPS is limited to 10Hz for commercial systems [11][10].

For indoor localization, multiple solutions are currently in use. Besides using the on-board Inertial Measurement Unit (IMU), indoor localization methods in research usually rely on external motion capture systems, such as the OptiTrack system that uses vision and reflective

markers to accurately estimate the location. This system consists of multiple cameras that combine near-infrared strobes with reflective markers on the robot to estimate its location [13]. This way, an accuracy in the order of millimeters can be achieved with an update-frequency of 200Hz [14].
The MOCAP system has a couple of disadvantages as well. The costs of such a system are high and installation and calibration takes a lot of time, which makes them inaccessible for small research groups and small companies.
Also, camera systems need Line Of Sight (LOS), and do not perform well under low ambient light conditions. Most important to this research, this system does not allow the robot to operate autonomously because the location of each robot has to be calculated on a central station. Because of these features, a MOCAP system is mostly used to provide validation data for scientific research.

Also on-board camera systems are used to assist in the localization of robots [15]. Since actual localization requires reference-points to get an absolute location, cameras are mostly used to estimate velocity. For quadrotor applications, this is done by using optical flow of a camera that is attached to the bottom [16]. Another use is for obstacle detection and avoidance and most commercial quadrotors use camera systems for both of these applications [17]. Other indoor localization methods use signal emitting beacons for localization. In almost all research on autonomous robots, RF signals are used as a way of communicating with each other or with a central station. As discussed in [18], these wireless systems can also be used to calculate distances based on Time Of Flight (TOF) or Received Signal Strength (RSS) and perform multilateration for localization. Examples of these are cellular based [19], bluetooth based [20][21], methods based on WLAN [22][23] and methods based on UWB [14][24]. These systems all use radio signals to estimate the location of a robot in a certain space. With exception of the UWB-based systems, they have an accuracy in the order of meters, where UWB-based systems have an accuracy in the order of decimeters [18]. A commercial UWB-system like the Decawave DW1000 can obtain the same update-frequency as a MOCAP-system, but it largely depends on the amount of data that is to be send and the density of the beacons [25].

The example applications described in the beginning of this introduction apply certain restrictions to the system that will be used to provide position data. The ideal system

- Is able to be placed indoors.
- Is quick to set-up and robust to changes in the setup.
- Has a high update frequency.
- Does not require a central computation unit.
- Is not dependent on light conditions and Line Of Sight (LOS).
- Has an accuracy of around 10-20 centimeters.

Since an UWB-system is the only system that meets these requirements, such a system will be used as an input to our estimator. An UWB-system also has the advantage that because it operates on a broad band of RF signals, its signal appears as background noise to other systems, has a low energy consumption and almost causes no interference. It is also

possible to have multiple users use the same system simultaniously, without compromising on update frequency or accuracy. Last but not least these systems are cheap, which makes the technology accessible to a larger research community. A complete positioning system costs around 1500€ while a professional MOCAP-system costs around 150.000€. Using UWB does add some significant challenges, since the positioning error of around 10-15cm can deteriorate the position estimate, especially with a lower number of anchors and it causes biases in the position estimation that are both spatially varying and varying on the relative orientation of the quadrotor to a specific beacon.

1-2 The State-Of-The-Art of Localization Based on UWB

UWB has been successfully used in research to estimate the states of a quadrotor indoors [14, 26, 27]. In most of these papers, UWB is used to follow a reference track with a quadrotor.

Using UWB does have a couple of downsides that significantly affect the performance of the quadrotor. One of the problems of UWB is the bias in the measurements, based on the relative position and orientation to the anchor. Due to the bias in the UWB measurements, highly accurate flight of dynamic maneuvers and following a reference trajectory is hard to achieve without compensating for it in some way. Another problem is that the UWB sensors can suffer from serious outliers which have considerable influence on estimators such as an Extended Kalman Filter (EKF). This problem is especially deteriorating the estimators result when a low amount of UWB beacons is used. As is demonstrated in [14], using UWB-data with an Extended Kalman Filter (EKF) shows promising results. This method, however, does not take bias into account and uses a large amount of UWB beacons to reduce the influence of outliers. In work by the same research group [15], a camera was added to the quadrotor to estimate the ranging bias. Although the mean absolute positioning error was decreased from 0.30m to 0.12m, the solution requires a powerful CPU and a mounted camera which will not be available to a micro-quadrotor.

Currently, the estimators that are in use are not sufficient for this particular task. They only perform well in small areas with 5 or more UWB beacons. Also, they are heavily influenced by outliers and bias that is inherent to the UWB range measurements. Because of this, the solutions do not offer an accurate and stable estimate of the state with this setup, which makes reference tracking a difficult task.

1-3 Research Question & Scope

The goal of the research described in this thesis is to enable accurate localization and control of robotic agents in an area with a low beacon coverage that is also cheap to implement. First, the research objective will be described and after this, the scope of this research will be further narrowed and explained.

The main research objective will be to:

Design an estimation method that uses a time-window of range measurements for a micro-quadrotor by using Ultra Wideband (UWB) measurements

The research scope will be further narrowed down below.

The robot that is considered in this research will be a quadrotor. Due to the improvements in battery quality and other technology, quadrotors are a suited platform for robotic research, both indoors and outdoors. The popularity of quadrotors in research is also because the system is mechanically easy to understand but offers a lot of challenges in both control and estimation nonetheless. The specific quadrotor that will be used in this research is the Crazyflie 2.0 that is produced by Bitcraze [28]. Since this micro quadrotor is completely open-source, necessary adaptations can be easily made.

Considering the task that should be performed by our micro-quadrotor, our estimator will be limited to estimating the (x, y)-position of micro-quadrotors in an indoors environment. Since the quadrotor will be operating at a constant height, the z-height of the quadrotor is assumed to be constant and known to the estimator. Since the method should be usable in large indoor areas, it will be assumed that the quadrotor has access to the distance measurements of only four beacons. This amount should theoretically be enough to estimate the position of a robotic agent. The quadrotor does not need to perform aggressive maneuvers and will need to fly a reference track with low velocities. Since there is limited space and capacity on the quadrotor, the on-board estimator will only have access to the onboard IMU and UWB receiver.

It will be assumed that the operation space in the X-Y plane of the quadrotor will be within the convex hull that is spanned by the (x,y)-locations of all UWB beacons.

$$\boldsymbol{p}_{\text{des},k} \in \text{Conv}(S) \tag{1-1}$$

where $\boldsymbol{p}_{\text{des},k}$ is the desired position at time k and S is the set of all (x,y)-locations of all beacons and $\text{Conv}(S)$ is the convex hull of set S. Because research shows that measurement data quickly degenerates when the robot leaves the convex hull within the beacons, this research will only focus on movement within the convex hull. The method that will be designed should be able to run in real-time on the Crazyflie 2.0. This is important to enable autonomous flight, since no communication should be necessary with a central computer. In order to achieve this, an iterative method such as in [29, 30] could be investigated. Another option would be to use a linearized simplification of the dynamic system, in which case an adaptation of the MHE problem can be solved in a single iteration. An estimation method in this scope resembles the use of robotic agents for multiple tasks, including persistent coverage tasks in greenhouses with quadrotors or order fulfillment in warehouses.

Because UWB beacons are known to produce biased sensor readings, a second research objective is also defined:

Design a low-dimensional bias model that can be used to reduce the error in the ranging measurements

If it is possible to create a simple model that estimates the bias, this can be implemented in the estimator to improve the accuracy. This would eventually enable in-flight estimation of the bias model of a particular setup, although the design of such an estimator is out of scope of this research.

1-4 Main Contributions

The main goal of the research described in this thesis is to design an estimation method for a quadrotor. This estimation method is to be used in indoor environments with a low amount of beacons. The main contributions of this thesis are summarized below:

- ***Design and investigation of a bias model for UWB sensors***
  This work shows a way of analyzing the bias characteristics of individual UWB sensors and reducing this bias by fitting a three-dimensional model. The bias-model reduces the bias of a sensor by up to 65% and is implementable in a real-time estimator because of the low dimensionality.
- ***Implementation of a simulation environment for quadrotors***
  This simulator can be used to quickly generate data-sets that include both the real states and the noisy measurements. This data can be used to simulate the estimator responses and allows for quick comparisons between estimators in the presence of certain noise models for any commanded flight-trajectory.
- ***Design and implementation of the Moving Horizon Model Parametrization (MH-MP) that uses range measurements***
  A novel solution for using UWB range measurements for quadrotor state-estimation has been devised that uses a time-window of measurements. This approach deduces position estimates from range-measurements and calculates the offset of a predicted state-trajectory with a least-squares solution. From simulation and real test results presented in this thesis, it follows that the MH-MP estimator is outperforming the benchmark EKF estimator in our setup.

1-5 Thesis Outline

Chapter 2 gives an introduction of the Ultra Wideband (UWB) technology to the reader. In this chapter, a description of how these signals are used for localization is given and the characteristics of the UWB-system that will be used in the tests are investigated. This includes the calibration of the sensors, calculation of the noise statistics and an investigation in the bias characteristics.

In Chapter 3 models are studied for both the dynamics of the quadrotor and the sensors that are used. Besides that, the simulation environment is discussed that was designed to conveniently create datasets of both the quadrotor states as the measurements that can be used to test the designed estimators. After this, the subject of our research is discussed; the Crazyflie 2.0. It displays the technical specifications of the quadrotor but also explains the estimators that are currently implemented. These include the implementation of the Extended Kalman Filter (EKF), which is currently the default when a positioning system is supplied, and complementary filters that are used when no positioning system is supplied and that only produce attitude estimates.

Chapter 4 gives the derivation of the estimator, including methods of using the range measurements, outlier rejection and prevention of overfitting.

Chapter 5 comprises the results of the designed estimator. First, the results of the simulations are shown and after this the results of the real tests are discussed.

In Chapter 6 the results of this thesis are summarized and the possibilities for future work are discussed.

Chapter 2

Ultra Wideband (UWB)

Ultra Wideband (UWB) is a radio technology that can be used to create range-measurements. This chapter discusses the localization features of Ultra Wideband (UWB) and how this relates to our setup. Section 2-1 describes how UWB signals can be used to create a range measurement.

2-1 Localization With UWB

This section investigates how UWB signals are used to create range measurements. This will be done by examining two common lateration methods, Time Of Arrrival (TOA) and Time Difference Of Arrival (TDOA). After this, ranging methods are discussed that enable the use of these lateration methods.

2-1-1 Time Of Arrrival (TOA)

The TOA method uses the departure and arrival times of a signal from transmitter to receiver to estimate the distance between both. When the locations of the transmitters are known and sufficient transmitters are used, an unambiguous estimate of the receivers location can be made. For 2D, at least three transmitters are necessary while for 3D, this number is four. Each transmitter sends a signal with the time of departure (t_d). The receiver can now calculate its distance to a transmitter by comparing the time of departure to the time of arrival

$$D = c(t_a - t_d) \tag{2-1}$$

Where c [m/s] is the speed of light and $t_a - t_d$ [s] is the time of flight. This distance is also equal to the Euclidean distance between the position of the transmitter ($\boldsymbol{x}_T$) and receiver ($\boldsymbol{x}_R$)

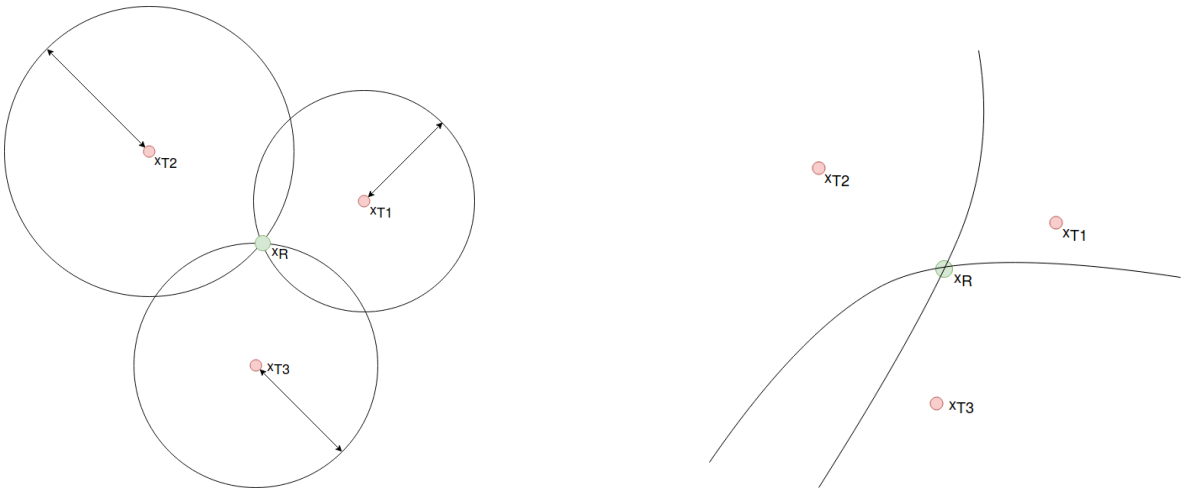
$$D = ||\boldsymbol{x}_T - \boldsymbol{x}_R|| \tag{2-2}$$

Essential to this method is that the clocks of the transmitter and receiver are synchronized. If this has not been done, an extra term $\theta_{\text{offset}}c$ has to be added to the distance measurement in Equation 2-1 and it will result in the following equation

$$D = c(t_a - t_d) + \theta_{\text{offset}}c \tag{2-3}$$

Because EM waves propagate with speed of light, small errors in clock-time can result in large errors in the distance measurement as can be seen in Equation 2-3 (1 ns results in an error of 30cm). Different ways of achieving clock synchronization will be discussed in Subsection 2-1-3.

If all the measurements are free of error and if enough beacons are used, a set of equations can now be solved to find $\boldsymbol{x}_R$. This is visualized in Figure 2-1a.



(a) Localization with Time Of Arrrival (TOA)-measurements in 2D

(b) Localization with Time Difference Of Arrival (TDOA)-measurements in 2D

Figure 2-1: Time Of Arrrival (TOA) and Time Difference Of Arrival (TDOA) trilateration

But in case of noisy measurements (in real life always the case) an optimization problem needs to be solved to obtain the most probable location of the receiver. Although 4 beacons are sufficient for an unambiguous 3D position measurement, more beacons can be added to the setup. By combining more measurements in an optimization problem, more robust measurements can be obtained [31]. This is because in the case of four beacons, every measurement is necessary to obtain a position estimation. If one of the measurements does not arrive or is discarded as an obvious outlier (see Section 4-2-2), no position estimate can be obtained.

The TOA measurements can also be used directly in an estimator, as is shown in 3-4-2.

2-1-2 Time Difference Of Arrival (TDOA)

The Time Difference Of Arrival (TDOA) method uses the same signals, only now compares arrival times between two transmitters. The point $\boldsymbol{x}_R$ is no longer on a circle around a transmitter, as can be seen in figure 2-1a, but on a hyperboloid between two transmitters. This can be seen in Figure 2-1b.

The hyperboloid follows from the distance equation between two transmitters

$$R_{i,j} = ||\boldsymbol{x}_T - \boldsymbol{x}_{R_i}|| - ||\boldsymbol{x}_T - \boldsymbol{x}_{R_j}|| \tag{2-4}$$

One mayor advantage of TDOA on TOA is that the robot does not need to have it's clock synchronized to those of the transmitting beacons. This way the clock offset between transmitter and receiver drops out of the equation (assuming the transmitters do have synchronized clocks). Because both distances are subtracted from eachother, the clock-offset between the transmitters and the receiver drops out of the equation. This is because the term $\theta_{\text{offset}}c$ in Equation 2-3 is the same for both transmitters. It has to be noted that this assumes that all beacons have synchronized their clocks.

Calculating the intersection point of hyperbola is sensitive to measurement noise [32], so using more than the necessary 3 beacons is desirable for the TDOA method.

2-1-3 Ranging Methods

To obtain the necessary estimates of the values used in both methods described above, multiple methods exist.

In [14] a method based on Two Way Ranging (TWR) is implemented to localize quadrotors indoors. This method requires a message to be sent from the tag to the receiver and then wait for its response to calculate a TOA measurement [33]. The TWR method is visualized in Figure 2-2a. In this figure, t_{f} is the time of flight of the signal defined as $t_a - t_d$ in Equation 2-1, t_{delay} is a predetermined time that the beacon has to wait before sending a reply and t_{round} is the total roundtrip time.

Since the value t_{delay} is known by the robot, the time of flight t_{f} can now be calculated as

$$t_{\text{f}} = \frac{1}{2}(t_{\text{round}} - t_{\text{delay}}) \tag{2-5}$$

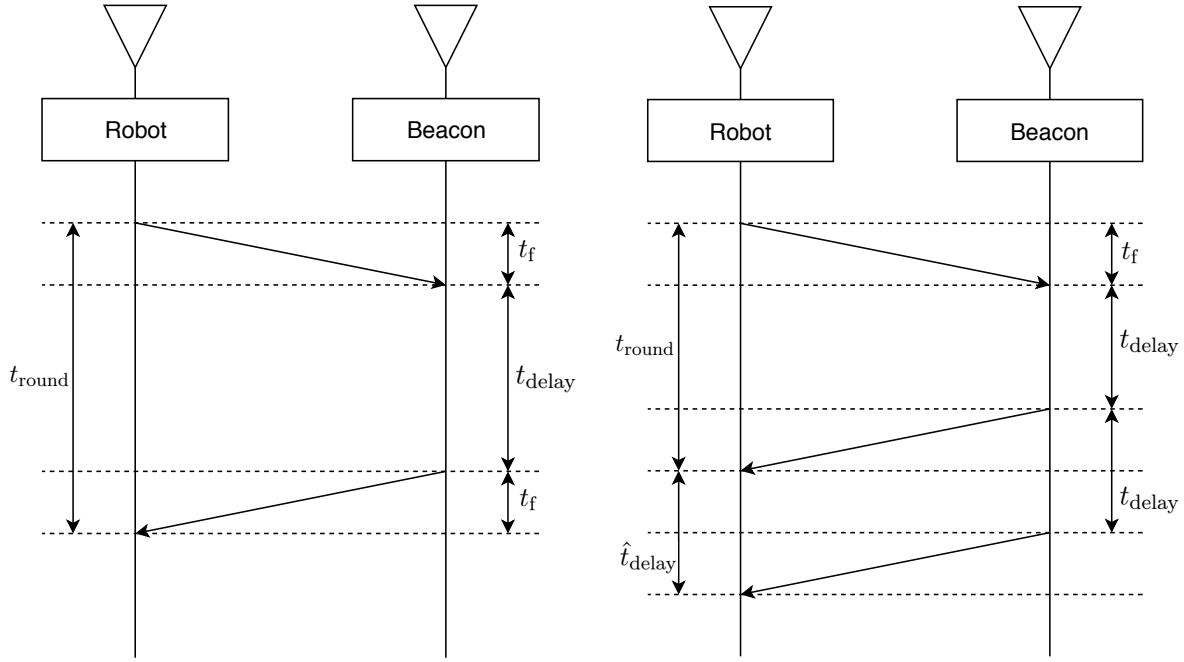
The main problem with this method is that the measured delay at the robot cannot be assumed to be exactly t_{delay}. In [14], it was found that due to a frequency drift and noise in the crystal oscillator that drives the clock, the expected delay can differ up to more than 3ns from the chosen value t_{delay}, which might result in range measurement errors of 1 meter. An extension to the TWR to overcome this is also proposed in this paper.

The TWR method in this paper is extended by a repeated reply, that is used to cancel out the error between actual and expected delay. This method is visualized in Figure 2-2b.

Instead of subtracting the value t_{delay} from t_{round}, the time delay $\hat{t}_{\text{delay}}$ can now be used to estimate the difference between the delay and the expected delay. The time of flight can now be calculated as

$$t_{\text{f}} = \frac{1}{2}(t_{\text{round}} - \hat{t}_{\text{delay}}) \tag{2-6}$$

In [14], 5 transmitting beacons were used, which is one more than will be used in our own research. The states were then estimated by using an Extended Kalman Filter (EKF). The



(a) Visualization of the Two Way Ranging (TWR) method

(b) Visualization of the Two Way Ranging with Repeated Reply (TWR-RR) method

Figure 2-2: Visualization of the Two Way Ranging (TWR) and Two Way Ranging with Repeated Reply (TWR-RR) methods

accuracy that was obtained had a RMSE of 13cm. This method now has significant improvements over the previously described method, since it has a 3D location estimation, incorporates the dynamic model of the quadrotor and does not make use of a central calculation hub. The main disadvantage is that still two-way communication between tag and beacon is necessary.

One method to perform localization without two-way communication is a one-way-ranging method as described by Ledergerber et al. [27]. In this method, the anchors are first synchronized by a TWR-method, and then send their messages. This way, the tag receives messages all in the time of one anchor and can use this to calculate TDOA measurements. This method enables the use of a large amount of robots in the same operating space and can be easily scaled to larger areas by adding more beacons.

2-2 UWB Characteristics

The UWB system that will be used in our experiments is the Loco Positioning system by Bitcraze. This system is chosen because it can be used as a plug-and-play system along with the Crazyflie but still offers the possibility for modifications in its open-source software. The system used consists of four Loco positioning nodes and a Loco positioning deck. Both the nodes and the deck use a Decawave DWM1000 UWB module. The nodes and deck all use a TWR method to gather TOA measurements. In this configuration, it is only possible to use 1 deck since the deck is actively communicating with the nodes. It should be noted that

in order to use multiple decks in the same space, one of two TDOA methods, implemented by Bitcraze, could be used. For these tests, the TWR-method was chosen because it offered a reliable and quick solution. Another consideration should be that although theoretically ranges of up to 100 meters should be achievable with the DWM1000, but due to regulation the power output of the sensors is limited, which significantly reduces the maximum obtainable range.

To gain insight in the statistical characteristics of the UWB-method, measurements were performed. These measurements aim to create a complete measurement-model that can later be used in our estimators. Subsection 2-2-1 shows how the sensors are calibrated and shows the statistical properties that follow after the calibration. In Subsection 2-2-2, the distance bias characteristics based on relative orientation will be compared to results found in literature and the spatially-varying bias will be analysed and mapped.

2-2-1 Sensor Calibration

The TWR implementation on the Crazyflie considers one calibration parameter, which is the antenna delay $\Delta t_{\text{antenna}}$ [s] which can be seen in Figure 2-3.

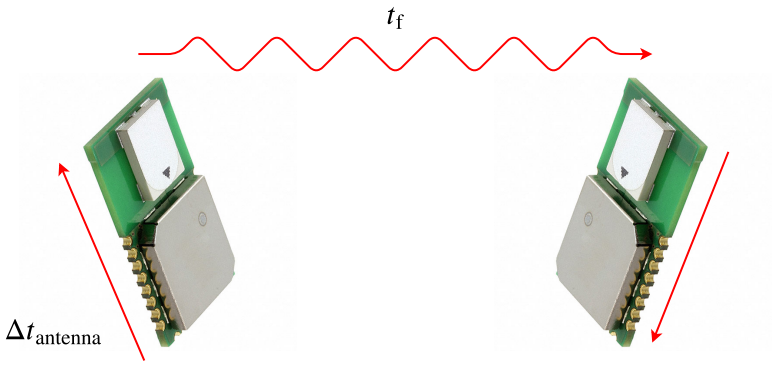


Figure 2-3: The antenna delay can be seen as the time it takes for a signal to be created by the processor and sent to the antenna.

