Iterative Bias Estimation for an Ultra-Wideband Localization System

D.S. van der Heijden

Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Iterative Bias Estimation for an Ultra-Wideband Localization System

Master of Science Thesis

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

D.S. van der Heijden

December 1, 2019

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

The work in this thesis was supported by the Institute for Dynamic Systems and Control (IDSC) at ETH Zurich.

Abstract

Three bias estimation frameworks are presented that mitigate position-dependent ranging errors often present in ultra-wideband localization systems. State estimation and control are integrated, such that the positioning accuracy improves over iterations. The frameworks are experimentally evaluated on a quadcopter platform. Two state augmentation frameworks show that the anchor placement has a significant influence on the observability of the problem. A third framework circumvented any observability issues by using a classifier. This framework performed best as it improves the tracking performance with respect to ground truth, and also smoothens the overall flight by significantly reducing unwanted oscillations; see `https://youtu.be/J-htfbzf40U` for a video.

Table of Contents

Acknowledgements

First and foremost, I would like to thank my daily supervisors Tony and Rajan for their guidance throughout the thesis. They were always available for questions and familiarized me with the challenges of embedded systems. Their advice was essential in getting the results that are presented in this report. A special thanks to Tony for providing me with his code base (onboard C++ code of the quadcopter, the rigid body state estimator, and the trajectory fitting routine), which I could use as a starting point to further build upon. I would like to thank Sant Kumar for his path following controller code, which I used as a reference for my own controller.

I would like to thank Prof. D'Andrea in giving me the opportunity to conduct research in his group at the IDSC. All PhD and students in Prof. D'Andrea's group were great and made my exchange in Zurich very pleasant.

I would like to thank my supervisor Dr. ing. S. Grammatico for his supervision of the thesis and for his aid in arranging my exchange to ETH Zurich.

Finally, I would like to express my gratitude to my family and my beloved ones. They were encouraging when needed and showed patience when time was limited.

Delft, University of Technology — D.S. van der Heijden
December 1, 2019

Chapter 1

Introduction

Ultra-wideband (UWB) localization systems are one of the enabling technologies for indoor robotics [2]. Often, the time-of-flight of transmitted UWB radio signals is measured to acquire range measurements for positioning. Under non line-of-sight (NLOS) conditions, these time-of-flight measurements are usually biased. NLOS conditions frequently occur in indoor robotics and the resulting systematic errors in the range measurements limit the positioning accuracy of UWB localization systems, as described in [6].

Previous works addressed NLOS conditions by building channel classifiers that include additional information, such as floor plans of the environment in [16], channel impulse response data of the received UWB radio signal in [21] and references therein or models trained with labeled data in [15]. Other works mitigated range errors directly, by using deep learning on channel impulse response data in [23], or using tracking algorithms, while assuming a temporal evolution model for the range error in [7] and [10].

The scenario considered in this thesis is that of an autonomous robotic agent that is tasked to traverse a given reference path repeatedly. The reference path is a parametrized curve, defined as

$$\boldsymbol{\sigma}(\cdot) : [0, \lambda_{\max}] \rightarrow \mathbb{R}^3 \text{ with } \lambda_{\max} \in \mathbb{R}^+ \tag{1-1}$$

The agent moves autonomously, therefore requiring an estimate of its state $\boldsymbol{x}$ to determine the motion control input $\boldsymbol{u}$. The scenario is situated in a densely cluttered indoor environment, causing GPS localization and conventional UWB localization to be inaccurate. In addition to a UWB transceiver, the agent carries an inertial measurement unit (IMU) and barometer, that provide angular rates, accelerations, and altitude measurements. Though the agent is able to roughly traverse the path, the agent is unable to track the reference path accurately due to the aforementioned systematic measurement errors.

Unlike previous works, this paper exploits the repetitive nature of indoor robotic applications by improving the positioning accuracy over iterations. Three frameworks are proposed that combine estimation with control and can deal with non-static environments because the systematic range error is estimated adaptively. Apart from an IMU and barometer, the proposed frameworks do not require any additional information such as labeled data (e.g. from a motion

capture system) or floor plans. Two frameworks take a state augmentation approach, while one framework relies on a classifier. Moreover, two frameworks run over iterations, while one framework performs sequential updates.

The thesis report is organized as follows. In Chapter 2, we present experimentally obtained range error data from which a biased range measurement model is derived. In Chapter 3 the quadcopter platform is introduced that is used to experimentally validate the proposed frameworks. Then the proposed frameworks are presented. First the sequential bias estimator in Chapter 4 followed by the iterative bias estimator and reliable iterative bias estimator in Chapter 5 and Chapter 6, respectively. We conclude the thesis with a summary of the main findings and give suggestions for future research in Chapter 7. See Appendix G for a paper in which the main results of this thesis are summarized. The paper has been submitted to the 21st International Federation for Automatic Control (IFAC) World Congress, 2020.

Generally, column vectors are denoted by small, bold font letters, e.g. $\boldsymbol{p}$. Sets are denoted with calligraphic font letters, e.g. $\mathcal{Z}$. Matrices are denoted by capitalized, normal font letters, e.g. U. For the ease of notation, vectors are expressed as n-tuples $(x_1, x_2, ...)$, with dimension and stacking clear from context. The index set of measurements $\mathcal{Z}$ is defined as the union

$$\mathcal{Z} = \mathcal{Z}^{\text{uwb}} \cup \mathcal{Z}^{\text{acc}} \cup \mathcal{Z}^{\text{gyr}} \cup \mathcal{Z}^{\text{bar}}, \tag{1-2}$$

of the range, acceleration, angular rate, and altitude index measurement sets, respectively. General measurements are denoted as $\boldsymbol{z}_m, m \in \mathcal{Z}$, range measurements as $z_k, k \in \mathcal{Z}^{\text{uwb}}$, acceleration measurements as $\boldsymbol{z}_n, n \in \mathcal{Z}^{\text{acc}}$, angular rate measurements as $\boldsymbol{z}_w, w \in \mathcal{Z}^{\text{gyr}}$, and altitude measurements as $z_l, l \in \mathcal{Z}^{\text{bar}}$.