This delay represents the time it takes from when the processor sends a signal to the antenna, until the antenna creates the EM-pulse that is tranmitted. In the TWR implementation, this antenna delay is included in the calculated TOF $\hat{t}_f$, but as can be seen in Figure 2-3, the real t_f does not include this delay.

The Crazyflie code uses the assumption that the antenna delay of all different nodes is equal, and so it can be estimated by using only 1 Loco Positioning Node and the Loco Positioning Deck on the Crazyflie. Since we will be flying the quadrotor mainly in the center of a cube spanned by sides of around 10m, the sensors will be calibrated at around $\sqrt{2}(\frac{1}{2}10) \approx 7$m which is the distance from one of the corners to the center of the cube. This way we guarantee that the sensors are as accurate as possible at the most visited locations.

By gathering data at the specified range, the antenna delay can now be estimated by defining the antenna delay as

$$\Delta t_{\text{antenna}} \approx \frac{\bar{e}_{\text{dist}}}{2c} \tag{2-7}$$

Where $\bar{e}_{\text{dist}}$ [m] is the mean measured error of the distance between node and tag over a certain time-window and c [m/s] is the speed of light. The results of these measurements are depicted in Table 2-1.

	anchor 1	anchor 2	anchor 3	anchor 4
$\bar{e}_{\text{dist}}$	154.52m	154.71m	154.72m	154.78m

Table 2-1: Distance error and antenna delay

The average value of all four beacons is used in the Crazyflie code to compensate for this antenna delay. This is done by adjusting the `ANTENNA_OFFSET` parameter in the `locodeck.c` file in the Crazyflie firmware. It should be noted that this delay is different for all four anchors and therefor small constant errors are introduced for certain anchors.

With the calibrated sensor, the noise characteristics can be determined. Especially the standard deviation of the range measurements is important in the design of estimators. The noise will be modeled as a zero-mean normal distribution, so after calibrating the antenna delay this is straightforward by taking a large sample of stationary data at the calibration distance and fitting a normal distribution to it. This can be done using the same datasets as were used for Table 2-1, and subtracting the average and calculating the standard deviation of the resulting data. The resulting standard deviation was found to be $\sigma_{\text{UWB}} = 0.0225$m, which is comparable to the standard deviation calculated in [27]. In this research the standard deviation was calculated to be 0.025m.

2-2-2 Bias

As stated above, the error of a UWB measurement is usually assumed to be zero-mean white Gaussian noise [14]. This assumption is experimentally found to be incorrect since there are obvious biases in the measurements related to the relative pose and location of the receiver of up to 0.3 m [27].

The bias characteristics will be determined for the same setup that is in scope of this research. Therefor, four Loco Positioning Nodes will be placed on the corners of the Cyberzoo, which is a (10m $\times$ 10m $\times$ 10m) area dedicated to robotic research in the Aerospace Department at the Delft University of Technology (TU Delft). These nodes come with open-source software to control the DWM1000 module that is attached to it.

Bias based on relative orientation

The relative angle of the tag to the anchor is known to cause an offset on the distance measurements [27]. This offset is supposed to be introduced by a non-isotropic radiation pattern of the antennas. In [1], this offset was measured for a couple of relative angles, as can be seen in Figure 2-4.

This shows that the antenna and the position of the sensor on the quadrotor heavily influences the distance measurement.

To compensate for this, in the same research an error method was designed that uses a sparse pseudo-input Gaussian process to predict the error and noise based on the relative location

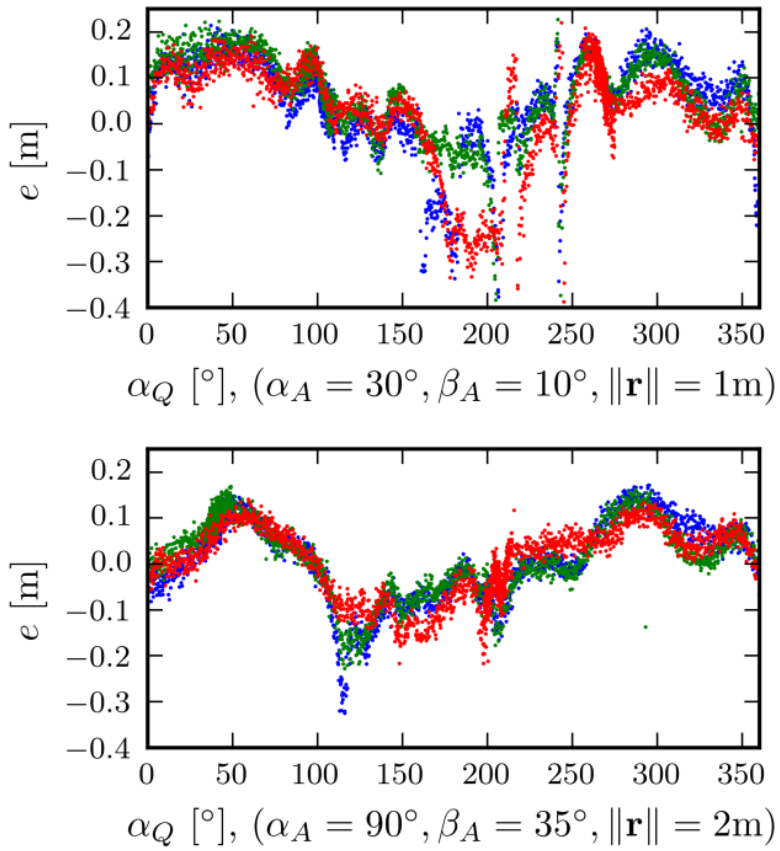


Figure 2-4: Range measurement errors for specific relative azimuth angle α_A and elevation angle β_A for different antenna pairs [1]

and pose to of the receiver was shown to be able to reduce the RMSE with almost 50%. This was done with a Two Way Ranging (TWR) method however, but it can be assumed that with one-way communication the same improvement can be obtained.

Although the same Partron antenna is used here as in our research, it is clear that the error even varies from antenna to antenna. It is important to know this noise is present in our own setup as well and therefor the ranging error for a changing relative orientation will be measured at the nominal operating distance ($\approx$ 7m). This will be done in the same orientation that will be used in the real tests, with the tag mounted horizontally beneath the Crazyflie and the anchor positioned vertically. The results, as can be seen in Figure 2-5, show that also in our setup large errors are caused by the relative orientation.

Even small angles can cause a significant offset so this will influence the estimator results.

Spatially-varying bias

Besides the bias caused by relative orientation, also a spatially varying bias can be observed. This bias is caused by how the room and the sensors are situated. Multiple causes for this effect

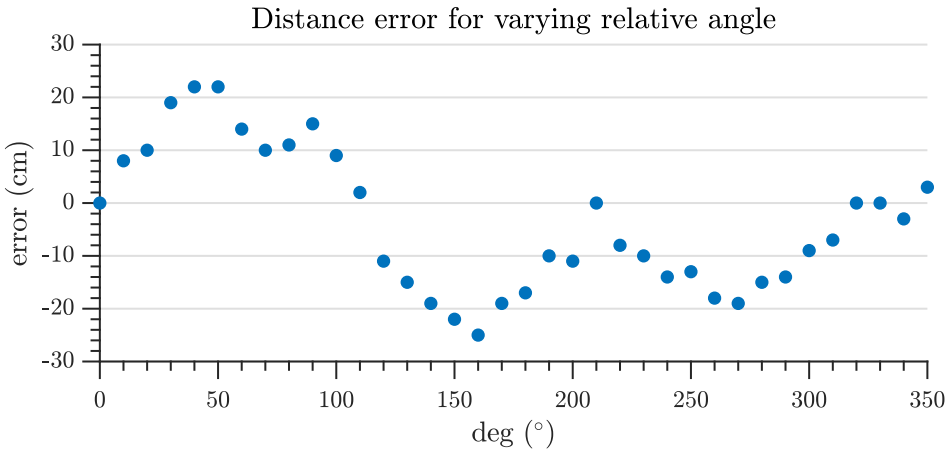


Figure 2-5: Range measurement errors for the test setup at 7 meters distance

can be given, such as the location of the anchor in the room that could cause multipathing or other electronic devices that interfere with the signals. As is researched in Hoeller et al. [15], this spatially varying bias is constant and considering that the same space is used for further experiments, can be mapped accordingly. In Hoeller et al. [15], the spatially-varying bias is taken into account by dividing the research area (3m × 3m × 3m) in a grid of cubes with a span of 0.2m. By combining the UWB-system with an on-board camera, the bias can be learned while in flight and be used for subsequent flights when the on-board camera is not available (due to technical failure or under ambient light conditions). Since in our research the research area is much larger and an on-board camera is not considered, other ways of dealing with the bias need to be addressed.

The bias will be estimated by using the OptiTrack-system in the Cyberzoo as ground-truth while the range measurements will be stored on an SD-card that is mounted on the Crazyflie. The location of the anchors is measured in a right-handed frame and are defined as follows:

	$p_{\text{node 1}}$	$p_{\text{node 2}}$	$p_{\text{node 3}}$	$p_{\text{node 4}}$
x	-4.72	-4.82	4.90	4.80
y	-4.54	5.00	5.03	-4.61
z	2.67	1.66	0.97	0.28

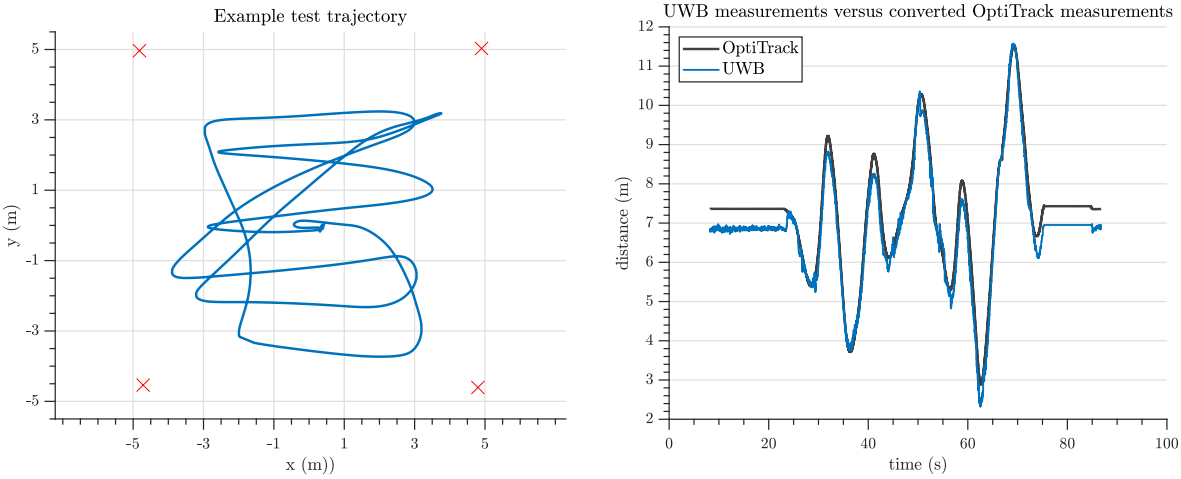
Table 2-2: Anchor positions

As can be seen in Table 2-2, the difference in x and y is much larger than in z. This has its consequences for the variance of estimators in the z direction.

The data is gathered by flying through the entire Cyberzoo with a fixed yaw and approximate z-height of 2 meters for 50 seconds. The ground truth for the range measurement at time k $\hat{d}_{\text{opti, k}}$ is calculated by transforming the OptiTrack measurements to range measurements for a specific anchor as follows:

$$\hat{d}_{\text{opti, k}} = \|\boldsymbol{p}_{\text{opti, k}} - \boldsymbol{p}_{\text{node i}}\|_2 \tag{2-8}$$

Where $\boldsymbol{p}_{\text{opti, k}}$ is the (x, y, z) OptiTrack measurement at time k. The measurement error $e_{\text{i, k}}$



(a) OptiTrack data of an example trajectory for bias estimation. UWB Anchors shown with red crosses

(b) UWB range measurements versus converted OptiTrack measurements

Figure 2-6: Ultra Wideband (UWB) bias test data visualized

can now be calculated by subtracting the ground truth from the distance measurements at node i.

In Figure 2-6a one of the flight trajectories is shown. In Figure 2-6b the range measurements for anchor 2 for this dataset can be seen compared to the ground-truth OptiTrack data

In Figure 2-7 the error $e_{\mathrm{i,\,k}}$ is shown. In this figure it is clear that the error is not zero-mean and a further analysis is necessary.

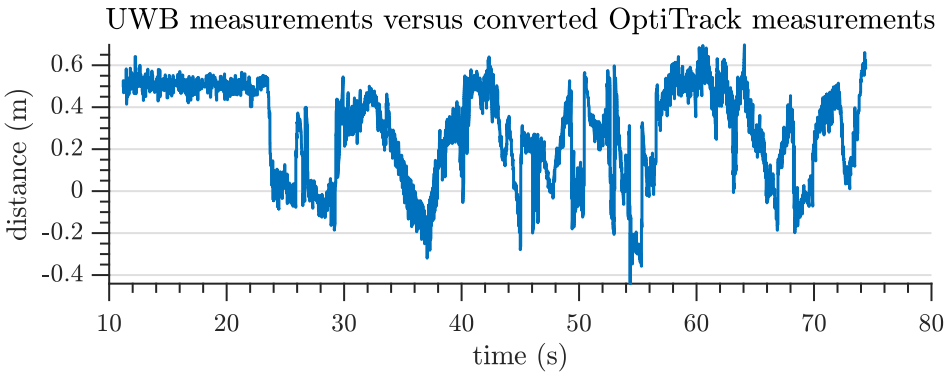


Figure 2-7: Error between UWB measurements and OptiTrack ground-truth

As was mentioned in [15], a part of the bias is spatially varying. To analyse this, the error e_{i} is binned over the (x, y)-plane into squares of 0.5m. In Figure 2-8, the average measurement error of each of the bins is shown in a heatmap

From this figure it is visible that for example for anchor 2, the bias is especially large close to the anchor and decreases with a larger distance. This suggests that a simple model can be used to reduce the influence of the bias.

The model that is suggested is a plane, based on the (x, y)-location of the quadrotor. This model is fitted by the following minimization:

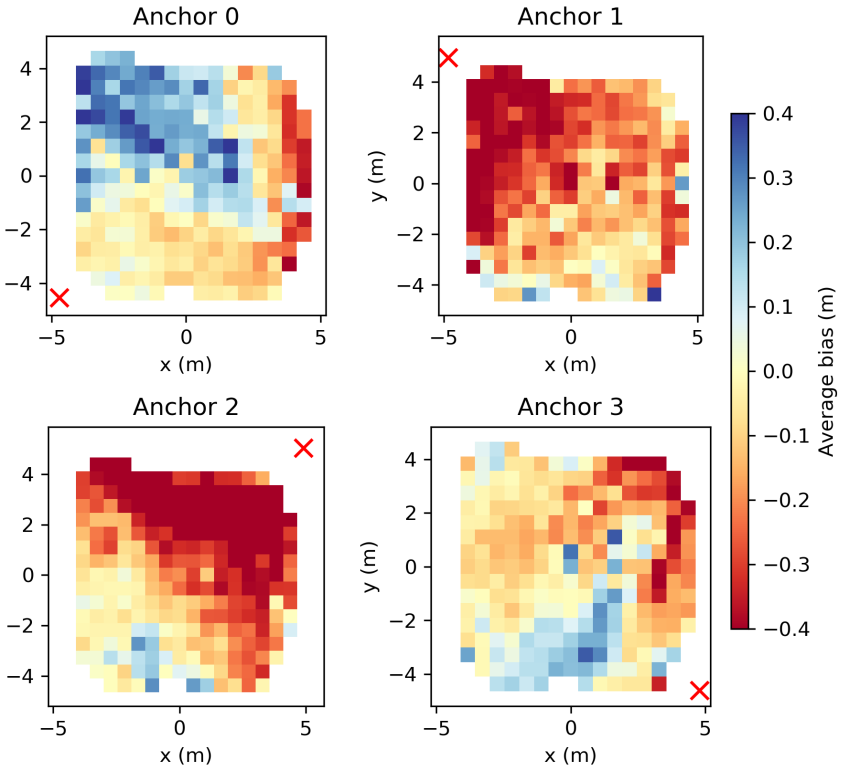


Figure 2-8: Average measurement error in $(0.5\text{m} \times 0.5\text{m})$ bins for each of the four anchors. The position of the anchor is depicted with a red cross.

$$\min_{a,b,c} \sum_{i=1}^{N} \sum_{j=1}^{N} (b_{i,j} - (ax_i + by_j + c))^2 \tag{2-9}$$

Where (x_i, y_j) is the (x, y) value in the middle of bin (i, j) and $b_{i,j}$ is the mean bias value at bin (i, j) and N is the number of bins in one direction. This minimization can be described as a linear-least squares problem $\min A\boldsymbol{x} - \boldsymbol{b} = 0$ with A and b as:

$$A = \begin{bmatrix} x_1 & y_1 & 1 \\ x_1 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_1 & y_N & 1 \\ x_2 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_N & y_{N-1} & 1 \\ x_N & y_N & 1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} b_{1,1} \\ b_{1,2} \\ \vdots \\ b_{1,N} \\ b_{2,1} \\ b_{2,2} \\ \vdots \\ b_{N,N-1} \\ b_{N,N} \end{bmatrix} \tag{2-10}$$

And the solution can be found as $(a, b, c) = (A^T A)^{-1} A^T \boldsymbol{b}$. This plane can be used to correct the distance measurements based on an estimated position in an estimator.

This plane will be fitted on training data and subsequently tested on validation data. In Figure 2-9, the results of this compensation on the validation data can be seen. Especially the areas close to beacons 1 and 2 have a much smaller average bias.

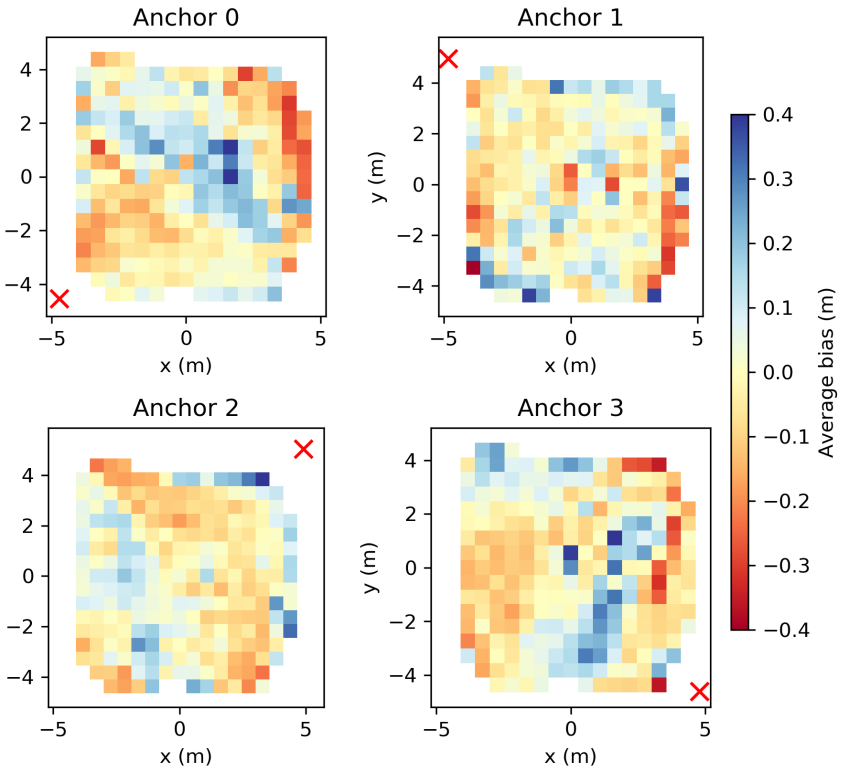


Figure 2-9: Average measurement error in $(0.5\text{m} \times 0.5\text{m})$ bins for each of the four anchors after compensation. The position of the anchor is depicted with a red cross.

In order to make a comparison between the situation before and after compensation, the Frobenius norm and the standard deviation are used as a metric. The Frobenius norm of a matrix A is defined as:

$$\|A\|_F = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} a_{i,j}^2} \tag{2-11}$$

In Table 2-3 the calculated parameters for our system with the test data can be seen.

In Table 2-4 the Frobenius norm and the standard deviation of the binned validation datasets are shown before and after compensation. As was expected from Figure 2-8, especially Anchor 1 and Anchor 2 show a large improvement of 55% and 65% respectively in the Frobenius norm and 25% and 47% respectively in the standard deviation. It is important to add that both anchors that benefit most from the compensation have a high c value, which is independent of the x and y position. This means that the bias compensation also accounts partly for small

	Anchor 0	Anchor 1	Anchor 2	Anchor 3
a	0.0300	-0.0321	-0.0592	-0.0275
b	-0.0331	0.0301	-0.0345	-0.0211
c	0.0546	-0.2247	-0.2309	-0.0491

Table 2-3: (a, b, c) parameters for all four anchors in our setup

calibration errors, which is beneficial in the Locodeck system since all anchors are calibrated at once.

	Anchor 0	Anchor 1	Anchor 2	Anchor 3
$\|A\|_F$ before	2.928	4.146	4.739	2.690
$\|A\|_F$ after	2.043	1.883	1.672	2.058
σ before	0.182	0.163	0.202	0.164
σ after	0.131	0.121	0.107	0.132

Table 2-4: Frobenius norm and standard deviation before and after compensation

Currently this bias compensation method requires data to be acquired and compared to ground-truth OptiTrack data. The optimization problem itself could be implemented on the Crazyflie. It should also be noted that the estimated parameters will vary when the setup of the UWB-beacons changes.

2-3 Concluding Remarks

In this chapter, Ultra Wideband (UWB) is shown to be a promising technology that can be used to obtain range measurements in an indoors environment. In order to design an estimation method, it is important to know what kind of noise the measurements are dealing with. It is well known from literature that UWB-measurements are biased and to investigate the cause and size of these biases some properties of the system are discussed. This includes the influence of the difference in frequency of the crystal oscillators in the micro-processors. Besides this, biases are introduced by the non-isotropic radiation pattern formed by the antennae. This theoretical knowledge was applied to the Loco Positioning System from Bitcraze and the system was calibrated and tested. Besides, a simple model was designed to reduce spatially-varying bias. This model was verified on validation data and was shown to reduce the measurement error with up to 65%.

With a thorough analysis of the UWB-system, it is possible to include these measurements in an estimator for the research platform that is the subject of this thesis, the quadrotor.

Chapter 3

Quadrotors

In order to design an estimation method, extensive knowledge of the estimators subject is necessary. This chapter will discuss the dynamics of a quadrotor, as highlighted in Figure 3-1. In this chapter, the dynamic model of a quadrotor will be investigated in Section 3-1. Models for the different sensors that are used on the quadrotor are explained in Section 3-2. In Section 3-3, all this knowledge will be combined to create a simulation environment, that allows the creation and adaptation of test-data quickly.

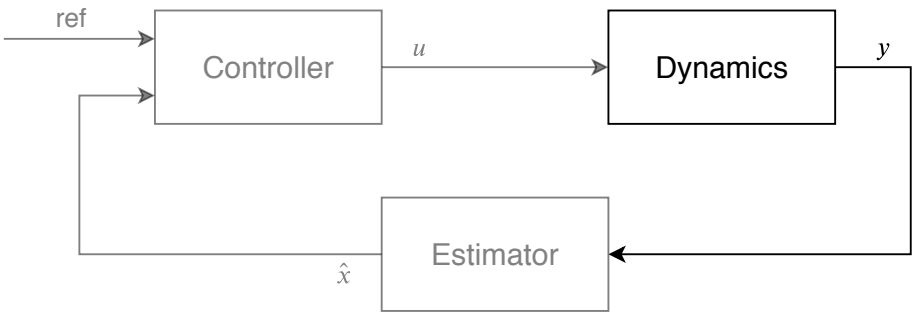


Figure 3-1: Quadrotor control loop block diagram

3-1 Dynamic Model

This section will discuss the dynamic modeling of a quadrotor. A short derivation of the commonly used equations of motions will be presented and the specific design choices will be discussed. We will be looking into models that will have the following state-space form:

$$\dot{x}(t) = A(x)x(t) + B(x)u(t) \tag{3-1}$$
$$y(t) = C(x)x(t) \tag{3-2}$$

Where $x(t) \in \mathbb{R}^{n_x}$ are the states of the system with n_x the number of states, $u(t) \in \mathbb{R}^{n_u}$ is the input signal that is sent to the system with n_u the number of inputs and $y(t) \in \mathbb{R}^{n_y}$ are the measurements with n_y the number of measurements. $A(x) \in \mathbb{R}^{n_x \times n_x}$, $B(x) \in \mathbb{R}^{n_x \times n_u}$ and $C(x) \in \mathbb{R}^{n_y \times n_x}$ are the state-space matrices.

The model that will be discussed are rigid-body dynamics based on Tait-Bryan angles. Although this representation is easily interpretable, the trigonometric terms may lead to singularities. To overcome the risk of reaching a singularity, multiple solutions are discussed in literature. The first one is by using the Multiplicative Extended Kalman Filter (MEKF) that will be discussed in section 3-4-2. Another method is implementing a model based on quaternion dynamics. By using quaternions, rotations can be described without using trigonometric terms and without risking singularities. After deriving the model based on Tait-Bryan angles, an extension to this model with quaternions will be discussed to remove the singularities.

3-1-1 Rigid-body dynamics based on Tait-Bryan angles

Rigid-body dynamics based on Euler rotations are used very often in literature. Although various forms are used, the ZXY [34] and ZYX [35, 4] notation are most common. The only difference between either notation is the form of the rotation matrix, and for sake of brevity, only the ZXY notation will be shown. In this thesis, vectors will be presented in bold and their dimension will be presented as well. Derivatives will be presented with the dot-notation ($\dot{\boldsymbol{x}}$) and estimates will be presented with the hat-notation ($\hat{\boldsymbol{x}}$). The (x, y, z)-location of center of mass of the quadrotor in a global Cartesian frame will be denoted as $\boldsymbol{p}$ [m], $\boldsymbol{\Psi}$ [rad] the extrinsic ZYX Tait-Bryan angles (ψ, ϕ, θ), $\boldsymbol{\omega}$ [rad/s] are the angular velocities of the quadrotor in the global frame and $\dot{\theta}_i$ [rad/s] is the angular velocity of rotor i.

$$\boldsymbol{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \ \boldsymbol{\Psi} = \begin{bmatrix} \psi \\ \phi \\ \theta \end{bmatrix}, \ \boldsymbol{\omega} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}, \ \boldsymbol{\Theta} = \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix} \tag{3-3}$$

Two right-handed reference frames will be used to describe the dynamics, one for the body-fixed frame $\mathcal{B}$ and one for the inertial-frame $\mathcal{G}$. These frames can be seen in figure 3-2. It can be seen that the so-called 'X'-configuration is used as opposed to the '+'-configuration. In this configuration, the forward facing x-axis is not chosen to be parallel with the arms of the quadrotor. Two reasons for this decision in literature are that more rotational acceleration can be obtained around the principle axes of rotation, and a camera can easily be added to the 'X'-configuration, while in the '+'-configuration, the arms and motors would obstruct vision.

Each frame is represented by a set of vectors $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ with a subscript describing to which frame it belongs. The frames are related by the rotation matrix R, which transforms a vector in $\mathcal{B}$ to $\mathcal{G}$. This rotation matrix is obtained by using the ZXY Tait-Bryan angles. This is equal to a rotation of ψ around $\boldsymbol{z}_{\mathcal{G}}$, then ϕ around the x-axis of the newly obtained frame and finally θ around the new y-axis. These rotations can be seen in Figure 3-3. This results in the following rotation matrix [34]:

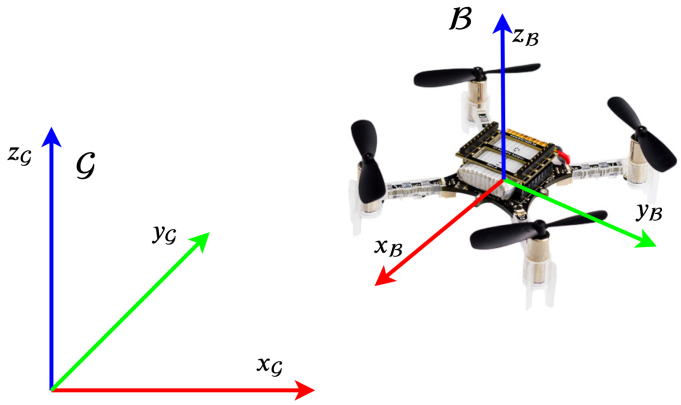


Figure 3-2: quadrotor frames $\mathcal{G}$ and $\mathcal{B}$

$$R = \begin{bmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi + c\psi s\phi s\theta & c\phi c\psi & s\psi s\theta - c\psi c\theta s\phi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix} \tag{3-4}$$

Where c and s are the cosine and sine functions.

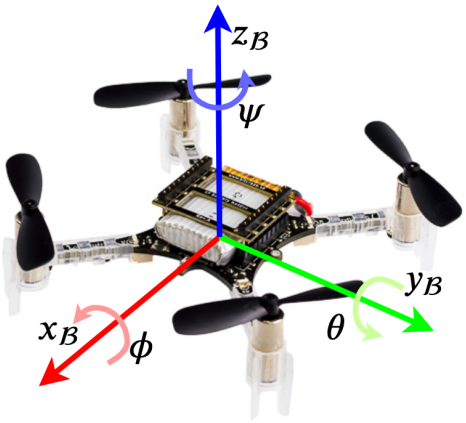


Figure 3-3: Tait-bryan angles visualized

Although the frame might deform due to the thrust generated by the rotors, the quadrotor is modeled as a single rigid-body with the thrust of the rotors pointing in the upper z-direction of the body-fixed frame $\mathcal{B}$.

The rigid-body equations of motion can now easily be derived as [34]

$$m\ddot{\boldsymbol{p}} = m\boldsymbol{g} + R(f\boldsymbol{e}_{z_\mathcal{B}} + \boldsymbol{f}_\mathrm{a}) \tag{3-5}$$
$$\dot{R} = R[\boldsymbol{\omega}]_\times \tag{3-6}$$
$$I\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times I\boldsymbol{\omega} + \boldsymbol{\tau} \tag{3-7}$$

Where the mass of the quadrotor is denoted with m [kg], the thrust force with f [N], $\boldsymbol{e}_{z_\mathcal{B}}$ is the unity vector pointing in the z-axis of frame $\mathcal{B}$, I [kg $\cdot$ m^2] is the inertia matrix, $[\boldsymbol{\omega}]_\times$ is the skew-symmetric matrix containing the angular velocity of the quadrotor such that $[\boldsymbol{\omega}]_\times \nu = \boldsymbol{\omega} \times \nu$ where $\times$ is the vector cross product and $\nu \in \mathbb{R}^3$.

$$[\boldsymbol{\omega}]_\times = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{3-8}$$

The vector $\boldsymbol{\tau} = [\tau_1, \tau_2, \tau_3]^T$ is the moment applied to the quadrotor by the rotors. Considering the frame in Figure 3-2, this moment can be calculated as follows [36]

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} -dc_\mathrm{T}/\sqrt{2} & -dc_\mathrm{T}/\sqrt{2} & dc_\mathrm{T}/\sqrt{2} & dc_\mathrm{T}/\sqrt{2} \\ -dc_\mathrm{T}/\sqrt{2} & dc_\mathrm{T}/\sqrt{2} & dc_\mathrm{T}/\sqrt{2} & -dc_\mathrm{T}/\sqrt{2} \\ -c_\mathrm{Q} & c_\mathrm{Q} & -c_\mathrm{Q} & c_\mathrm{Q} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix} \tag{3-9}$$

where c_Q is a lumped torque coefficient that can be determined by static thrust tests.

The thrust force f can be calculated when the angular velocity $\dot{\theta}_i$ of the rotors is known. In this case, it can be calculated as the sum of individual forces for every rotor.

$$f_i = C_\mathrm{T} \rho A_{r_i} r_i^2 \dot{\theta}_i^2 \tag{3-10}$$

Where C_T is called the thrust coefficient, which is dependent on the shape of the rotor, ρ is the density of air, A_{r_i} is the rotor disk area and r_i is the radius of the rotor. These four parameters can be lumped into a single positive parameter c_T to result in the following equation:

$$f_i = c_\mathrm{T} \dot{\theta}_i^2 \tag{3-11}$$

This parameter c_T can be determined by performing static thrust tests. The total thrust f generated by the rotors on the system can now be described by

$$f = c_\mathrm{T} \sum_{i=1}^{4} \dot{\theta}_i^2 \tag{3-12}$$

Assuming that every rotor has the same parameter c_T.

The remaining aerodynamic effects are captured in the parameters $\boldsymbol{f}_\mathrm{a}$ in Equation 3-5. A linear model for these effects will be presented as is discussed in [14]. The aerodynamic force is assumed to be a linear combination of two proportionality constants and the product of the airspeed and propeller speeds. The $k_\perp$ constant is the constant for the force in the plane of the quadrotor and $k_\parallel$ is the constant that is proportional in the direction of the thrust ($\boldsymbol{z}_\mathcal{A}$). The total aerodynamic force can be calculated as

$$\boldsymbol{f}_\mathrm{a} = K_\mathrm{aero} \dot{\theta}_\Sigma R^{-1} \dot{\boldsymbol{p}} \tag{3-13}$$

where

$$K_{\text{aero}} = \begin{bmatrix} k_{\perp} & 0 & 0 \\ 0 & k_{\perp} & 0 \\ 0 & 0 & k_{\parallel} \end{bmatrix}, \ \theta_{\Sigma} = \sum_{i=1}^{4} |\dot{\theta}_i| \tag{3-14}$$

Using this notation, it is assumed that the relative airspeed of the quadrotor equals the quadrotors speed relative to frame $\mathcal{A}$ which is a valid assumption as long as the quadrotor is flown indoors.

3-1-2 Rigid-body dynamics based on quaternions

In order to remove singularities from the model and to decrease the computational complexity of the model, models based on quaternions is suggested in [4, 37, 29, 38]. A quaternion can be described as

$$q = a + bi + cj + dk \in \mathbb{H} \tag{3-15}$$

where $a, b, c, d \in \mathbb{R}$, and i, j, k are imaginary units. By using this notation, a non-commutative multiplication with inverses is possible, which is essential for rotations. Some useful algebraic identities are given as

$$i^2 = j^2 = k^2 = ijk = -1, ij = -ji = k, jk = -kj = i, ki = -ik = j \tag{3-16}$$

By using Euler's identity a compact vector notation with a real and imaginary part can be obtained. A quaternion vector $\boldsymbol{q}$ is defined as [29]

$$\boldsymbol{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} \cos(\alpha/2) \\ \boldsymbol{e}_{q_x} \sin(\alpha_q/2) \\ \boldsymbol{e}_{q_y} \sin(\alpha_q/2) \\ \boldsymbol{e}_{q_z} \sin(\alpha_q/2) \end{bmatrix} = \begin{bmatrix} q_w \\ \boldsymbol{q}_v \end{bmatrix} \tag{3-17}$$

where $\boldsymbol{e}_q = [\boldsymbol{e}_{q_x}, \boldsymbol{e}_{q_y}, \boldsymbol{e}_{q_z}]^T$ is the principal rotation axis and α_q is the quaternion rotation angle. It can be seen that the real part of the quaternion is $q_w \in \mathbb{R}$ and the imaginary part is $\boldsymbol{q}_v \in \mathbb{R}$. Because a four-dimensional vector is used to describe three-dimensions, a constraint is added to ensure that the components are not independent. This constraint is given as $\boldsymbol{q}^T\boldsymbol{q} = 1$. In this research, the quaternion vector is defined such that the orientation of the body-fixed frame $\mathcal{A}$ relative to the reference coordinate system is described by this vector. The same rotation that was previously described by $x_{\mathcal{B}} = Rx_{\mathcal{A}}$ can now be described by

$$\begin{bmatrix} 0 \\ \boldsymbol{x}_{\mathcal{B}} \end{bmatrix} = \boldsymbol{q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{x}_{\mathcal{A}} \end{bmatrix} \otimes \boldsymbol{q}^* \tag{3-18}$$

where $\boldsymbol{q}^*$ is the complex-conjugate and $\boldsymbol{q} \otimes \boldsymbol{q}'$ is quaternion multiplication and can be seen as the combined rotation of two quaternions $\boldsymbol{q}$ and $\boldsymbol{q}'$. Quaternion multiplication is defined as

$$\boldsymbol{p} \otimes \boldsymbol{q} = \begin{bmatrix} p_w q_w - \boldsymbol{p}_v^T \boldsymbol{q}_v \\ p_w \boldsymbol{q}_v + q_w \boldsymbol{p}_v + \boldsymbol{p}_v \times \boldsymbol{q}_v \end{bmatrix} \tag{3-19}$$

It can now be shown that $\text{Im}(\boldsymbol{q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{x}_{\mathcal{A}} \end{bmatrix} \otimes \boldsymbol{q}^*) = \boldsymbol{x}_{\mathcal{B}}$.

The quaternion time-derivative can be found in terms of the angular velocity $\boldsymbol{\omega}$ to be [4]

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} \tag{3-20}$$

Combining all these equalities and by assuming that $k_{\parallel} = k_{\perp}$ [4], Equations 3-5, 3-6 and 3-7 can be rewritten as

$$m\ddot{\boldsymbol{p}} = m\boldsymbol{g} + \text{Im}(\boldsymbol{q} \otimes \begin{bmatrix} 0 \\ f\boldsymbol{e}_{z_{\mathcal{B}}} \end{bmatrix} \otimes \boldsymbol{q}^*) + K_{\text{aero}}\theta_{\Sigma}\dot{\boldsymbol{p}} \tag{3-21}$$

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} \tag{3-22}$$

$$I\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times I\boldsymbol{\omega} + \tau \tag{3-23}$$

3-1-3 Discretization

In order to use the dynamic model on an on-board controller, a discrete-time system has to be derived. In this case it is assumed that the controller input follows a zero-order hold, i.e. $\boldsymbol{u}_t = \boldsymbol{u}_{t_k}\ \forall t \in [t_k, t_k + \Delta t]$. Now by integrating over the sample-time, we can obtain the following discrete-time state-space equation

$$x(k+1) = \Phi x(k) + \Gamma u(k) \tag{3-24}$$

$$y(k) = Cx(k) \tag{3-25}$$

where the matrices Φ and Γ can be defined as [39]

$$\Phi = e^{Ah},\ \Gamma = \int_0^h e^{\tau A} d\tau B \tag{3-26}$$

With A and B the state-space matrices as defined in Equation 3-1.

When using this discrete-time state-space in a real-time application, calculation of the matrix exponential $e^{\tau A}$ is computationally costly. In this case, simplifications might be necessary. The matrix exponential is defined by [40]

$$e^{\tau A} = I + \frac{\tau}{1!}A + \frac{\tau^2}{2!}A^2 + \dots \tag{3-27}$$

From this definition, it can be seen that for a small enough sampling time τ, the matrix exponential can be approximated by

$$e^{\tau A} = I + \tau A + \mathcal{O}(\tau^2) \tag{3-28}$$

3-2 Sensor Models

To estimate the dynamics described in the sections above, measurements are necessary to gather data on the states. The sensors that will be used in the thesis are the angular rate gyroscope, the accelerometer and the UWB distance measurements. For each of these sensors, a model will be described below that can be used to estimate states in the dynamical models given above.

3-2-1 Angular rate gyroscope

A gyroscope measures the angular velocity of the quadrotor body-fixed frame $\mathcal{B}$ relative to $\mathcal{A}$. Multiple models with varying complexity exist and will be shown. A simple model is by assuming that there are no scale errors or gyro biases in the system [14]. This measurement is modeled as

$$\boldsymbol{\omega}_{\text{gyro}} = \boldsymbol{\omega} + \boldsymbol{\eta}_{\text{gyro}} \tag{3-29}$$

where $\boldsymbol{\eta}_{\text{gyro}}$ is the gyro measurement noise and is assumed to be zero-mean white noise.

When the sensor is well calibrated and does not suffer from severe time-varying gyro bias, this model suits most applications. Another model is suggested in [34], where this time-varying gyro bias is also added

$$\boldsymbol{\omega}_{\text{gyro}} = \boldsymbol{\omega} + \boldsymbol{\eta}_{\text{gyro}} + \boldsymbol{b}_{\text{gyro}} \tag{3-30}$$

The gyro bias $\boldsymbol{b}_{\text{gyro}}$ can be a constant or slowly varying value. If this bias is not constant, it can be described by [37]

$$\dot{\boldsymbol{b}}_{\text{gyro}} = \boldsymbol{\eta}_{\text{gdrift}} \tag{3-31}$$

where $\boldsymbol{\eta}_{\text{gdrift}}$ is the gyro bias drift noise and is also assumed to be a zero-mean white-noise process.

In practice it might sometimes not be needed to estimate the noise since in some high-end gyroscopes the measurement noise and bias is negligible and therefore the measurement can be directly used as an estimate of the quadrotors angular velocity

$$\hat{\boldsymbol{\omega}} = \boldsymbol{\omega}_{\text{gyro}} \tag{3-32}$$

3-2-2 Accelerometer

A gyroscope measures the instantaneous linear acceleration of the quadrotor body-fixed frame $\mathcal{B}$. A common model for an accelerometer measurement is given as [34, 14]

$$\ddot{\boldsymbol{p}}_{\text{acc}} = R^T(\ddot{\boldsymbol{p}} - \boldsymbol{g}) + \boldsymbol{\eta}_{\text{acc}} + \boldsymbol{b}_{\text{acc}} \tag{3-33}$$

where $\boldsymbol{\eta}_{\text{acc}}$ is the accelerometer measurement noise, assumed to be a zero-mean white-noise process, and $\boldsymbol{b}_{\text{acc}}$ is the accelerometer bias. The largest part of disturbances on the accelerometer measurements are due to vibrations of the sensor. To counter these, significant low-pass filter are to be applied either mechanically or electrically. The bias in accelerometers, in contrast to gyroscopes, is mostly a constant, and can therefore be estimated by averaging the output of the sensor while the quadrotor is not moving yet. This bias is assumed constant and is then subtracted from the output in-flight, thus removing the bias.

In [14], the acceleration $\ddot{\boldsymbol{p}}$ is substituted by equation 3-5 to obtain the following equation

$$\ddot{\boldsymbol{p}}_{\text{acc}} = \frac{1}{m}(f\boldsymbol{e}_{\boldsymbol{z_B}} + \boldsymbol{f}_{\text{a}}) + \boldsymbol{\eta}_{\text{acc}} + \boldsymbol{b}_{\text{acc}} \tag{3-34}$$

By using this notation, the measurement $\ddot{\boldsymbol{x}}_{\text{acc}}$ can be used to obtain information on the aerodynamic force $\boldsymbol{f}_{\text{a}}$ if the rotor-speed is known. This can be exploited in the design of an estimator.

3-2-3 Ultra Wideband (UWB) range measurements

By using methods described in Chapter 2, a measurement of the distance $d_{\text{uwb, i}}$ between beacon i and the quadrotor can be obtained. These measurements can be described as

$$d_{\text{uwb, i}} = \|\boldsymbol{p}_{\text{uwb,i}} - \boldsymbol{x}\| + \eta_{\text{uwb}} + b_{\text{uwb}} \tag{3-35}$$

where η_{uwb} is the UWB distance measurement noise and b_{uwb} is the UWB measurement bias (see Section 2-2-2 for an in-depth analysis of this bias). By using 4 or more of these measurements, an estimate of the quadrotors location $\boldsymbol{x}$ can be made.

3-3 Simulation Environment Design

To make a reliable comparison between state estimators, it is important to have a simulation environment in which data can be quickly generated based on different scenarios. With the dynamic equations of the quadrotor and the sensor descriptions in Section 3-1, such an environment has been created in Simulink based on a model designed in [4], where all twelve states (position, velocity, attitude and angular velocity) are simulated based on a reference.

The outer loop in Simulink is defined as can be seen in Figure 3-4.

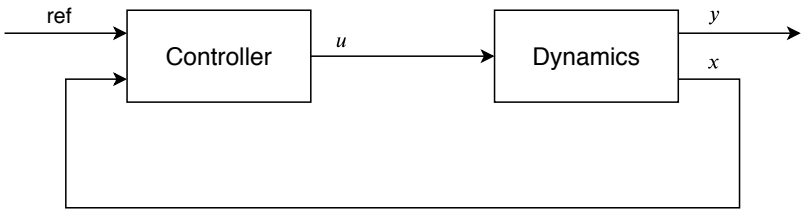


Figure 3-4: Outer simulation loop

In this Figure, ref is the reference signal that is sent to the quadrotor, this consists of a reference for each of the twelve states. ω are the rotor speeds of the quadrotor, x are the real states of the quadrotor and y are the measurements that are available for an estimator.

3-3-1 Controller

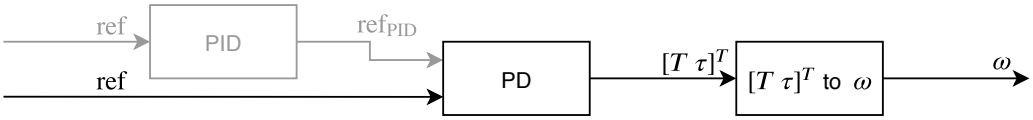


Figure 3-5: Simulation control scheme

The Controller block in Figure 3-4 is the controller, which calculates the desired rotor speeds ω for the given reference. A basic scheme of how this is done is shown in Figure 3-5. As can be seen in Figure 3-4, this simulator assumes that the controller has access to the real states of the system.

Two types of control can be used here, of which the first is by simulating manual control. In this case, only four reference signals are considered: thrust, yaw, pitch and roll. This is equal to when an operator would be controlling the quadrotor with manually. These references are fed to a simple PD-type controller that calculates the necessary thrust T and torques τ to obtain these references from the current state. Using a mapping from torque to rotor speeds (see Section 3-4-1) and bounding the signal by the lower and upper limits, this is translated to rotor speeds ω.

The second type is by simulating position reference tracking, in which the quadrotor only receives a position as a reference. This position is used in a PID-type controller to calculate the desired pitch and roll angles. These desired attitudes are then bounded to ensure stability of the quadrotor. Using the desired pitch and roll angles in the same PD-type controller as for the manual case, the rotor speeds can be calculated in the same way.

3-3-2 Process

In the simulation of the quadrotor, first the input rotor speed ω is mapped to thrust and torques by the mappings described in Section 3-4-1. Together with the previously calculated states and the model parameters (as discussed in Section 3-4), these inputs are used in the continuous-time rigid-body model based on quaternions, described by Equations (3-21, 3-22, 3-23), to calculate the next time-step state.

These states are used to simulate the measurements, as they are produced on a normal quadrotor. These were all based on the sensor models as described in Section 3-2 these measurements are gathered in a single vector that can be used by an estimator to estimate the quadrotors state, as it would in a real experiment. The following measurements are implemented:

Gyroscope

The angular rate gyroscope is a biased measurement of the angular velocity, but this bias is normally estimated during calibration at the start-up of the quadrotor, after which the gyroscope measurements is as described in Equation 3-29. Since this angular velocity is included in the states of the quadrotor, the zero-mean white Gaussian noise can simply be added to this state to obtain the gyroscope measurement. The size of this gyroscope noise $\boldsymbol{\eta}_{\text{gyro}}$ is taken to be $[0.0083\Delta t, 0.0083\Delta t, 0.0543\Delta t]$, as was used in [41].

Accelerometer

By including the inertial-frame acceleration and rotation matrix from the dynamic equations, it is straightforward to calculate the accelerometer output with Equation 3-33 by adding the gravitation to the z-component and rotating the result. The zero-mean white Gaussian noise is then added to obtain the accelerometer measurements as they would be on the Crazyflie. The size of this accelerometer noise $\boldsymbol{\eta}_{\text{acc}}$ is taken to be $[4.873\Delta t, 4.873\Delta t, 0.453\Delta t]$, as was used in [41].

UWB range measurement

The anchor-positions that will be used in the real experiments can be used here to generate the Ultra Wideband (UWB)-measurements. A measurement is calculated by using Equation 3-35. The bias that is used can be for instance a random time-varying bias or, supported by the results of Section 2-2-2, a bias that depends on the (x, y)-location of the quadrotor. Since UWB-measurements may suffer from serious outliers in the range of meters, outliers can be added to the range measurements. These outliers are created by using a by using a random number generator with a variance of $\sigma^2_{\text{outliers}} = 1$ and adding these values to the range measurements if the absolute value is greater than a specific threshold. This threshold can be chosen such that the probability of an outlier is a certain percentage by choosing the appropriate values from the standard normal distribution. The size of the UWB noise η_{uwb} is taken to be 0.025, which was estimated by gathering range measurements at a constant distant and analyzing the statistical properties and compared to the results of [14]. The sample-rate of the UWB system is a lot lower than the accelerometer and gyroscope, so this is set to 25Hz, which was estimated from real data.

Attitude and Heading Reference System (AHRS) measurement

Both complementary filters that are described in 3-4-3 can be used as an input to the system, and it is necessary to simulate these measurements as well. The AHRS-measurements produce biased attitude estimations because of the accelerometer bias. This bias (ϕ_b, θ_b) can be modeled as a function of the yaw angle ψ of the quadrotor [5]

$$\begin{bmatrix} \phi_b \\ \theta_b \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \boldsymbol{b}_{\text{acc}} \tag{3-36}$$

Where $\boldsymbol{b}_{\text{acc}}$ is the accelerometer bias in degrees in the body-fixed frame. From literature, this bias in real-world experience is less than 3° [5]. In the simulation environment this is set to $\boldsymbol{b}_{\text{acc}} = [2^\circ \ \ -1^\circ]^T$. The AHRS-measurement can now be described as:

$$\begin{bmatrix} \tilde{\phi}_k \\ \tilde{\theta}_k \end{bmatrix} = \begin{bmatrix} \phi_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \boldsymbol{b}_{\text{acc}} + \begin{bmatrix} \epsilon_\phi \\ \epsilon_\theta \end{bmatrix} \tag{3-37}$$

Where $[\epsilon_\phi \ \epsilon_\theta]^T$ is the detection noise.

3-3-3 Conclusion

The simulation environment depicted above can be used to quickly generate flight data. Since in this simulation, the size of all errors can be easily adjusted, all the statistical properties are exactly known. This supports a comparison between different estimators.

Also, instead of feeding x in Figure 3-4 to the input of the controller, the state estimate of a filter might be used here to simulate a real experiment even more. This might be especially useful in comparing the reference tracking performance of different estimators.

3-4 Quadrotor Used in This Research

This section provides the user with information on the quadrotor that is used in our research. Section 3-4-1 will discuss the specifications of the Crazyflie 2.0 UAV. This includes the technical specifications, mappings from input to output and model parameters. Section 3-4-2 discusses the Extended Kalman Filter (EKF) that is implemented on the Crazyflie 2.0 and will be used as a benchmark estimator in our research. Section 3-4-3 investigates both complementary filters that are available on the Crazyflie 2.0.

Figure 3-6: Crazyflie 2.0 by Bitcraze

3-4-1 Crazyflie Specifications

The Crazyflie 2.0 is an open-source project designed by Bitcraze with the aim to provide an affordable flying development platform. Because of its small size and low weight, it is accessible in indoors robotics research for a great variety of research groups. This chapter will present the specifications of the UAV but also mappings from input to output and model parameters that are necessary for implementation. Subsection 3-4-1 discusses all the technical specifications of the Crazyflie 2.0 and discusses the implications that arise by using these. Section 3-4-1 discusses the mappings from the input PWM to the rotor speed and thrust, so that they can be used in the dynamic models described in Chapter 3. Section 3-4-1 discusses the inertia matrix and the drag coefficients of the Crazyflie 2.0. For sake of brevity, the version number 2.0 will be omitted in the continuation of this text.

Technical specifications

The Crazyflie is the second quadrotor that is designed by Bitcraze. It is equipped with four core-less DC motors with the following specifications:

- diameter: 7mm
- length: 16mm
- weight: 2.7g
- Kv: 14000rpm
- Rated voltage: 4.2V
- Rated current: 1000mA

The propellers that come with the kit have a diameter of 45mm.

The Crazyflie has a diagonal distance between motors of 92mm and the total weight of the quadrotor ads up to 27g. The battery is a 1 Cell (3.7V) 240 mAh LiPo battery that weighs 7.1g. As stated by Bitcraze, this should allow for 7 minutes of continuous flight [42].

Two micro-processors are used by the quadrotor; one for power energy and radio communication and one for the control-loop. The micro-controller that is used for all on-board calculations is the ARM 32-bit STM32F405 Cortex-M4 (with floating point unit) that runs at 168MHz with 192kb of SRAM.

The Crazyflie can be controlled either over Bluetooth Low Energy (BLE) or over the 2.4GHz Crazyradio PA.

The on-board IMU is the MPU-9250, a 9-DOF IMU with a gyroscope, accelerometer, magnetometer and a LPS25H high precision pressure sensor (barometer).

Mappings from input to output

The models that were shown in the previous chapter are based on the assumption that the rotor speed is known. Since the Crazyflie uses core-less motors, this is not a valid assumption. In literature, mappings from input to rotor speed and from input to thrust have been created in order to overcome this problem. A linear relation between PWM and rotor speed was found in [2] and a quadratic relation between PWM and thrust is shown by Bitcraze [3]. First, the mapping to rotor speed will be discussed and after that the mapping to thrust.

Mapping from Pulse Width Modulation (PWM) to rotor speed To estimate this mapping, a experiment was set up in [2]. By placing reflective tape on the bottom of the propellers and combining this with a tachometer, the Rounds Per Minute (RPM) can be measured. The motor drivers that the Crazyflie uses, allow a 16-bit integer as input, which means that a signal of 0 results in no motor activity and a value of 65535 results in full throttle. In the experiment, increasing values in this interval were sent to the Crazyflie and a linear fit was applied to the resulting values. From these tests, the following linear transfer function could be found

$$\theta_i = 0.04076521 \text{ PWM}_i + 380.8359 \tag{3-38}$$

with the value PWM_i being the input value sent to motor i. A note to this transfer function is that the real motor response in the interval $[0, 1000]$ is not linear and should therefor not be used here. In [2] it is advised to assume that the rotor speed is zero in this interval.

Mapping from Pulse Width Modulation (PWM) to thrust This is another interesting mapping, since parameter c_T in Equation 3-11 has to be determined and a direct mapping from PWM to thrust might result in a more accurate estimation. Since a linear model is assumed for $\theta_i(\text{PWM}_i)$, a quadratic model in the form of $y = ax^2 + bx + c$ has to be fitted through the data. In literature [2, 3, 4] this is done by placing the Crazyflie on a scale and calculating the thrust that is generated at a certain PWM by looking at the decrease in weight.

In [3], an 8-bit PWM signal was send to all four motors and the resulting data was used to fit a second order model for the thrust. This model is as follows

$$f_\text{total} \, [g] = 0.409 \cdot 10^{-3} \text{ PWM}_i^2 + 140.5 \cdot 10^{-3} \text{ PWM}_i - 0.099 \tag{3-39}$$

where f_total is defined in grams. In [2], a 16-bit PWM signal was send, instead of an 8-bit signal and only two motors were considered at a time. The resulting thrust was divided by 2 to get the thrust per motor. Again, a second order model was fitted and the resulting equation is as follows

$$f_i \, [\text{N}] = 2.130295 \cdot 10^{-11} \text{ PWM}_i^2 + 1.032633 \cdot 10^{-6} \text{ PWM}_i + 5.48456 \cdot 10^{-4} \tag{3-40}$$

where f_i is now defined in Newtons. In [4], the last term in the model was defined to be 0. If the value $\text{PWM}_i = 0$, the total thrust should also be zero. The fitted model in this reference was

$$f_{\text{total}} = 0.26 d_{\text{PWM}}^2 + 0.35 d_{\text{PWM}} \tag{3-41}$$

where f is defined in Newtons and d_{PWM} [0,1] is the PWM duty cycle.

To make a comparison between these fitted models, the all models were rewritten in the same units as the second model. All resulting models are shown in Figure 3-7.

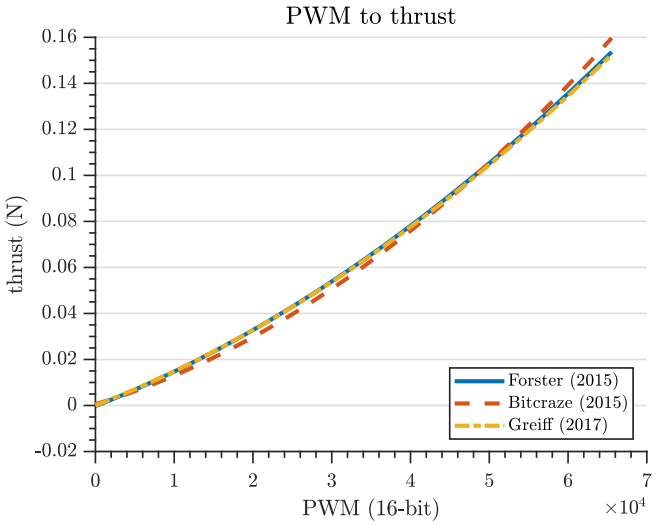


Figure 3-7: Pulse Width Modulation (PWM) to thrust mapping for a single motor [2, 3, 4]

It can be seen that although all models resemble each other, there are some slight deviations for higher PWM values and from 15000-35000 as well with the model from [3]. This shows that an investigation has to be done on the necessary accuracy of this mapping for our own applications and maybe perform experiments ourselves as well. Otherwise, one of these models could be used.

Model parameters

In this section, necessary model parameters are presented. In Subsection 3-4-1 the inertia matrix and in Subsection 3-4-1 the drag coefficients are shown as found in literature.

Inertia matrix The non-linear model of the moving quadrotor in Equation 3-5 depends on the inertia matrix I. The inertia matrix can be estimated by performing experiments that involve rotations around certain axes. In [2] such experiments were executed and the resulting inertia matrix is shown below

$$I = \begin{bmatrix} 16.571710 & 0.830806 & 0.718277 \\ 0.830806 & 16.655602 & 1.800197 \\ 0.718277 & 1.800197 & 29.261652 \end{bmatrix} \tag{3-42}$$

Drag coefficients The aerodynamic forces that act on the quadrotor were assumed to be linear in the velocity in Equation 3-13. The drag coefficients K_{aero} have to be estimated from experiments as well. In [14], the values were estimated by analyzing data from quadrotor flights for a slightly larger quadrotor than the Crazyflie

$$k_{\perp} = -0.00011 \text{ Ns}^2\text{rad}^{-1}\text{m} \tag{3-43}$$
$$k_{\parallel} = -0.00023 \text{ Ns}^2\text{rad}^{-1}\text{m} \tag{3-44}$$

In [2], a more complex model was proposed with a full K_{aero}-matrix for the Crazyflie. The values that were found by performing tests in a wind tunnel were as follows

$$K_{\text{aero}} = \begin{bmatrix} -10.2506 & -0.3177 & -0.4332 \\ -0.3177 & -10.2506 & -0.4332 \\ -7.7050 & -7.7050 & -7.5530 \end{bmatrix} \cdot 10^{-7} \text{ kg} \cdot \text{rad}^{-1} \tag{3-45}$$

It has to be noted that the value -7.7050 is questionable, because of symmetry in the quadrotor. This value would mean that a translation in negative x-direction would result in a positive drag in z-direction, while a translation in positive x-direction would result in the the same drag but with the opposite sign [2]. Another drag model would be necessary to include this properly.

3-4-2 Crazyflie Extended Kalman Filter (EKF)

When the Crazyflie recognizes the Flowdeck or the Loco Positioning Deck, the Extended Kalman Filter (EKF) will be used for state estimation. This filter is based on work by Mueller et al. [14]. In this Chapter, the derivation of the filter and the design choices will be described.

Kalman Filter

The Kalman Filter is an algorithm that is used for sensor fusion and (optimal) state estimation based on knowledge of the statistical noise that is present in both the measurements and the model. It is an iterative method, that consists of two steps; the prediction step and the correction step. Consider the following system

$$x(k+1) = A(k)x(k) + B(k)u(k) + w(k) \tag{3-46}$$
$$y(k) = C(k)x(k) + v(k) \tag{3-47}$$

With Gaussian noise $w(k)$ and $v(k)$ with the following covariance properties

$$\mathbb{E}\begin{bmatrix} v(k) \\ w(k) \end{bmatrix} = 0 \quad \mathbb{E}\left[\begin{bmatrix} v(k) \\ w(k) \end{bmatrix}\begin{bmatrix} v(k) \\ w(k) \end{bmatrix}^T\right] = \begin{bmatrix} R(k) & 0 \\ 0 & Q(k) \end{bmatrix} \tag{3-48}$$

With R(k) and Q(k) positive definite. This assumes that the measurement and process noise is uncorrelated, which might not be the case of a quadrotor but is a good first approximation.

The first step is predicting the next output $\hat{x}(k|k-1)$ and state-error covariance $P(k+1|k)$, based on the model and the state-error covariance matrix $P(k-1|k-1)$.

$$P(k|k-1) = A(k)P(k-1|k-1)A^T(k) + Q(k) \tag{3-49}$$
$$\hat{x}(k|k-1) = A(k)\hat{x}(k-1|k-1) + B(k)u(k) \tag{3-50}$$

In the correction-step that follows, measurements are added to improve the estimation. Now the Kalman-gain ($K(k)$), the updated state-estimation and the updated covariance matrix are calculated as follows

$$K(k) = P(k|k-1)C^T(k)\Big(R(k) + C(k)P(k|k-1)C^T(k)\Big)^{-1} \tag{3-51}$$
$$\hat{x}(k|k) = \hat{x}(k|k-1) + K(k)\Big(y(k) - C(k)\hat{x}(k|k-1)\Big) \tag{3-52}$$
$$P(k|k) = \Big(I - K(k)C(k)\Big)P(k|k-1)\Big(I - K(k)C(k)\Big)^T + K(k)R(k)K^T(k) \tag{3-53}$$

For proof that the solution is a minimum variance unbiased estimate, the reader is referred to [43, p. 136-139].

Extended Kalman Filter (EKF)

The normal Kalman Filter can only be used in the case of a linear system. Since the dynamic model of a quadrotor is highly nonlinear, an extension to this filter is used; the Extended Kalman Filter (EKF). By taking the Jacobian of the non-linear state- and measurement-equations, a model similar to the normal Kalman filter can be obtained and is solved in exactly the same way. This method is commonly used in quadrotor research and its use can be found in [14, 15, 27, 4].

The nonlinear system, in comparison to equation (3-46), can now be defined as:

$$x(k+1) = f(x, u, w) \tag{3-54}$$
$$y(k) = h(x, v) \tag{3-55}$$

In this model, every update the matrices $A(k)$, $B(k)$, $C(k)$, $Q(k)$ and $R(k)$ are replaced by Jacobians:

$$A(k) = \left.\frac{\partial f(x,u,w)}{\partial x}\right|_{\hat{x}(k-1|k-1),u(k)} \qquad C(k) = \left.\frac{\partial h(x,v)}{\partial x}\right|_{\hat{x}(k|k-1)} \tag{3-56}$$
$$Q_{new}(k) = W(k)Q(k)W(k)^T \quad R_{new}(k) = V(k)R(k)V(k)^T \tag{3-57}$$

Where $W(k)$ and $V(k)$ are the following Jacobians

$$W(k) = \left.\frac{\partial f(x,u,w)}{\partial w}\right|_{\hat{x}(k-1|k-1),u(k)} \qquad V(k) = \left.\frac{\partial h(x,v)}{\partial v}\right|_{\hat{x}(k|k-1)} \tag{3-58}$$

The full derivation of the EKF and a stability analysis can be found in [44].

In the EKF, the state distribution is approximated by Gaussian variables and than propagated through a first order approximation (by using the Jacobians) of the nonlinear system. Although the computational complexity is relatively low and accuracy high (especially in the case with functions that have a well defined first-order derivative), the filter might diverge from the real values or may result in poor performance when the model does not resemble the real system adequately or the chosen initial values differ too much from the real initial values.

Since in the case of quadrotors, the model that is used is a simplified version of reality, convergence of the EKF is hard to prove. If the initial values are too far from the real values, the estimation might never converge to the real values.

Implementation on the Crazyflie

In order to reduce the computational complexity of the filter, the filter would only consist of the nine-dimensional stochastic state $\boldsymbol{x}$:

$$\boldsymbol{x} = (\boldsymbol{p}, \boldsymbol{v}, \boldsymbol{\delta}) \tag{3-59}$$

Where $\boldsymbol{p}$ is the position of the Crazyflie in the inertial-fixed frame $\mathcal{A}$ as described in Section 3-1, $\boldsymbol{v} = R^{-1}\dot{\boldsymbol{p}}$ is the velocity in the body-fixed frame $\mathcal{B}$ and $\boldsymbol{\delta}$ is an attitude error representation, that is used to update the reference rotation R_{ref}. This will be further explained at the end of this Subsection.

The prediction equations of the filter are derived from the equations of motions as defined in Section 3-1 as follows:

$$\dot{\hat{\boldsymbol{p}}} = \hat{R}\hat{\boldsymbol{v}} \tag{3-60}$$

$$\dot{\hat{\boldsymbol{v}}} = \frac{1}{m}(f\boldsymbol{e}_{z_{\mathcal{B}}} + \boldsymbol{f}_{\text{a}}) - [\hat{\boldsymbol{\omega}}]_{\times}\boldsymbol{v} - R^{-1}g\boldsymbol{e}_{z_{\mathcal{A}}} \tag{3-61}$$

$$\dot{\hat{\boldsymbol{\delta}}} = \hat{\boldsymbol{\omega}} \tag{3-62}$$

One of the disadvantages of the Tait-Bryan angles that was discussed in Section 3-1 were singularities for values of (ψ, ϕ, θ) where axis would align. In [14] this problem is solved by using a Multiplicative Extended Kalman Filter (MEKF). In this filter, instead of using the states (ψ, ϕ, θ), a reference orientation R_{ref} is introduced combined with a vector $\boldsymbol{\delta}$ to represent the attitude error. Now the estimated rotation matrix from frame $\mathcal{A}$ to $\mathcal{B}$ is set as

$$\hat{R} = \hat{R}_{\text{ref}}(I + [\hat{\boldsymbol{\delta}}]_{\times}) \tag{3-63}$$

This state can be updated by using the gyroscope data as follows

$$\dot{\hat{\boldsymbol{\delta}}} = \hat{\boldsymbol{\omega}} + \frac{1}{2}[\hat{\boldsymbol{\delta}}]_\times \hat{\boldsymbol{\omega}} \tag{3-64}$$

By using the accelerometer measurements and gyroscope measurements, the discrete-time prediction can be obtained as follows:

$$\hat{\boldsymbol{p}}_k = \hat{\boldsymbol{p}}_{k-1} + R\Big(\boldsymbol{v}_{k-1}\Delta t + \frac{a_k^{(\text{acc})}\Delta t^2}{2}\Big) - \frac{g\Delta t^2}{2}\boldsymbol{e}_{z_\mathcal{A}} \tag{3-65}$$

$$\hat{\boldsymbol{v}}_k = \hat{\boldsymbol{v}}_{k-1} - [\omega_k^{(gyro)}]_\times \hat{\boldsymbol{v}}_{k-1}\Delta t - gR^{-1}\boldsymbol{e}_{z_\mathcal{A}}\Delta t + a_k^{(\text{acc})} \tag{3-66}$$

Matrix A and C in the covariance prediction and UWB measurement update steps are defined as follows:

$$A = \begin{bmatrix} I & R_k\Delta t & -R_k[\boldsymbol{v}_k]_\times \Delta t \\ 0 & I - [\omega_k]_\times \Delta t & -g[R_k^{-1}\boldsymbol{e}_{z_\mathcal{A}}]_\times \Delta t \\ 0 & 0 & I + [-\frac{\boldsymbol{\omega}_k\Delta t}{2}]_\times + [-\frac{\boldsymbol{\omega}_k\Delta t}{2}]_\times^2/2 \end{bmatrix} \tag{3-67}$$

$$C_{\text{UWB}} = \begin{bmatrix} (\hat{\boldsymbol{p}}_k - \boldsymbol{p}_{\text{UWB,j}})/\|(\hat{\boldsymbol{p}}_k - \boldsymbol{p}_{\text{UWB, j}})\| & 0^{1\times 6} \end{bmatrix} \tag{3-68}$$

Where $\boldsymbol{p}_{\text{UWB, j}}$ is the position of UWB-beacon j in the inertial-fixed frame $\mathcal{A}$.

After each Kalman filter iteration, K_{ref} is updated with the attitude error and the attitude error is reset to zero. This way, singularities will not be reached. For a more comprehensive description of this method and the derivation of the bottom-right equation, the reader is referred to [45].

3-4-3 Crazyflie Complementary Filter

If no position measurements are available, the Crazyflie defaults to a complementary filter. These filters combine the gyroscope and accelerometer measurements to estimate the attitude of the quadrotor. On the Crazyflie, two of these filters are implemented and they will both be described in the following Sections. Both filters use quaternions to represent the attitude. The quaternion form is commonly used in robotics for operations on $SO(3)$ because it is not susceptible to singularities, enables smooth interpolation of rotations and is computationally efficient. An introduction to quaternions can be found in Subsection 3-1-2.

Mahony's Complementary Filter

The default filter that is used, is a complementary filter designed by R.E. Mahony [46] in its quaternion form. The dynamic equations of the attitude in quaternion representation can be written as:

$$\dot{q} = \frac{1}{2} q \otimes \boldsymbol{p}\{\Omega\} \tag{3-69}$$

Where Ω is the angular velocity in the body-fixed frame $\{\mathcal{B}\}$ and $\boldsymbol{p}\{\boldsymbol{x}\} = [0\ \boldsymbol{x}]$ is the pure quaternion, that can be related to the quaternion velocity.

The gyroscope can be used to estimate the angular velocity distorted by a bias $\Omega_{\text{gyro}} = \Omega + b$. This bias is estimated by an integrator using the accelerometer data. The observer that is proposed by Mahony is as follows:

$$\dot{\hat{q}} = \frac{1}{2}\hat{q} \otimes \boldsymbol{p}\{\Omega^{(\text{gyro})} - \hat{b} + k_P e\} \tag{3-70}$$

$$\dot{\hat{b}} = -k_I e \tag{3-71}$$

Where k_P and k_I are the proportional and integral gain respectively and the prediction error is calculated from the accelerometer measurements as follows:

$$e = \mathrm{v} \otimes \boldsymbol{p}\{\mathrm{v}^{(\text{acc})}\} \tag{3-72}$$

$$\mathrm{v} = \hat{q}^* \boldsymbol{p}\{\begin{bmatrix}0\\0\\1\end{bmatrix}\}\hat{q} \tag{3-73}$$

$$\mathrm{v}^{(\text{acc})} = \frac{a^{(\text{acc})}}{\|a^{(\text{acc})}\|} \tag{3-74}$$

Where v is the predicted gravitational vector in the body-fixed frame $\mathcal{B}$ as a pure quaternion and v_{acc} is the measured gravitational vector that is obtained by normalizing the accelerometer measurement a_{acc}. These equations are discretized and implemented on the quadrotor as follows:

$$\hat{q}_k = \hat{q}_{k-1} + (\frac{1}{2}\hat{q}_{k-1} \otimes \boldsymbol{p}\{\Omega^{(\text{gyro})} - \hat{b}_k + k_P e\})\Delta t \tag{3-75}$$

$$\hat{b}_k = \hat{b}_{k-1} - k_I e \Delta t \tag{3-76}$$

Since this complementary filter is locally exponentially stable, the estimate will not diverge in contrast to dead reckoning methods that integrate the gyroscope measurements. A stability proof for this filter can be found in [46].

Madgwick's Complementary Filter

The second complementary filter that is implemented on the Crazyflie is designed by S.O.H. Madgwick [47]. In this filter, the gyroscope and accelerometer data are both used to create an estimate of the attitude quaternion and these are fused in the following way:

$$\hat{q}_k = \gamma \hat{q}_k^{(\text{acc})} + (1-\gamma)\hat{q}_k^{(\text{gyro})} \quad \gamma \in [0,1] \tag{3-77}$$

The attitude estimate based on the gyroscope data is using the same definition as Equation 3-69 where Ω is substituted by the gyroscope measurement and integrating:

$$\hat{q}_k^{(\text{gyro})} = \hat{q}_{k-1} + (\frac{1}{2}\hat{q}_{k-1} \otimes \boldsymbol{p}\{\Omega^{(\text{gyro})}\})\Delta t \tag{3-78}$$

The dead reckoning method is then compensated by an optimization based approach to estimate the attitude quaternion using the accelerometer data. This attitude can be described as the rotation of the gravitational vector to the body-fixed frame $\mathcal{B}$:

$$\boldsymbol{p}\{\mathrm{v}^{(\text{acc})}\} = \hat{q}^{*(\text{acc})} \otimes \boldsymbol{p}\{\boldsymbol{g}\} \otimes \hat{q}^{(\text{acc})} \tag{3-79}$$

The quaternion $\hat{q}^{(\text{acc})}$ can be found by solving the optimization problem

$$\min_q \frac{1}{2} f(q)^T f(q) \tag{3-80}$$

Where

$$f(q) = q \otimes \boldsymbol{p}\{\boldsymbol{g}\} \otimes q - \boldsymbol{p}\{\mathrm{v}^{(\text{acc})}\} \tag{3-81}$$

The solution to this optimization problem is found by taking one gradient descent step:

$$\hat{q}_k^{(\text{acc})} = \hat{q}_{k-1}^{(\text{acc})} - \mu \frac{\frac{1}{2}\text{Jac}(f(q))^T f(q)}{\|\frac{1}{2}\text{Jac}(f(q))^T f(q)\|} \tag{3-82}$$

Now Equation 3-77 can be simplified by choosing μ really large and using the gyroscope divergence rate β to choose the fusion coefficient γ as follows:

$$\gamma = \frac{\beta \Delta t}{\mu} \tag{3-83}$$

This results in the following update equation for the attitude estimate:

$$\hat{q}_k = \hat{q}_{k-1} + \Big(\frac{1}{2}\hat{q}_{k-1} \otimes \boldsymbol{p}\{\Omega^{(\text{gyro})}\} - \beta \frac{\frac{1}{2}\text{Jac}(f(q))^T f(q)}{\|\frac{1}{2}\text{Jac}(f(q))^T f(q)\|}\Big)\Delta t \tag{3-84}$$

Experimental results

To choose one of the filters described above, both filters are tested in our simulation environment. The Crazyflie's position is commanded to be a set of random numbers, uniformly distributed between -5m and 5m with a sample time of 3 seconds. The yaw of the Crazyflie is also commanded to be a random number between -0.8rad and 0.8rad. 6 data-sets of 21 seconds are created using these random position and attitude commands. Both complementary filters are tested on these data-sets and the average RMSE for these filters can be seen in Table 3-1. For both filters, the parameters as implemented on the Crazyflie are used. For Mahony's complementary filter, $k_P = 0.4$ and $k_I = 0.001$. For Madgwick's filter, $\beta = 0.01$.

	Mahony	Madgwick
ϵ_{RMS} yaw (deg)	1.40°	0.54°
ϵ_{RMS} pitch (deg)	2.86°	0.69°
ϵ_{RMS} roll (deg)	3.47°	0.80°

Table 3-1: RMSE for both complementary filters on simulation data

From this data, it follows that Madgwick's filter performs better under these circumstances. An extensive comparison of both methods on the real system would be preferable but since the attitude measured by the OptiTrack system is in the same order of magnitude as the error that was calculated for Madgwick's filter, and due to lack of time it was decided to leave this for further research. Since the amount of computations necessary for both filters is comparable, Madgwick's filter will be used in our design.

3-5 Concluding Remarks

To be able to design an estimation method, complete knowledge of the system is necessary. This chapter investigates the dynamic properties of a quadrotor and the on-board sensors. This information is used in a simulation environment, developed in Simulink, that can be used to quickly create data-sets including the dynamics and the sensory output. The Crazyflie is an interesting development platform for research on aerial robotics and will be used in the research described in this thesis. A reasonable amount of research has been performed on this quadrotor in literature and this data can be easily implemented in our own system. The main focus point for an estimation technique will be the on-board micro processor because this might become a bottleneck. The estimators that are implemented on this system are discussed so that they can be used as a benchmark and as input for our own estimator.

Chapter 4

Moving Horizon Model Parametrization (MH-MP)

In this chapter, the designed state estimator will be derived and explained. This estimator is the highlighted block in Figure 4-1. Many state estimation methods for quadrotors exist, such as the Extended Kalman Filter (EKF), the Unscented Kalman Filter (UKF), Particle Filters and methods that rely on a Moving Horizon. The benefit of the latter is that a series of measurements is used, instead of just the last measurement. Since we are dealing with relatively low-frequency data with a low accuracy, using a time-window of measurements can improve the accuracy, deal with outliers and smooth the estimates which is especially important for control. The design of this filter is an adaptation of the method used by Li et al. [5].

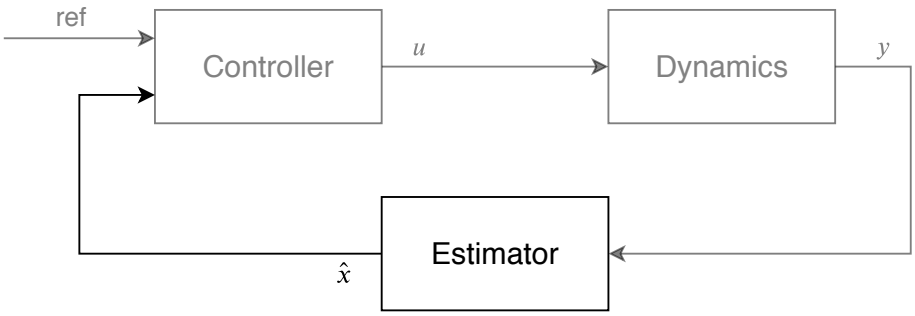


Figure 4-1: Quadrotor control loop block diagram

4-1 State Estimator Background & Motivation for Using a Moving Horizon

The goal of a state estimator is to obtain an estimate of the internal state $\boldsymbol{x}$ that is as accurate as possible, from the system dynamics $f(\boldsymbol{x})$ and the available measurements $\boldsymbol{y}$. This can be

described by the following optimization problem:

$$\min_{\hat{\boldsymbol{x}}_k, k=0,\ldots,N} \sum_{k=0}^{N} (\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k)^2 \tag{4-1}$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}}_k = f(\boldsymbol{x}_k, \boldsymbol{u}_k) \quad \forall k = 0, \ldots, N \tag{4-2}$$

$$\boldsymbol{y}_k = g(\boldsymbol{x}_k) \quad \forall k = 0, \ldots, N \tag{4-3}$$

Where $g(\boldsymbol{x})$ is the measurement function that maps the states to the measurements.

The standard for state estimation in quadrotor research until now has been the Extended Kalman Filter (EKF) as is described in 3-4-2. This extension to the famous Kalman Filter uses linearizations of the system dynamics to estimate the system matrices at the operating points. Although this method is successfully used in a lot of applications, it is prone to errors and might even start diverging from the actual states. Besides this, the velocity estimate is very noisy in the case of only four beacons. Since most position controllers use the velocity as an input to the controller, the control output will not be stable in the case of noisy velocity estimates.

A different method has to be designed in order to generate more accurate positioning data that can be used for accurate flight and precision maneuvers. Options beside the EKF include the Unscented Kalman Filter, Particle Filters and Moving Horizon Estimation (MHE). In the Unscented Kalman Filter, the mean and covariance at each time-step are estimated by sampling some points around the previous estimate and propagating these through the nonlinear dynamic equations. In the case of highly nonlinear systems, this method shows to create a better estimate of the estimate covariance and does not require the calculation of the Jacobians. A Particle Filter uses a set of randomly chosen particles to estimate the states and noise distribution of a dynamic system. These filters can be used to estimate any form of noise distribution and is therefor useful in systems where the noise distribution of the measurements or the dynamic model is not known. The price to pay, however, is computational since a lot of particles need to be considered to get an accurate estimate. The MHE method is different to the previous methods in that it does not include the previous measurements in a state- and covariance estimate but it estimates the states based on a time-window of measurements. The standard MHE problem is given as follows:

$$\min_{\hat{\boldsymbol{x}}_i, i=0,\ldots,N} \frac{1}{2} \sum_{i=2}^{N} \|\boldsymbol{y}_i - g(\boldsymbol{x}_i)\|^2_{W_m} + \frac{1}{2} \sum_{i=1}^{N-1} \|\boldsymbol{x}_{i+1} - f(\boldsymbol{x}_i, \boldsymbol{u}_i)\|^2_{W_p} + \frac{1}{2} \|\boldsymbol{x}_1^{(-)} - \boldsymbol{x}_1\|^2_{W_a} \tag{4-4}$$

$$\text{s.t.} \quad l(\boldsymbol{x}_i) = 0 \quad \forall k = 0, \ldots, N \tag{4-5}$$

Where W_m, W_p, W_a are weighting matrices. The following notation is used: $\boldsymbol{a}^T W \boldsymbol{a} = \|\boldsymbol{a}\|^2_W$. This way, the optimal state at each time-step in the window will be estimated based on the weighting matrices. The first term is the measurement cost, the second the model cost and the last term is the arrival cost. This arrival cost encaptures all information from before the time-window. This method is computationally more expensive than Kalman filters, but with a short time-window and proper simplifications of the model real-time implementations are possible on small microprocessors.

A method based on a moving horizon has been chosen because it handles nonlinear models efficiently and reduces the effect of outliers. Finding a method based on the Moving Horizon Estimation (MHE)-principle that combines more past data could potentially be less susceptible to outliers and biases. In recent years, some work has been done on developing real-time applications of these kinds that estimate the complete state trajectory in a time-window [29]. In work by Li et al. [5], a different kind of method uses position measurements based on camera images to estimate the deviation from the dynamical model over a time-window instead of the complete state trajectory. This adaptation of the regular MHE utilizes assumptions on the errors that are introduced in the dynamic model. By assuming that the attitude estimates are biased, with a constant bias over a short time-frame, an offset in velocity and position can be estimated. Although in our research we do not have position measurements, a similar approach will be used which we shall call the Moving Horizon Model Parametrization Estimator.

4-2 Derivation of the Estimator

To obtain a real-time implementable filter that uses a number of measurements, a linear model of the quadrotor shall be used. This model is obtained by assuming a constant height and indoors operation so there is no wind. The free-body diagram that is obtained by this simplification for the x-axis is shown in Figure 4-2.

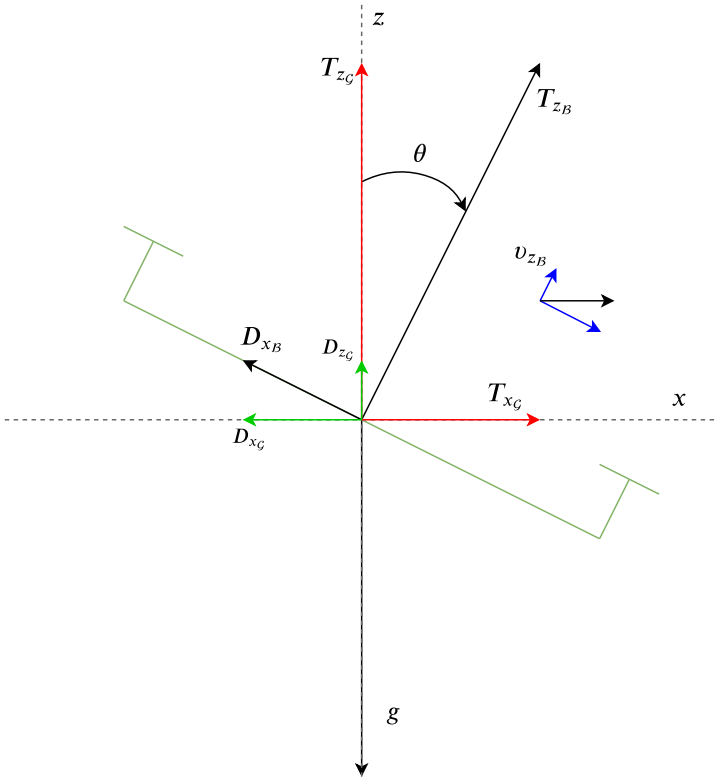


Figure 4-2: Free-body diagram

Since height is assumed to be constant, the sum of the thrust and drag in z-direction can be assumed to be equal to the gravity g.

$$T_{z_\mathcal{G}} + D_{z_\mathcal{G}} = g \tag{4-6}$$

The acceleration in x-direction can now be calculated as a function of $T_{x_\mathcal{G}}$ and $D_{x_\mathcal{G}}$.

$$a_{x_\mathcal{G}} = T_{x_\mathcal{G}} - D_{x_\mathcal{G}} \tag{4-7}$$
$$a_{x_\mathcal{G}} = -g\tan(\theta) - \upsilon_{x_\mathcal{G}}\sin^2(\theta)k_{\text{aero}} - \upsilon_{x_\mathcal{G}}\cos^2(\theta)k_{\text{aero}} \tag{4-8}$$

Since we will be flying with a pitch and roll angle below 20 deg, we can consider $\sin^2(\theta) \approx 0$ and $\cos^2(\theta) \approx 1$, which reduces the acceleration model even further. This definition can be used in the y-direction as well and since in this representation both axes are independent, we can define the following simplified model for the quadrotor

$$\dot{x}(t) = \upsilon_x(t) \tag{4-9}$$
$$\dot{y}(t) = \upsilon_y(t) \tag{4-10}$$
$$\dot{\upsilon}_x(t) = -g\tan\theta(t) - \upsilon_x(t)k_{\text{aero}} \tag{4-11}$$
$$\dot{\upsilon}_y(t) = g\tan\phi(t) - \upsilon_y(t)k_{\text{aero}} \tag{4-12}$$

Where θ [rad] and ϕ [rad] are the pitch and roll values that are used as inputs to the model. In our implementation, it is assumed that the attitude estimation is performed by the complementary filter that is available on the Crazyflie. In our comparison in Section 3-4-3, it was shown that the Madgwick-filter was more accurate and had a lower computational complexity, and therefor this filter will be used. This attitude can now be used as input to a localization algorithm. The parameter k_{aero} is the linear drag coefficient that is described in Section 3-4-1 and is assumed to be constant and the same for x and y. This value has to be tuned for the specific cases, but in the simulation tests the value is chosen to be $\frac{1}{m}K_{\text{aero}} \approx 3.7 \times 10^{-5}$. For the real tests, the drag is taken from [41] and is equal to $k_{\text{aero}} = 0.35$. Because in this model the position and velocity in x- and y-direction are independent of each other, the system can be decoupled and the analysis of the x-axis can be applied to the y-axis as well. In this model, it is assumed that yaw is constant at zero. Since this is not always the case in reality, and the roll and pitch values are expressed in the body-fixed frame, it is important to rotate these inputs with the estimated yaw before using as inputs. This can be done as follows:

$$\begin{bmatrix}\theta_\mathcal{G}\\ \phi_\mathcal{G}\end{bmatrix} = \begin{bmatrix}\cos(\psi) & -\sin(\psi)\\ \sin(\psi) & \cos(\psi)\end{bmatrix}\begin{bmatrix}\theta_\mathcal{B}\\ \phi_\mathcal{B}\end{bmatrix} \tag{4-13}$$

Where ψ is the estimated yaw value from the complementary filter.

In a state-space representation like Equation 3-1 this decoupled continuous-time system for the x-axis is as follows:

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix}0 & 1\\ 0 & -k_{\text{aero}}\end{bmatrix}\boldsymbol{x}(t) + \begin{bmatrix}0\\ -g\end{bmatrix}\tan\theta(t) \tag{4-14}$$

By using the proposed discretization method from Section 3-1-3 the discrete-time system can be described as follows:

$$\boldsymbol{x}_{k+1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 - k_{\text{aero}}\Delta t \end{bmatrix} \boldsymbol{x}_k + \begin{bmatrix} 0 \\ -g\Delta t \end{bmatrix} \tan\theta_k \tag{4-15}$$

Where Δt is the discrete-time time-step.

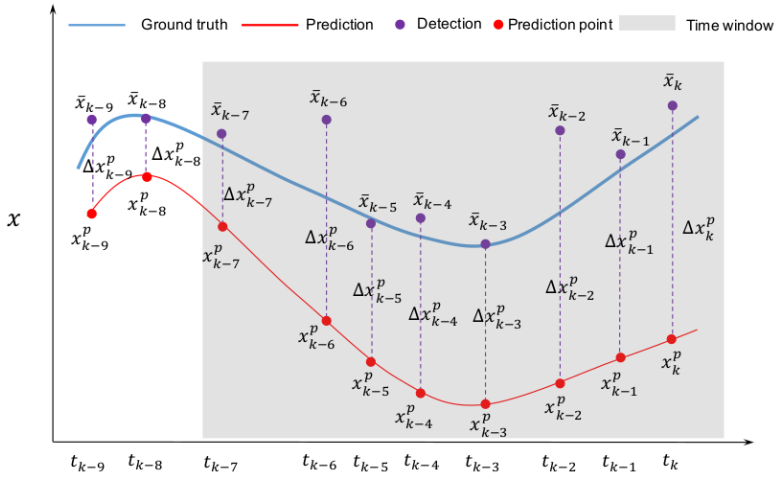


Figure 4-3: Visual representation of the MH-MP-method with diverging prediction with prediction denoted with a (x^p) and measurement with a ($\bar{x}$) (Adopted from [5])

As can be seen in Figure 4-3, the predicted values for x_k can be adjusted by estimating an offset $\Delta\boldsymbol{x}$ by using a sliding time-window. While the predicted states diverge because of the bias in the AHRS-method and errors in the prediction model, the offset will correct for this. In this chapter, two notations will be used to differentiate between predictions and measurements. States predicted with our estimator will be denoted with a hat ($\hat{\boldsymbol{x}}$) while position measurements obtained with the UWB sensors are denoted with a tilde ($\tilde{\boldsymbol{x}}$). At every time-step, the predicted $\hat{x}_{k|k-1}, \hat{v}_{x,k|k-1}$ are calculated using the value of $\hat{\boldsymbol{x}}_{k-1|k-2}$ and the input angles at time-step k.

Now a least-squares problem can be obtained by calculating the error between the prediction and the measurements at every time-step and using the following prediction error model [5]:

$$\Delta\hat{x}_{k-N+i} = \Delta\hat{x}_{k-N} + (t_{k-N+i} - t_{k-N})\Delta\hat{v}_{x,k-N} \tag{4-16}$$

Where N is the time-window and i is the position in this time-window. This way the following least-squares cost function can be defined:

$$J(\Delta\hat{x}_{k-N}, \Delta\hat{v}_{x,k-N}) = (A \begin{bmatrix} \Delta\hat{x}_{k-N} \\ \Delta\hat{v}_{x,k-N} \end{bmatrix} - \boldsymbol{b})^T (A \begin{bmatrix} \Delta\hat{x}_{k-N} \\ \Delta\hat{v}_{x,k-N} \end{bmatrix} - \boldsymbol{b}) \tag{4-17}$$

Where

$$A = \begin{bmatrix} 1 & t_{k-N} - t_{k-N} \\ 1 & t_{k-N+1} - t_{k-N} \\ \vdots & \vdots \\ 1 & t_k - t_{k-N} \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} \hat{x}_{k-N} - \tilde{x}_{k-N} \\ \hat{x}_{k-N+1} - \tilde{x}_{k-N+1} \\ \vdots \\ \hat{x}_k - \tilde{x}_k \end{bmatrix} \tag{4-18}$$

Where $\tilde{x}_k$ is the measured value of x at time k. By taking the derivative of this cost-function and setting this to zero, the least-squares solution can be found as:

$$\begin{bmatrix} \Delta\hat{x}_{k-N} \\ \Delta\hat{v}_{x,k-N} \end{bmatrix} = (A^T A)^{-1} A^T \boldsymbol{b} \tag{4-19}$$

After calculating the initial offset, the prediction at time k is updated by

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \begin{bmatrix} 1 & t_k - t_{k-N} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta\hat{x}_{k-N} \\ \Delta\hat{v}_{x,k-N} \end{bmatrix} \tag{4-20}$$

4-2-1 Obtaining position measurements

In the work by Li et al. [5], position measurements are obtained by locating gates with an on-board camera. Since we are using range-measurements, we have no direct measurement of the position $\tilde{x}_k$. To obtain measurements of x, we need to convert these range-measurements in some way. To do this, the following methods are suggested:

- Multilateration
- Subtraction of squared distances
- Projection of $\hat{x}_k$ on the distance circle

These methods will be described below.

Multilateration

The most straightforward method is multilateration. By taking at least four range measurements, the following system of quadratic equations can be arranged:

$$(x_1 - \tilde{x})^2 + (y_1 - \tilde{y})^2 + (z_1 - \tilde{z})^2 = d_1^2 \tag{4-21}$$
$$(x_2 - \tilde{x})^2 + (y_2 - \tilde{y})^2 + (z_2 - \tilde{z})^2 = d_2^2 \tag{4-22}$$
$$(x_3 - \tilde{x})^2 + (y_3 - \tilde{y})^2 + (z_3 - \tilde{z})^2 = d_3^2 \tag{4-23}$$
$$(x_4 - \tilde{x})^2 + (y_4 - \tilde{y})^2 + (z_4 - \tilde{z})^2 = d_4^2 \tag{4-24}$$
$$\tag{4-25}$$

Where $d_j \ \forall j = 1, ..., 4$ is the range measurement for anchor j and (x_j, y_j, z_j) is the location of the anchor. This can be arranged in matrix representation as:

$$\underbrace{\begin{bmatrix} 1 & -2x_1 & -2y_1 & -2z_1 \\ 1 & -2x_2 & -2y_2 & -2z_2 \\ 1 & -2x_3 & -2y_3 & -2z_3 \\ 1 & -2x_4 & -2y_4 & -2z_4 \end{bmatrix}}_{A} \begin{bmatrix} \tilde{x}^2 + \tilde{y}^2 + \tilde{z}^2 \\ \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} = \underbrace{\begin{bmatrix} d_1^2 - x_1^2 - y_1^2 - z_1^2 \\ d_2^2 - x_2^2 - y_2^2 - z_2^2 \\ d_3^2 - x_3^2 - y_3^2 - z_3^2 \\ d_4^2 - x_4^2 - y_4^2 - z_4^2 \end{bmatrix}}_{b} \tag{4-26}$$

Equation 4-26 can be described in the same way as Equation 4-17 to obtain an estimate of $(\tilde{x}, \tilde{y}, \tilde{z})$:

$$\begin{bmatrix} \tilde{x}^2 + \tilde{y}^2 + \tilde{z}^2 \\ \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} = (A^T A)^{-1} A^T \boldsymbol{b} \tag{4-27}$$

Since the configuration of the anchors as described in 2-2-2 does not allow for a reliable estimate of the z-height, the z-height in the least-squares solution can be fixed to 1.5m to increase the accuracy of the $(\tilde{x}, \tilde{y})$ estimate. The maximum error that is introduced to $(\tilde{x}, \tilde{y})$ by using a wrong z value can be easily estimated to be around 2.5cm for our configuration by knowing that the flying height is between 1m and 2m and using Pythagoras' rule. This way we can obtain a reliable estimate of $(\tilde{x}, \tilde{y})$.

Subtraction of range measurements

Another way to obtain a measurement of $(\tilde{x}, \tilde{y})$ is by subtracting two range-measurements from each other:

$$d_1^2 = (x_1 - \tilde{x})^2 + (y_1 - \hat{y})^2 + (z_1 - \hat{z})^2 \tag{4-28}$$
$$d_2^2 = (x_2 - \tilde{x})^2 + (y_2 - \hat{y})^2 + (z_2 - \hat{z})^2 \tag{4-29}$$
$$\tag{4-30}$$

When we subtract d_1^2 from d_2^2 and insert our predicted value of $\hat{y}$ and fix $\hat{z}$ at 1.5m, we get the following expression for $\tilde{x}$:

$$\tilde{x} = \frac{\frac{1}{2}(x_2^2 - x_1^2 + y_2^2 - y_1^2 + z_2^2 - z_1^2 + d_1^2 - d_2^2) - \hat{y}(y_2 - y_1) - \hat{z}(z_2 - z_1)}{x_2 - x_1} \tag{4-31}$$

The same can be done to obtain a measurement of $\tilde{y}$. This way, we can obtain two measurements of $\tilde{x}$ and $\tilde{y}$ with the same data, potentially increasing the possibility to remove outliers. One advantage of this method over multilateration is that it does not require the calculation of a 4×4 matrix-inverse. There are however also two disadvantages of this method that can be described. The first is that it needs at least two range measurements to create an estimate. Since measurements arrive asynchronously, it might take some time before two measurements are taken and the older measurement is already outdated. Besides, it also uses the estimated position which reduces the accuracy if the estimate of the position is incorrect.

Projection of $(\hat{x}, \hat{y})$ on the distance circle

The last option to create measurements of $\tilde{x}$ and $\tilde{y}$ is by projection of the estimate $\hat{x}, \hat{y}$ on the circle that is created by the range measurement as is shown in Figure 4-4.

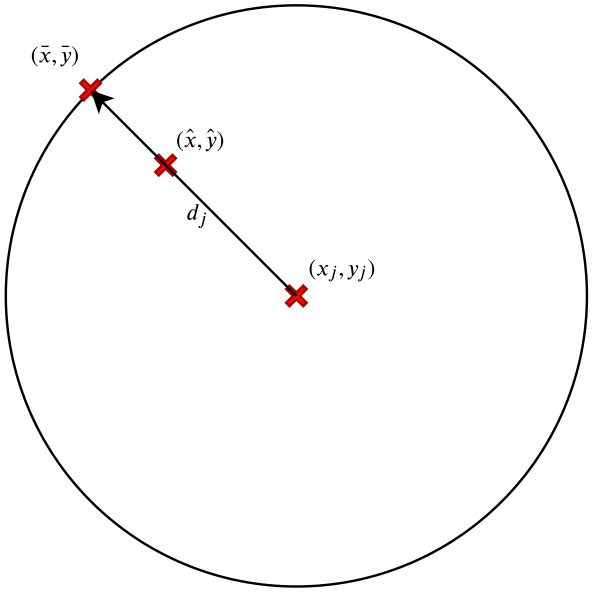


Figure 4-4: Visualization of the projection method

This is calculated in the following manner:

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} = \begin{bmatrix} x_j \\ y_j \end{bmatrix} + d_j \frac{\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} - \begin{bmatrix} x_j \\ y_j \end{bmatrix}}{\| \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} - \begin{bmatrix} x_j \\ y_j \end{bmatrix} \|_2} \tag{4-32}$$

Even more so than the previous method, this method relies on the estimates $\hat{x}$ and $\hat{y}$ and will therefor suffer when these estimates deteriorate. On the other hand, this methods supplies a linear estimate of the position and the estimation error declines as the distance between both sensors increases as long as the position estimate stays relatively close to the real position. This method is therefor especially useful in large areas with a large inter-sensor distance and is a really interesting method to linearize the non-linear range measurements by using the estimation of the current position.

All three methods will be implemented and tested to compare them.

4-2-2 Outlier rejection

Since UWB measurements suffer from outliers, it is important to deal with those properly. In the least-squares solution 4-17, all measurements are weighted equally so outliers will have large consequences for the estimation error. To implement a strategy for removing outliers, two possibilities are investigated. In the first, outliers are detected based on the Mahalanobis

distance and in the second, the same Random Sample Consensus (RANSAC) as was used in [5] is employed.

Mahalanobis distance

The Mahalanobis distance is a measure of how a sample $\boldsymbol{x}$ relates to a specific distribution. This distance is defined as a function of the mean μ of the distribution and the covariance matrix S as follows:

$$D_M = \sqrt{(\boldsymbol{x} - \boldsymbol{\mu})^T S^{-1} (\boldsymbol{x} - \boldsymbol{\mu})} \tag{4-33}$$

In the case of our range measurements, this can be utilized to determine outliers by rejecting all measurements with a Mahalanobis distance above a certain threshold. The mean $\boldsymbol{\mu}$ and the covariance matrix S can be calculated by using a time-window of measurements for that same beacon and calculating the mean and variance in this window. This is computational inexpensive since the range measurements are scalar and so is the mean and variance. This method is also implemented on the Crazyflie and a time-window of 32 samples and a threshold of 4 are used here.

Random Sample Consensus (RANSAC)

In Random Sample Consensus (RANSAC), not all the time-steps in the time-window will be included but only a subset will be used in the least-squares estimate. After calculating the A and b matrices in Equation 4-18, a random set of integers will be used to determine which rows will be included in the estimate. The amount of rows that will be used, depends on the value $N_s = N\sigma_s$ where N is the window size and σ_s the sampling ratio.

Using this subset, the offsets $\Delta\hat{x}^i_{k-N}$ and $\Delta\hat{v}^i_{x,k-N}$ can be calculated using Equation 4-19 where i is the ith RANSAC iteration. Using this offset, the prediction error ϵ_i can be calculated by

$$\epsilon_i = \sum_{j=k-N}^{k} \max\left(\|\Delta\hat{v}^i_{x,k-N}(t_{k-N+j} - t_{k-N}) + \Delta\hat{x}^i_{k-N} - (\hat{x}_{k-N} - \tilde{x}_{k_N})\|_2,\ \theta_m\right) \tag{4-34}$$

Where θ_m is a threshold for the maximum error that is added.

After all iterations i are done, the parameters $\Delta\hat{x}^i_{k-N}, \Delta\hat{v}^i_{x,k-N}$ that cause the minimal error ϵ_i will be chosen. This way, measurements that introduce a large error with relation to the model are not included in the estimation.

4-2-3 Prevention of overfitting

The basic least-squares solution as given in 4-19 calculates the solution that has the minimal sum of squared residuals, but this result can highly unrealistic in terms of the estimated velocity. To prevent overfitting on the measurements, two solutions are suggested: Regularization by adding a prior to the cost of $\Delta\boldsymbol{x}$ and generalized regularization.

Prior

In this method, the ordinary least-squares method is extended by adding a penalty matrix Q to $\Delta\boldsymbol{x}$ in the cost function $J(\Delta\boldsymbol{x})$ as follows:

$$J(\Delta\hat{\boldsymbol{x}}) = (A\Delta\hat{\boldsymbol{x}} - \boldsymbol{b})^T(A\Delta\hat{\boldsymbol{x}} - \boldsymbol{b}) + \Delta\hat{\boldsymbol{x}}^T Q\Delta\hat{\boldsymbol{x}} \tag{4-35}$$

Using this cost-function, and again taking the derivative and setting it to zero results in the following least-squares solution.

$$\begin{bmatrix} \Delta\hat{x}_{k-N} \\ \Delta\hat{v}_{x,k-N} \end{bmatrix} = (A^T A + Q)^{-1} A^T \boldsymbol{b} \tag{4-36}$$

This matrix Q can be used to put a penalty on the size of $\Delta\hat{\boldsymbol{x}}$. In the case of a diverging estimate $\hat{x}_{k-N}$, this solution will perform worse when the time increases, since $\Delta\hat{\boldsymbol{x}}$ should compensate this divergence. In this case, the following suggested method performs better.

Generalized Tikhonov Regularization

In generalized Tikhonov regularization, the least-squares method is further generalized by adding a cost matrix to the model term and by adding a penalty to the difference between $\Delta\hat{\boldsymbol{x}}$ and a prediction $\Delta\hat{\boldsymbol{x}}_0$

$$J(\Delta\hat{\boldsymbol{x}}) = (A\Delta\hat{\boldsymbol{x}} - \boldsymbol{b})^T P(A\Delta\hat{\boldsymbol{x}} - \boldsymbol{b}) + (\Delta\hat{\boldsymbol{x}} - \Delta\hat{\boldsymbol{x}}_0)^T Q(\Delta\hat{\boldsymbol{x}} - \Delta\hat{\boldsymbol{x}}_0) \tag{4-37}$$

Where P is the inverse covariance matrix of $\boldsymbol{b}$ and $\Delta\hat{\boldsymbol{x}}_0$ is taken to be the previous value of $\Delta\hat{\boldsymbol{x}}$. In this formulation, the divergence of the state estimate does not harm the least-squares solution and the variance of the measurements can be used. It can be noted that in this form, the least-squares problem is comparable to the standard Moving Horizon Estimation (MHE) definition, while only estimating 4 parameters instead of the complete state at every time-step.

Taking the derivative again and setting it to zero results in the following formulation of $\Delta\hat{\boldsymbol{x}}$

$$\Delta\hat{\boldsymbol{x}} = (A^T P A + Q)^{-1}(A^T P\boldsymbol{b} + Q\hat{\boldsymbol{x}}_0) \tag{4-38}$$

The covariance matrix P^{-1} depends on how the measurements are converted to position measurements, but in the case of the subtraction of range measurements, this matrix can be estimated. The measurements d_i can be modeled as a normal distribution with mean μ_i the average of the last couple of measurements and the variance σ_i can be calculated from these measurements as well. The variance of d_i^2 can now be calculated as:

$$\mathrm{Var}[d_i^2] = \mathrm{E}[d_i^4] - (\mathrm{E}[d_i^2])^2 \tag{4-39}$$

Where

$$\mathrm{E}[d_i^4] = \mu_i^4 + 6\mu_i^2\sigma_i^2 + 3\sigma_i^4 \tag{4-40}$$

and

$$\mathrm{E}[d_i^2] = \mu_i^2 + \sigma_i^2 \tag{4-41}$$

This results in the following variance for d_i^2:

$$\mathrm{Var}[d_i^2] = 4\mu_i^2\sigma_i^2 + 2\sigma_i^4 \tag{4-42}$$

4-3 Concluding Remarks

In this chapter, the Moving Horizon Model Parametrization (MH-MP) estimator is derived. This estimator is an adaptation of the standard Moving Horizon Estimation (MHE) method, without estimating the states at every step in the moving window. Instead, an initial offset in position and velocity is estimated, which significantly reduces the computational complexity. This estimator uses range measurements to gather an estimate of the position in three different ways. One uses all four measurements in multi-lateration, another one uses the subtraction of two measurements and the last one projects the current position estimate on the circle with the size of the distance measurement. To reduce overfitting to the measurement data, two methods are proposed. The first one uses a prior to penalize large offsets, while the second one penalizes the difference between two subsequent offset estimates. Outliers are being handled in this method in two ways. The first one uses the Mahalanobis distance and the second one uses RANSAC to exclude outliers.

In order to compare the performance of our estimator to the EKF, different scenarios need to be tested in simulation first.

Chapter 5

Results and analysis

In this section the proposed method will first be tested in the simulation environment as described in Section 3-3 and compared to the EKF-estimator that is running on the Crazyflie and a regular Moving Horizon Estimator. After that, the method will also be tested on real data that is obtained by logging flight data on a μSD-card. Both flight scenarios will resemble a real-life case in which the quadrotor is commanded to fly a certain trajectory in the operating space at a certain altitude. This simulates for instance a persistent surveillance task where the quadrotor is used in a greenhouse to sense vital variables such as temperature and humidity.

5-1 Benchmark Estimators

The results in this section will be compared to two estimators. The first one will be the EKF that is implemented on the Crazyflie, as discussed in Section 3-4-2. The other estimator will be a regular Moving Horizon Estimation (MHE). This estimator minimizes the following cost-function every time-step:

$$\min_{\boldsymbol{x}_i} \frac{1}{2}\sum_{i=2}^{N} \|\boldsymbol{y}_i - h(\boldsymbol{x}_i)\|^2_{W_m} + \frac{1}{2}\sum_{i=1}^{N-1} \|\boldsymbol{x}_{i+1} - f(\boldsymbol{x}_i, \boldsymbol{u}_i)\|^2_{W_p} + \frac{1}{2}\|\boldsymbol{x}_1^{(-)} - \boldsymbol{x}_1\|^2_{W_a} \tag{5-1}$$

Where $f(\boldsymbol{x}, \boldsymbol{u})$ is the dynamic model and $h(\boldsymbol{x})$ the measurement model. The model $f(\boldsymbol{x}, \boldsymbol{u})$ is taken to be the same as 4-15 and $h(\boldsymbol{x})$ is the distance from a beacon to the estimated position, described as:

$$h_j(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{x}_j\| \tag{5-2}$$

Where $\boldsymbol{x}_j$ is the position of anchor j. The measurement cost-matrix W_m was chosen to be the inverse of the covariance matrix of the measurements and the process and arrival cost-matrices were tuned empirically. In our tests, this nonlinear least-squares problem is solved by using the `lsqnonlin`-function in MATLAB.

5-2 Simulation Results

In the simulation scenario, the quadrotor is commanded to fly a square trajectory at an approximate altitude of 1.5m. The square with sides of 6m and corners $(-3,-3),(-3,3),(3,3),(3,-3)$. To simulate the uncertainty in the z-height, the commanded z-value will be distorted by low-pass filtered Gaussian noise with mean 1.5m and variance 0.6. After getting to the desired attitude in 1 second, the quadrotor is commanded to follow this square trajectory with a velocity of 1.5m/s. The simulation is run for 21 seconds, enough to complete the trajectory at least once. The simulation setup and with the trajectory can be seen in Figure 5-1.

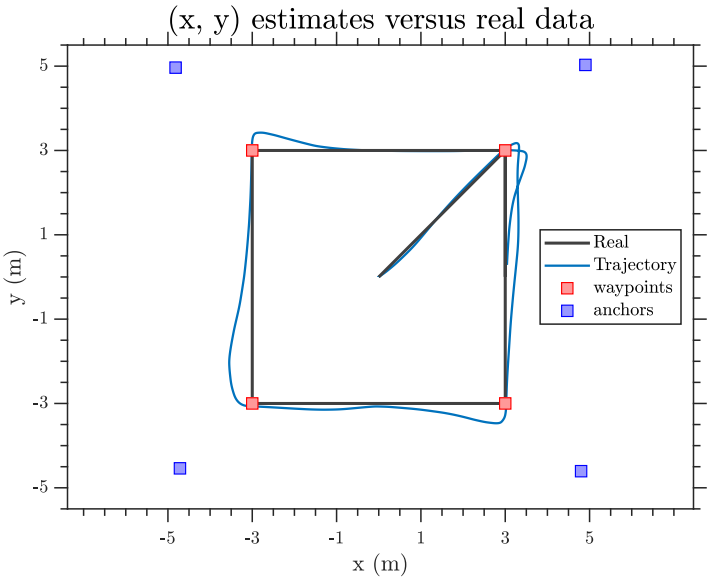


Figure 5-1: Visualization of the simulation setup, reference and flown trajectory

With the simulated measurement-data, the estimators can be compared offline. For both estimators it is assumed that the altimeter can be used to measure the height.

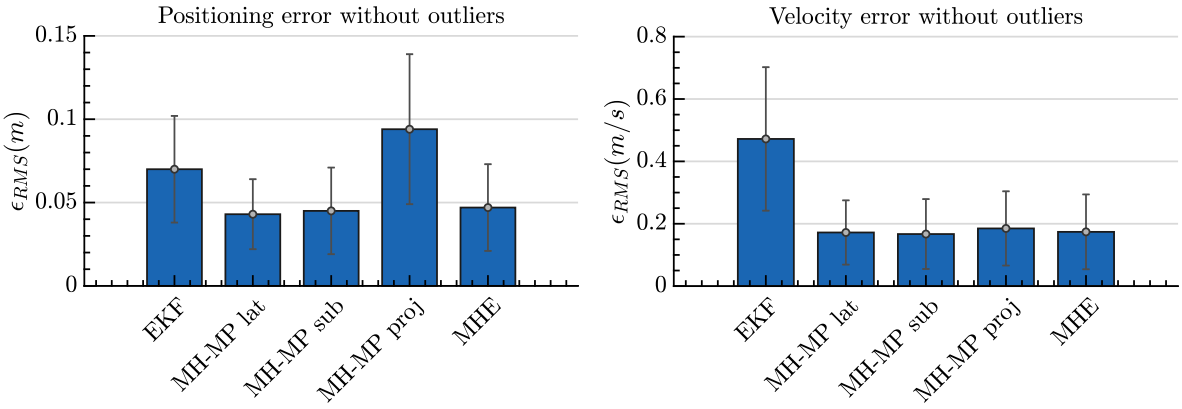
The performance measures are chosen to be the Root Mean Squared Error (RMSE) and the standard deviation. The RMSE is a measure of the total distance error to the real value and can be calculated with

$$\epsilon_{\text{RMS}}(\boldsymbol{x}) = \frac{\sum_{i=1}^{N} \sqrt{(\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i)^2}}{N} \tag{5-3}$$

Where $\hat{\boldsymbol{x}}$ is the estimated value of $\boldsymbol{x}$.

5-2-1 Gaussian measurement noise only

In the first test-case, only gaussian measurement noise is considered for the UWB-measurements. The simulation is run 5 times with a different seed for the measurement noise. In this comparison, the standard EKF is tested against the Moving Horizon Model Parametrization method with Generalized Tikhonov regularization. The least-squares method is used with a time-window of $N = 30$. All three methods of using the range measurements will be considered



(a) Positioning error comparison **(b)** Velocity error comparison

Figure 5-2: Positioning comparison in simulation without outliers. Standard deviation is depicted with errorbars.

and compared. The Q-matrix is chosen to penalize large differences in the velocity offset estimation $\Delta\hat{v}_{x,k-N}$ with respect to the previously estimated value. This matrix is different for every method of including range measurements and was tuned empirically.

$$Q_{\text{lat}} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad Q_{\text{sub}} = \begin{bmatrix} 0 & 0 \\ 0 & 0.5 \end{bmatrix}, \quad Q_{\text{proj}} = \begin{bmatrix} 0 & 0 \\ 0 & 100 \end{bmatrix} \tag{5-4}$$

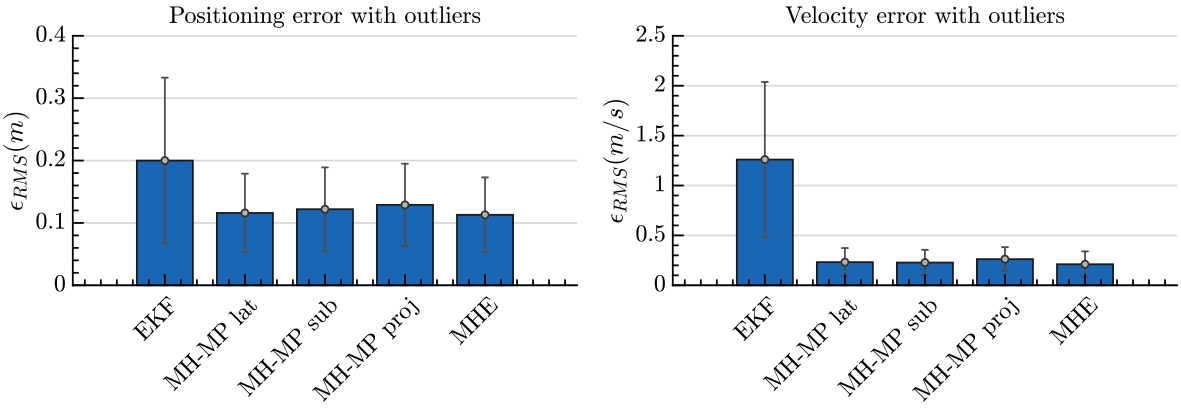
The average positioning error for 5 test-runs is shown in Figure 5-2a and the average velocity error is shown in Table 5-2b.

From this data, it is clear that all estimators can accurately estimate the position of the quadrotor and that the lateration method performs around 35% better in terms of absolute position estimate and standard deviation. The improvement is not very large, and the projection method even performs worse positioning wise. This can be attributed to the fact that our methods of including the range measurements assume constant height. When more outliers are introduced to the measurements, this effect gets lower but in this outlier-free data this effect is clearly visible. It is also interesting to compare the velocity estimations, as can be seen in Figure 5-2b. The velocity estimate improved with almost 65% with respect to the velocity estimate of the EKF for the lateration method.

5-2-2 Gaussian measurement noise and outliers

The second test-case uses the same measurements as the first test-case, but also includes outliers on the range-measurements. The outliers are as described in Section 3-3 with an outlier probability of 5%. Both estimators have no method to exclude outliers.

The average RMSE for 5 test-runs is shown in Figure 5-3a. This image shows that both the positioning error and velocity error decreased significantly. Especially the velocity estimate of the EKF deteriorated due to the outliers. In Figure 5-4, the influence of these outliers to the EKF (x, y)-estimate can be clearly seen, while the MH-MP does not suffer from outliers significantly.



(a) Positioning error comparison

(b) Velocity error comparison

Figure 5-3: Positioning comparison in simulation with outliers. Standard deviation is depicted with errorbars.

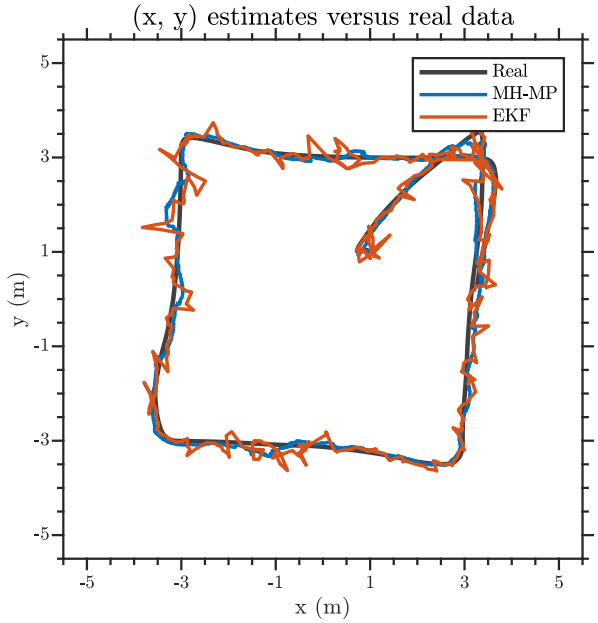


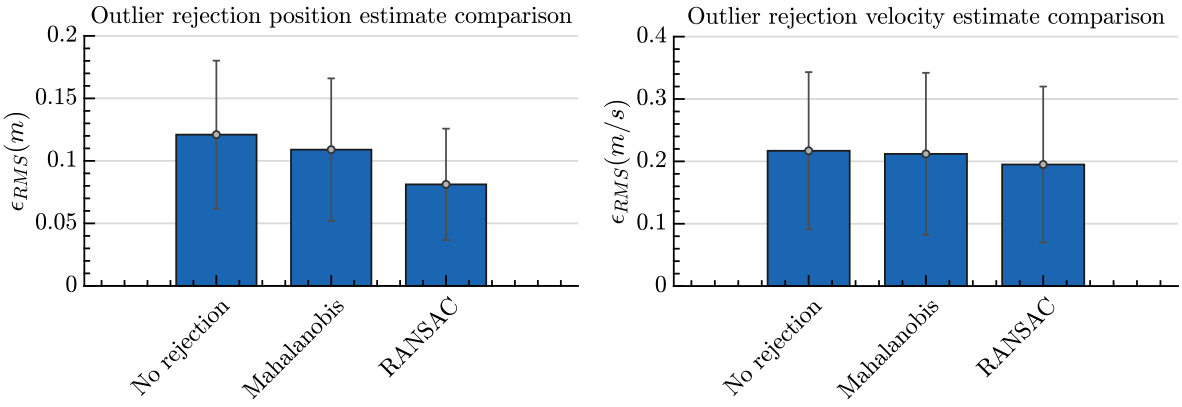
Figure 5-4: (x, y) plot with a comparison between the MH-MP and EKF for 5% outliers

From this data it is clear that even without outlier rejection, all the MH-MP methods outperform the EKF in both position and velocity estimate.

Outlier rejection

To improve the MH-MP even more, two suggestions were made in the previous chapter; the Mahalanobis distance and RANSAC. The lateration MH-MP from this simulation case with 5% outliers will therefore be tested with both outlier rejection methods to compare the results.

In Figure 5-5, the results of this comparison can be seen. The RANSAC-method improves the position estimate more than the Mahalanobis distance. The velocity estimate is barely



(a) Positioning error comparison **(b)** Velocity error comparison

Figure 5-5: Positioning comparison in simulation with outlier rejection by Mahalanobis distance and RANSAC. Standard deviation is depicted with errorbars.

affected by both methods. The only downside to the RANSAC method is the computation time. Since at every iteration a subset of all measurements is analysed for a fixed amount of iterations, the computation time increases linearly with this amount of iterations. The amount of iterations needs to be chosen with care, since an amount of iterations that is too low has a lower chance of excluding the outliers while a large number of iterations will increase the computation time.

5-2-3 Time-window comparison

As was also discussed in the previous part, the estimation error is largely dependent on the size of the time-window N. With a time-window that is too small, the influence of measurement-error will be very large, while a time-window that is too large will deteriorate because of errors in the diverging predicted model.

This comparison was made by running the simulation 5 times for a range of percentages. All the simulation data is used to calculate the error between the estimated and real position and velocity. The average RMSE value of these 5 simulations is taken and the results of this test can be seen in Figure 5-6.

It is clear that when the amount of outliers increases, the optimal window size grows larger. This is as was expected, since the influence of outliers now decreases. When an even larger time-window is chosen, the RMSE can be seen to rise again slowly especially for a lower amount of outliers, because the influence of errors in the prediction model increases.

5-2-4 Computation time comparison

In order to compare the results of the proposed method to the regular MHE and to test implementability on a real system, a computation time comparison has to be made. In this comparison, the computation time for both these methods will be calculated for an increasing window size. The computation time will be defined as the time it takes to process a new set of measurements and will be calculated by taking the average calculation time for a complete

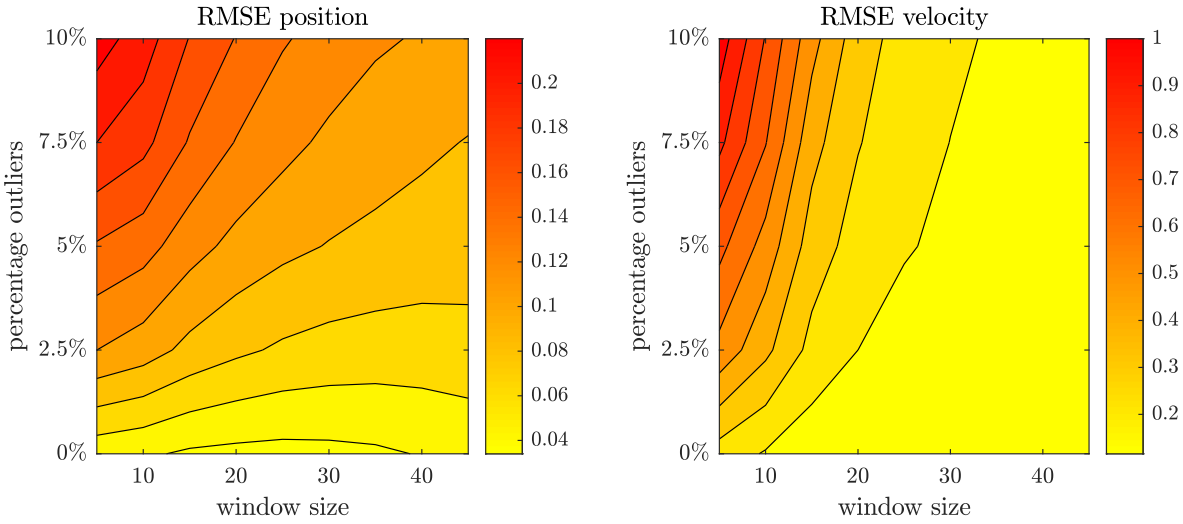


Figure 5-6: Comparison of the time-window versus outlier probability

test-run. The tests are run on a 2.40 Ghz quad-core laptop with 8 GB of RAM memory. The results of these tests can be seen in Figure 5-7. The black line shows the average data rate of the UWB measurements.

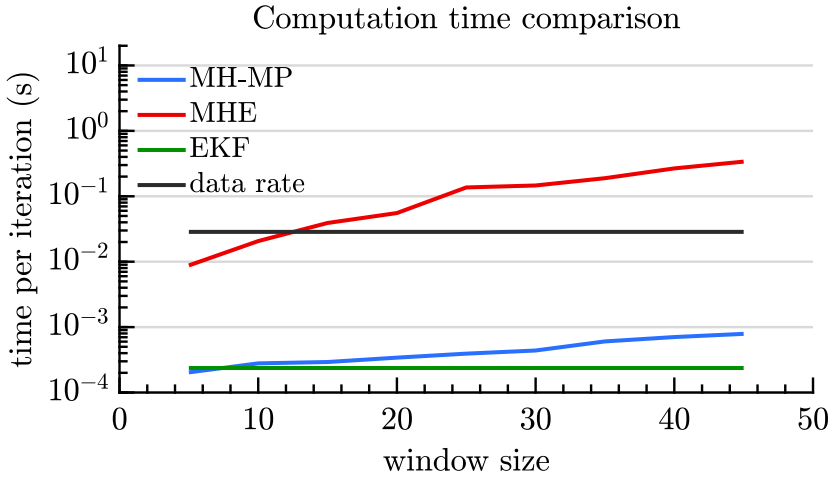


Figure 5-7: Comparison of the computation time for all three classes of estimators

It can be seen that the proposed method is able to run in real time, while the MHE would not be sufficient. It has to be noted that the MHE was not optimized for speed performance, but real-time implementations are designed in literature [29] that speed up the computations.

5-3 Real Test Results

The real tests aim to show that the derived method also works on the real system. In the first test, the Crazyflie was commanded to fly around the Cyberzoo at an approximate height of 2 meters. All sensor data was logged to an on-board μSD-card, so the estimators could

be tested offline. In the second test, the estimator was running in real-time on a laptop and the position and velocity estimates were send to the Crazyflie so that they could be used for control.

5-3-1 Offline data analysis

In total four measurement datasets will be investigated. These measurements all started with the Crazyflie on the ground at approximately $(x, y) = (0, 0)$. The Crazyflie was then controlled manually and flown around the Cyberzoo at around 1.5 meters high. Figure 5-8 shows the UWB measurements that were gathered during this test-flight. These measurements will be available to both the EKF and our designed method. Both estimators will have access to the measured OptiTrack z-height, to simulate the altimeter. This measured OptiTrack data is distorted with white Gaussian noise, similar to that present in the altimeter measurements.

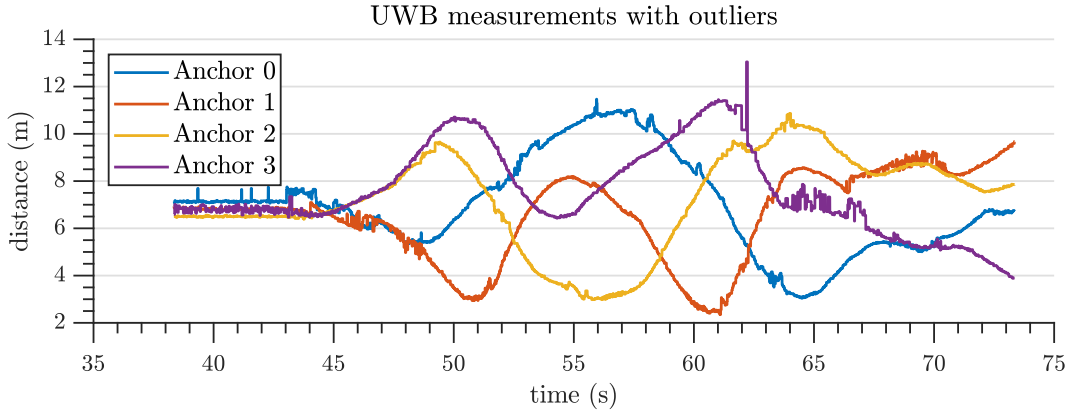


Figure 5-8: UWB range measurements with outliers

Besides the UWB measurements, both our designed method and the benchmark EKF will have access to the gyroscope and accelerometer measurements. We will investigate all three methods of using the distance measurements and outliers will be rejected based on the Mahalanobis distance with a threshold of 3.5. It should be added that the Mahalanobis distance was chosen to reject outliers in the real data. The RANSAC method would scale up the computation time and was not found to improve the estimate significantly. This might be because outliers in the real UWB are mostly grouped. This outlier behaviour is clearly visible in Figure 5-8, where Anchor 3 for example suffers from a group of outliers at around 65 seconds. The RANSAC method would not be able to remove these outliers, since they are all grouped together. Another method of removing outliers might be suggested but is out of scope for this research.

In Tables 5-1 and 5-2, it can be seen that our designed method decreases the ϵ_{RMS} for the position estimate with an average of 15% over the four testruns where the lateration method is used and for the velocity estimate even with 75% for the same method.

Projection method analysis

Some extra remarks about the performance of the projection method can be made. First, it is remarkable that although the position estimate is comparable to the EKF position estimate,

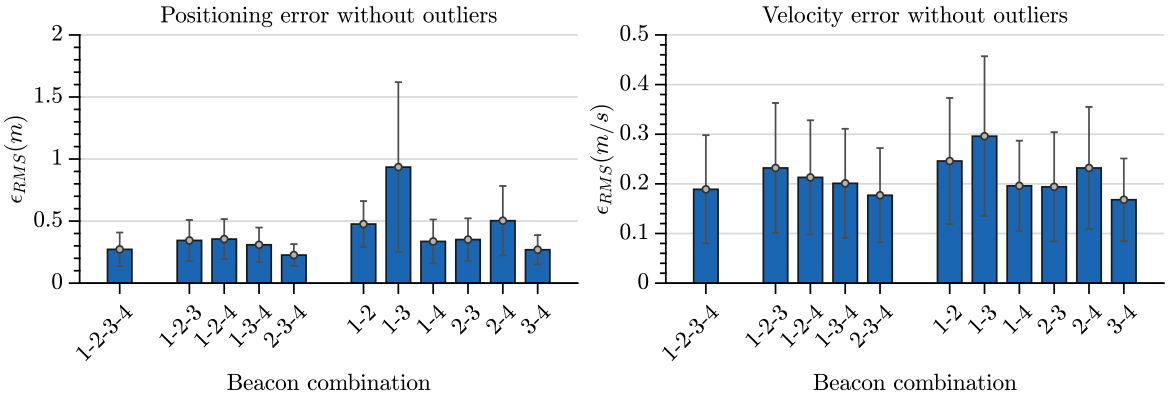
ϵ_{RMS} position				
test no.	1	2	3	4
EKF	0.258	0.230	0.265	0.336
MH-MP lat	0.214	0.235	0.178	0.287
MH-MP sub	0.219	0.255	0.190	0.334
MH-MP proj	0.274	0.243	0.233	0.303
MHE	0.209	0.197	0.180	0.265

Table 5-1: Positioning error of the Moving Horizon Estimation (MHE) versus Extended Kalman Filter (EKF) in the manual flight cases in meters.

ϵ_{RMS} velocity				
test no.	1	2	3	4
EKF	0.812	0.640	0.909	1.060
MH-MP lat	0.173	0.187	0.196	0.223
MH-MP sub	0.170	0.198	0.203	0.259
MH-MP proj	0.190	0.168	0.202	0.234
MHE	0.232	0.215	0.310	0.315

Table 5-2: Velocity error of the Moving Horizon Estimation (MHE) versus Extended Kalman Filter (EKF) in the manual flight cases in meters.

the velocity estimate is more accurate. This is most likely because the projection method has a higher update rate than the lateration and subtraction methods, since it does not require multiple measurements to be ready but can immediately process new measurements. Second, the projection method does not require all beacons to be available. This is a huge advantage compared to both other methods of transforming range measurements, since measurement dropouts now do not interrupt the estimator. The projection method was therefore tested with all possible combinations of two and three beacons. These results are visualized in Figure 5-9.



(a) Position error comparison

(b) Velocity error comparison

Figure 5-9: Comparison of the projection method with a combination of beacons

The leftmost bar in Figures 5-9a, 5-9b corresponds to the case where all beacons are used for estimation. For both the position and velocity estimate the estimation error for four beacons is even slightly larger than the error of the combinations of beacons 2-3-4 and beacons 3-4. The influence of beacon 1 on the estimation is therefore assumed to deteriorate the estimation.

The results further show that using only 3 beacons does not have a large effect on the performance of the estimator and that in some cases only choosing a subset of the available beacons even improves the position and velocity estimate. This can be accounted to the fact that some beacons suffer from less outliers and a lower bias than others and if only the well performing beacons are chosen the estimate can be improved.

5-3-2 Real-time data analysis

In order to test the real-time applicability of the proposed method, the simulation test-case was recreated on the real system. Due to time constraints, the method was not implemented on the real system but was made to run on a laptop that was connected to the Crazyflie. The Crazyflie sent its attitude estimation and the UWB range measurements over the radio connection. The position and velocity estimates were calculated on the laptop and sent to the Crazyflie in return. Besides this, also the setpoints for the position were send. This setup for the MH-MP tests and the data-rates are visualized in Figure 5-10.

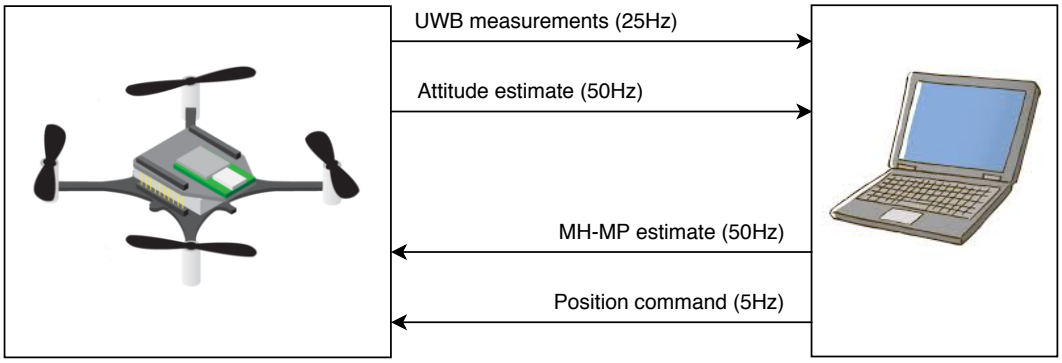


Figure 5-10: Real-time MH-MP test setup with the data-rates per channel

The EKF was also tested using the same setup. Since no calculations were necessary on the laptop, the Crazyflie was only sending the position and velocity estimates to the laptop and it only received the position commands.

For both tests, the position estimates and OptiTrack position and velocities were saved to compare both estimators. The results of two of these tests can be seen in 5-11.

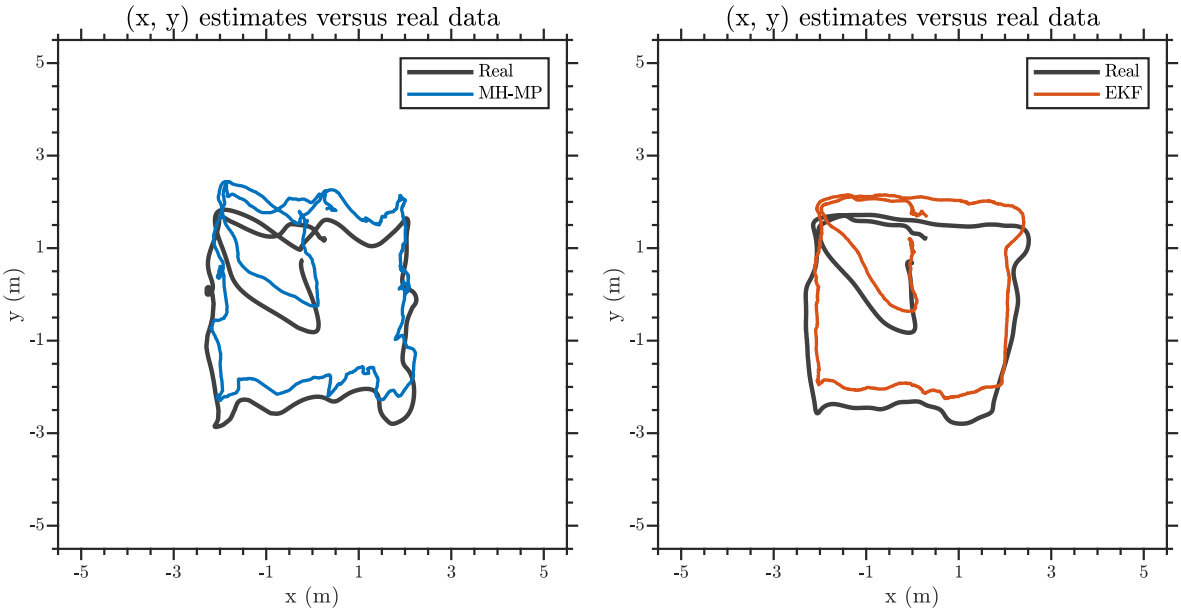
From these figures, it can be seen that the Crazyflie has a smoother flight with the EKF than with the MH-MP. This can also be seen in Table 5-3 where the RMSE of these tests is depicted.

	ϵ_{RMS} position	ϵ_{RMS} velocity
EKF	0.509m	0.186m/s
MH-MP	0.280m	0.365m/s

Table 5-3: Real-time data analysis results

Although our position estimate is better than the estimate with the EKF, the velocity estimate is worse. It is important to note that we are comparing the data of two separate test-flights. Since the quality of the range-measurements varies from test-flight to test-flight, no one-to-one comparison can be made. It is clear that our implementation is not enhancing the flight performance compared to the EKF in this comparison. A couple of arguments can be given for the differences in our observation:

On-board EKF versus off-board MH-MP
Since the EKF is implemented on-board, no communication delays are introduced to the estimator. In the case of the off-board MH-MP, the attitude estimates and UWB-measurements



(a) Flown trajectory with the Moving Horizon Model Parametrization (MH-MP)

(b) Flown trajectory with the Extended Kalman Filter (EKF)

Figure 5-11: (x, y) plot of the real flown trajectories and the estimates.

are sent to the laptop before calculations can be performed and after which the position and velocity estimate can be sent back. This can result in a considerable delay, that will influence both the position controller on the Crazyflie as the MH-MP. By comparing the real-time velocity estimate produced by the MH-MP to the velocity that was measured using the OptiTrack system, this delay can clearly be seen. This delay can be seen in a close up of the x-velocity in Figure 5-12. At the marked area, the delay is around 240ms.

This delay is not present in the analysis of the data that was gathered on the μSD card and suggests that the delay is caused by the communication.

Another effect of this difference in implementation is de update-frequency. The EKF has an update frequency of 500Hz, while the MH-MP is only updated at the frequency of the attitude estimates, which is 50Hz. Such a large difference might also influence the results.

MH-MP only tested with the lateration method
Currently, the MH-MP was only tested by using the lateration method to transform range-measurements into position measurements. This requires all four beacons to send a measurement, which can reduce the update frequency even more if the Crazyflie stops receiving measurements from a beacon for a certain time. The projection method and subtraction method also need to be tested in the real-time implementation because they offer a solution that is more robust to drop-out of one of the beacons.

OptiTrack is used to provide a z-measurement
Using the OptiTrack z-measurement will not be possible in a location without a MOCAP-system available. In this case, the altimeter from the Flowdeck will be used. On the other hand, the OptiTrack in our tests suffered from drop-outs a couple of times. This also significantly influences the performance of the position controller and might have changed the flight

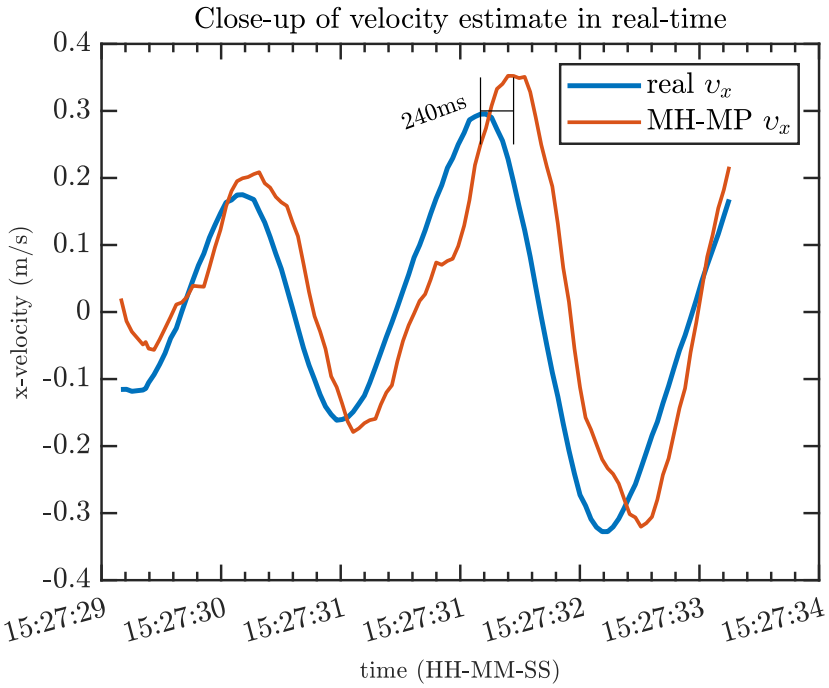


Figure 5-12: Close-up of the observed delay in the velocity estimated, presumably caused by the communication delay

performance in the tests.

Tuning of estimator parameters
The measurement window N, the tuning matrix Q and the drag factor k_{aero} were found to influence the performance of the MH-MP significantly. Since no extensive tuning was performed during the real-time tests, it might be that the chosen combination of parameters is sub-optimal.

Smoothness of flight
The MH-MP had a lot less smooth flight trajectory than the EKF. As was discussed above this might be because of a couple of reasons but this might also affect the performance of the estimator in return. This means that the error due to the communication delay might be amplified because the flight is less smooth.

5-4 Concluding Remarks

The designed estimator is tested in this chapter, compared to the EKF and MHE. In all the data that was gathered in simulation and real flight-data, the MH-MP was shown to outperform both benchmark estimators. Besides, the computation time was still in the same order of magnitude as the EKF, and is therefore implementable in real-time. The estimator was also tested in real-time, while the calculations were performed on a laptop and sent to the Crazyflie subsequently. This did not improve the flight performance compared to the EKF, although a qualitive comparison was not possible considering the EKF runs on-board while the MH-MP was not.

Chapter 6

Conclusion and recommendations

In this final chapter, the main conclusions drawn from this thesis are presented. The content of this thesis will be summarized, answers to the research questions will be discussed and a few recommendations for future work, along with some guidelines, will be presented.

6-1 Summary

This research aimed to design a method that uses a time-window of range measurements to estimate the states of a micro-quadrotor in real-time. Based on simulations and an offline investigation of data acquired with the Crazyflie, it can be concluded that the designed Moving Horizon Model Parametrization (MH-MP) improves the state estimates compared to the EKF and is still real-time implementable.

First the Ultra Wideband (UWB) localization system and its properties were investigated to obtain knowledge on the limitations imposed by our system. Especially the offset between the clocks of both transmitter and receiver and the non-isotropic radiation pattern of the antennae are important factors in obtaining accurate measurements. After discussing the calibration of the UWB-sensors, two distinct bias characteristics are discussed. A three-dimensional model was estimated to reduce the spatially varying bias, and was tested on validation data.

Since the research platform in this thesis is a quadrotor, also the quadrotor dynamics are discussed. With these dynamics and models for all the different sensors that are available to our estimator, a simulation environment is designed in Simulink. The specific quadrotor that will be used in this research is discussed. Technical specifications and mappings from input to output are summarized so they can be used in our simulator and estimator. The EKF-estimator and both complementary filters are discussed. The EKF will be used as a benchmark estimator and a complementary filter shall provide our estimator with attitude estimates. The EKF that is implemented on the Crazyflie is heavily influenced by outliers in the range measurements. To reduce this influence, an estimation method that uses a time-window of measurements will be used. The micro-processor on-board of the Crazyflie does not have the computation power necessary to implement the standard MHE, so an alternative

is proposed. This alternative is an adaptation of the Visual Model-predictive Localization method, that uses position estimates from camera images [5]. In this work, we assume that the quadrotor does not have access to a camera and is limited to range measurements. The method is denominated the Moving Horizon Model Parametrization (MH-MP), since not all the states are estimated at every time-step, but only the offset parameters from the initial position and velocity in the time-window. Three methods for using the range measurements to obtain a position measurement are discussed and implemented. Besides, methods of reducing the influence of outliers and methods to reduce overfitting to the measurement data are discussed. From the analysis of data that was obtained from simulation and real test-flights, it followed that this MH-MP method is implementable in real-time and produces accurate position and velocity estimates.

6-2 Answers to the Research Questions

The main research goal of this thesis was to implement a Moving Horizon inspired estimator that uses range-measurements and is able to run in real time on the Crazyflie. Besides, an investigation into the bias characteristics of the UWB-measurements in our setup was planned. The following answers to the initial research questions were the results of this thesis:

Design an estimation method that uses a time-window of range measurements for a micro-quadrotor by using Ultra Wideband (UWB) measurements
The proposed MH-MP has to be both implementable in real-time and produce accurate position and velocity measurements. In comparison to the EKF, the MH-MP increases the Root Mean Squared Error (RMSE) of the position with 15% and of the velocity with 75%. From test results, the estimator has a computation time comparable to the computation time of the EKF with a window of 10 samples but grows with an increasing window. However, the maximal window of 40 samples that was used in our tests still requires an estimation time that is lower than the average UWB data-rate. The Crazyflie was commanded to fly a square pattern by using position and velocity estimates from the MH-MP that were produced in real-time with success.

Design a low-dimensional bias model that can be used to reduce the error in the ranging measurements
A three-dimensional model has been designed and tested. This model reduced the Frobenius norm of the validation set with about 30-60% and reduced the standard deviation with about 25-50% for each beacon. Since this is a low-dimensional linear model, it can easily be used in online estimators and the model parameters might be estimated online.

6-3 Recommendations and Future Work

In this section some recommendations and guidelines for further research are discussed.

Implementation in the Crazyflie firmware
The main recommendation that follows from this research is the real-time implementation in the Crazyflie firmware. Our real-time tests showed the possibility of flying with our estimator, but in order to make a qualitative comparison to the EKF, both estimators should be tested

on-board. This also includes a study into the ideal window size N and tuning matrix Q, since these depend on the update frequency and method of implementing the range measurements.

Online estimation of the UWB bias parameters
In this research the bias parameters were estimated by examining gathered data offline. Reducing the spatially-varying bias with a simple model would even be more effective if the parameters of this model can be estimated online. As long as an over-determined UWB-configuration is used, patterns in the estimation correction might be used to estimate the bias of a specific anchor.

Online estimation of the accelerometer bias
The proposed estimator assumes that the accelerometer bias in a short time-window is constant and is the main consequence for the divergence of the prediction model. If this accelerometer bias can be estimated in-flight, the prediction model will be more accurate and this will influence the estimators performance.

Implement the MH-MP on alternative applications
Considering the linear nature of the proposed method, it could potentially also be successfully implemented on different robotic agents. This is especially the case for agents with an accurate prediction model that might suffer from data loss from time to time. An example of such an agent is the two-wheel-drive robot.

Extensive research into the projection method
Using the projection method to transform range-measurement into position-measurements was shown to produce accurate estimation results, even with only two beacons. This can significantly reduce the amount of beacons necessary in a large area while still maintaining a high accuracy. Although it was implemented and tested on simulated and real data, a thorough stability- and error-analysis should provide more details on the usability of this method in a real-time implementation.

Bibliography

[1] A. Ledergerber and R. D'Andrea, "Ultra-wideband range measurement model with Gaussian processes," in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 1929–1934, Aug 2017.

[2] J. Förster, "System identification of the Crazyflie 2.0 nano quadrocopter," Aug 2015. Bachelor's thesis, ETH Zurich.

[3] Bitcraze, "Measuring propeller RPM: Part 3," 2015. Retrieved Jan. 18, 2019 from https://www.bitcraze.io/2015/02/measuring-propeller-rpm-part-3/.

[4] M. Greiff, "Modelling and control of the crazyflie quadrotor for aggressive and autonomous flight by optical flow driven state estimation," Feb 2017. Master's thesis, Lund University.

[5] S. Li, E. van der Horst, P. Duernay, C. D. Wagter, and G. C. H. E. de Croon, "Visual model-predictive localization for computationally efficient autonomous racing of a 72-gram drone," *Computer Research Repository*, vol. 1905.10110, May 2019.

[6] R. D'Andrea and P. Wurman, "Future challenges of coordinating hundreds of autonomous vehicles in distribution facilities," in *2008 IEEE International Conference on Technologies for Practical Robot Applications*, pp. 80–83, Nov 2008.

[7] J. Roldán, G. Joossen, D. Sanz, J. del Cerro, and A. Barrientos, "Mini-UAV based sensory system for measuring environmental variables in greenhouses," *Sensors*, vol. 15, pp. 3334–50, Feb 2015.

[8] D. Greer, P. M. C. Kerrow, and J. Abrantes, "Robots in urban search and rescue operations," in *Proceedings of Australasian Conference on Robotics and Automation*, pp. 25–30, Nov 2002.

[9] N. Samama, *Global positioning: technologies and performance (Wiley Survival Guides in Engineering and Science)*. New York, NY, USA: Wiley-Interscience, 2008.

[10] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Precision flight control for a multi-vehicle quadrotor helicopter testbed," *Control Engineering Practice*, vol. 19, no. 9, pp. 1023 – 1036, 2011.

[11] U-Blox, *NEO-M8 u-blox M8 concurrent GNSS modules*, 10 2015. Retrieved Feb. 18, 2019 from https://www.u-blox.com/sites/default/files/NEO-M8_DataSheet_%28UBX-13003366%29.pdf.

[12] T.-M. Nguyen, A. Zaini, K. Guo, and L. Xie, "An ultra-wideband-based multi-UAV localization system in GPS-denied environments," in *2016 International Micro Air Vehicles Conference*, Oct 2016.

[13] S. Lupashin, M. Hehn, M. W. Mueller, A. P. Schoellig, M. Sherback, and R. D'Andrea, "A platform for aerial robotics research and demonstration: The flying machine arena," *Mechatronics*, vol. 24, no. 1, pp. 41–54, 2014.

[14] M. W. Mueller, M. Hamer, and R. D'Andrea, "Fusing ultra-wideband range measurements with accelerometers and rate gyroscopes for quadrocopter state estimation," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1730–1736, May 2015.

[15] D. Hoeller, A. Ledergerber, M. Hamer, and R. D'Andrea, "Augmenting ultra-wideband localization with computer vision for accurate flight," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 12734–12740, 2017.

[16] P. Li, M. Garratt, A. Lambert, M. Pickering, and J. Mitchell, "Onboard hover control of a quadrotor using template matching and optic flow," in *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*, p. 1, The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2013.

[17] DJI, *MAVIC PRO User Manual*, 2 2017. Retrieved Feb. 18, 2019 from https://dl.djicdn.com/downloads/mavic/20171219/Mavic%20Pro%20User%20Manual%-20V2.0.pdf.

[18] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, pp. 1067–1080, Nov 2007.

[19] V. Otsason, A. Varshavsky, A. LaMarca, and E. De Lara, "Accurate GSM indoor localization," in *International Conference on Ubiquitous Computing*, pp. 141–158, Springer, 2005.

[20] A. Kotanen, M. Hannikainen, H. Leppakoski, and T. D. Hamalainen, "Experiments on local positioning with Bluetooth," in *Proceedings ITCC 2003. International Conference on Information Technology: Coding and Computing*, pp. 297–303, April 2003.

[21] M. Coppola, K. N. McGuire, K. Y. W. Scheper, and G. C. H. E. de Croon, "On-board communication-based relative localization for collision avoidance in micro air vehicle teams," *Autonomous Robots*, vol. 42, pp. 1787–1805, Dec 2018.

[22] A. M. Ladd, K. E. Bekris, G. Marceau, A. Rudys, D. S. Wallach, and L. E. Kavraki, "Using wireless ethernet for localization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 402–408 vol.1, Sept 2002.

[23] A. M. Ladd, K. E. Bekris, A. P. Rudys, D. S. Wallach, and L. E. Kavraki, "On the feasibility of using wireless ethernet for indoor localization," *IEEE Transactions on Robotics and Automation*, vol. 20, pp. 555–559, June 2004.

[24] R. J. Fontana, E. Richley, and J. Barney, "Commercialization of an ultra wideband precision asset location system," in *IEEE Conference on Ultra Wideband Systems and Technologies, 2003*, pp. 369–373, Nov 2003.

[25] Decawave, *DW1000 User Manual*, 2017. Retrieved Jan. 15, 2019 from https://www.decawave.com/wp-content/uploads/2018/09/dw100020user20manual_0.pdf.

[26] S. van der Helm, "On-board range-based relative localization," Feb 2018. Master's thesis, TU Delft.

[27] A. Ledergerber, M. Hamer, and R. D'Andrea, "A robot self-localization system using one-way ultra-wideband communication," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3131–3137, Sept 2015.

[28] Bitcraze, "Crazyflie 2.0," 2019. Retrieved Apr. 18, 2019 from https://store.bitcraze.io/products/crazyflie-2-0.

[29] J. Vandersteen, M. Diehl, C. Aerts, and J. Swevers, "Spacecraft attitude estimation and sensor calibration using Moving Horizon Estimation," *Journal of Guidance, Control, and Dynamics*, vol. 36, pp. 734–742, May 2013.

[30] Y. Wan and T. Keviczky, "Real-time nonlinear moving horizon observer with pre-estimation for aircraft sensor fault detection and estimation," *International Journal of Robust and Nonlinear Control*, Dec 2017.

[31] M. Ghavami, L. B. Michael, and R. Kohno, *Ultra wideband signals and systems in communication engineering.*, vol. 2nd ed. Wiley, 2007.

[32] K. Yu, I. Sharp, and Y. J. Guo, *Ground-based wireless positioning*, vol. 5. John Wiley & Sons, 2009.

[33] R. Dalce, T. Val, and A. V. Bossche, "Comparison of indoor localization systems based on wireless communications," *Wireless Engineering and Technology*, vol. 2, no. 4, pp. 240–256, 2011.

[34] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: modeling, estimation, and control of quadrotor," *IEEE Robotics Automation Magazine*, vol. 19, pp. 20–32, Sept 2012.

[35] P. Martin and E. Salaün, "The true role of accelerometer feedback in quadrotor control," in *2010 IEEE International Conference on Robotics and Automation*, pp. 1623–1629, May 2010.

[36] C. Luis and J. L. Ny, "Design of a trajectory tracking controller for a nanoquadcopter," *Computing Research Repository*, vol. 1608.05786, Aug 2016.

[37] J. L. Crassidis, F. L. Markley, and Y. Cheng, "Survey of nonlinear attitude estimation methods," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 1, pp. 12–28, 2007.

[38] T.-A. Johansen and R. Kristiansen, "Quadrotor attitude estimation using adaptive fading multiplicative EKF," *2017 American Control Conference (ACC)*, pp. 1227–1232, 2017.

[39] T. Chen and B. A. Francis, *Optimal sampled-data control systems*. Berlin, Heidelberg: Springer-Verlag, 1995.

[40] M. Mori, "Approximation of exponential function of a matrix by continued fraction expansion," *Publications of the Research Institute for Mathematical Sciences*, vol. 10, Jan 1974.

[41] S. Li, C. De Wagter, and G. C. de Croon, "Unsupervised tuning of filter parameters without ground-truth applied to aerial robots," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4102–4107, 2019.

[42] W. Giernacki, M. Skwierczyński, W. Witwicki, P. Wroński, and P. Kozierski, "Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering," in *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*, pp. 37–42, Aug 2017.

[43] M. Verhaegen and V. Verdult, *Filtering and system identification: A least squares approach*. Cambridge University Press, 2007.

[44] G. A. Terejanu, "Extended Kalman Filtering." Retrieved 18-12-2018 from https://www.cse.sc.edu/~terejanu/files/tutorialEKF.pdf.

[45] M. W. Mueller, M. Hehn, and R. D'Andrea, "Covariance correction step for Kalman Filtering with an attitude," *Journal of Guidance, Control, and Dynamics*, vol. 40, pp. 1–7, Nov 2016.

[46] T. Hamel and R. Mahony, "Attitude estimation on SO [3] based on direct inertial measurements," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006*, pp. 2170–2175, 2006.

[47] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *2011 IEEE International Conference on Rehabilitation Robotics*, pp. 1–7, 2011.

Glossary

List of Acronyms

AHRS	Attitude and Heading Reference System
BLE	Bluetooth Low Energy
CPU	Central Processing Unit
EKF	Extended Kalman Filter
EM	Electro Magnetic
GPS	Global Positioning System
IMU	Inertial Measurement Unit
LOS	Line Of Sight
MEKF	Multiplicative Extended Kalman Filter
MHE	Moving Horizon Estimation
MH-MP	Moving Horizon Model Parametrization
MOCAP	Motion Capture
PWM	Pulse Width Modulation
RANSAC	Random Sample Consensus
RF	Radio Frequency
RMSE	Root Mean Squared Error
RPM	Rounds Per Minute
RSS	Received Signal Strength
TDOA	Time Difference Of Arrival

TOA	Time Of Arrrival
TOF	Time Of Flight
TU Delft	Delft University of Technology
TWR	Two Way Ranging
TWR-RR	Two Way Ranging with Repeated Reply
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter
UWB	Ultra Wideband
WLAN	Wireless Local Area Network

List of Symbols

α_q	Quaternion rotation angle
$\boldsymbol{\eta}_{\text{acc}}$	Accelerometer measurement noise
$\boldsymbol{\eta}_{\text{gdrift}}$	Gyro bias drift noise
$\boldsymbol{\eta}_{\text{gyro}}$	Gyro measurement noise
$\boldsymbol{\omega}$	Angular velocities of quadrotor in global frame
$\boldsymbol{\Psi}$	Extrinsic ZYX Tait-Bryan angles
$\boldsymbol{\tau}$	Moment applied to quadrotor by the rotors
$\boldsymbol{b}_{\text{acc}}$	Accelerometer bias
$\boldsymbol{b}_{\text{gyro}}$	Gyro bias
$\boldsymbol{e}_q$	Quaternion principal rotation axis
$\boldsymbol{p}$	(x,y,z)-position of quadrotor in global frame
$\boldsymbol{p}_{\text{des},k}$	Desired position at time k
$\boldsymbol{q}$	Quaternion vector
$\Delta t_{\text{antenna}}$	Antenna delay
$\dot{\theta}_i$	Angular velocity of rotor i
η_{uwb}	UWB distance measurement noise
$\hat{d}_{\text{opti, k}}$	Converted OptiTrack range measurement
ϕ	Quadrotor roll value
ρ	Density of air
PWM_i	Input signal for motor i
θ	Quadrotor pitch value
θ_{offset}	Clock offset between transmitting and receiving beacon
A_{r_i}	Rotor disk area
b_{uwb}	UWB distance bias

c	Speed of light
c_{Q}	Lumped torque parameter
C_{T}	Thrust coefficient
c_{T}	Lumped thrust parameter
d_{PWM}	PWM duty cycle
$d_{\mathrm{uwb,\ i}}$	Distance between quadrotor and beacon i
f	Thrust force of the quadrotor
I	Quadrotor inertia matrix
$k_{\perp}$	Force constant in the plane of the quadrotor
k_{aero}	Linear drag coefficient
m	Mass of the quadrotor
N	Moving Horizon Model Parametrization Time Window
r_i	Rotor radius
S	Set of (x,y) locations of all beacons
t_{delay}	Predetermined waiting time before the reply will be sent
t_{f}	Time of flight
t_{round}	Roundtrip time
t_d	Time of departure