Chapter 2

Range Measurement Analysis

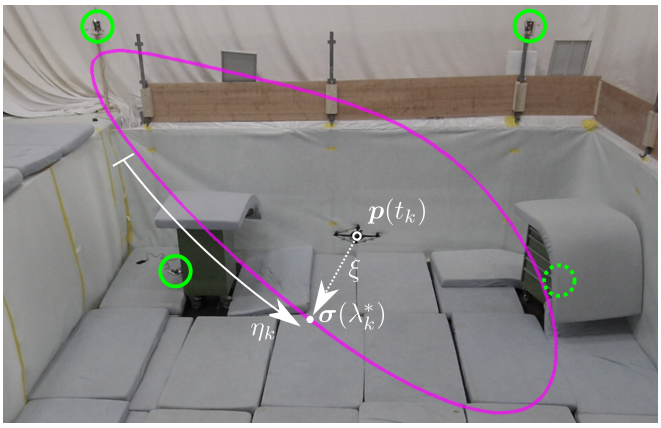

Figure 2-1: The state transformation is illustrated for an experimental set-up with a closed reference path $\boldsymbol{\sigma}$ (magenta). The locations of three visible anchors are marked with solid circles. One anchor is behind a metal trolley, and marked with a dashed circle. There are three other anchors not visible in the photo.

The ultra-wideband (UWB) localization system considered consists of multiple UWB transceivers placed at known locations $\boldsymbol{p}^a$, which are hereafter called anchors. Each anchor a provides the agent with the time-of-flight of transmitted radio signals, such that the distance towards the anchor can be inferred. The standard measurement model for such a range measurement $k \in \mathcal{Z}^{\text{uwb}}$ is

$$z_k = \| \boldsymbol{p}^{a_k} - \boldsymbol{p}(t_k) \| + \nu_k^{\text{uwb}}, \tag{2-1}$$

where z_k, t_k, and a_k denote the measured range, the timestamp and the anchor of the range measurement k, respectively. Furthermore, $\boldsymbol{p}(t_k)$ denotes the agent's position in the global frame at the time the range measurement k was taken. Often, the measurement noise ν_k^{uwb} is assumed to be zero mean, normally distributed white noise.

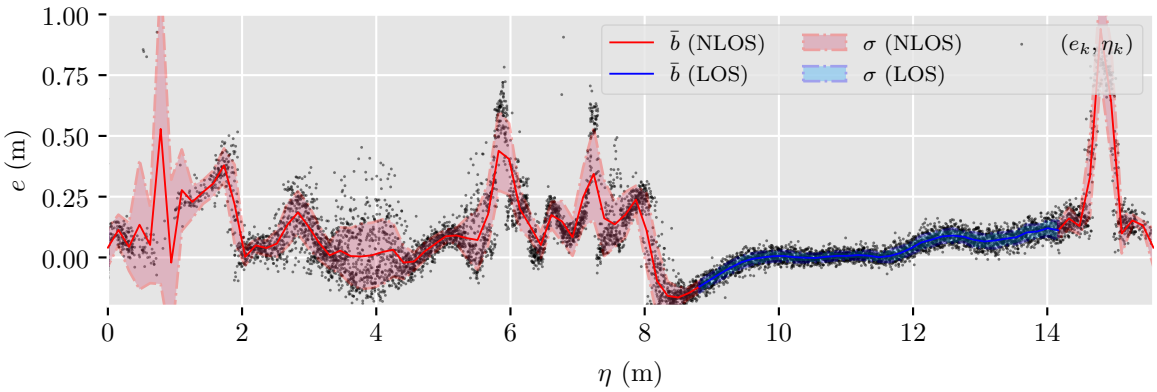

Figure 2-2: The range error set of the dashed anchor in Figure 2-1, with respect to the path variable η. Two piece-wise linear parametrizations (see Section 2-2-2) are fitted on the range error set (see (2-5)), using a least-squares approach with $N^b = 100$. The anchor has both LOS (blue) and NLOS (red) with parts of the path.

2-1 Position Dependent Bias

The model in (2-1) describes the ideal situation. However, in practice range measurements are often inaccurate. Main causes for range errors are propagation delays due to small manufacturing differences in the UWB modules, and NLOS conditions. The former results in a mostly constant error. In contrast, the latter is expected to vary with respect to the geometry of the environment, the anchor placement, and the agent's position. The error in the range measurement $k \in \mathcal{Z}^{\text{uwb}}$ is defined as

$$e_k = z_k - \| \boldsymbol{p}^{a_k} - \boldsymbol{p}(t_k) \| . \tag{2-2}$$

Consider the set-up in Figure 2-1. A quadcopter is commanded to traverse the path accurately, for 21 times, using motion-capture data and using a control approach as described in [11]. The quadcopter gathers a set of range measurements from the dashed anchor, which has NLOS with parts of the path. The motion-capture system provides the true position $\boldsymbol{p}(t_k)$ for all range measurements $k \in \mathcal{Z}^{\text{uwb}}$. The true position is used to calculate the range error e_k in (2-2), and a path variable η_k. The path variable η_k denotes the arc-length along the path, as visualized in Figure 2-1. The path variable is calculated with a state transformation, explained in more detail in [11]. First, the closest point to the reference path $\boldsymbol{\sigma}(\lambda_k^*)$ must be calculated with

$$\lambda_k^* = \underset{\lambda \in [0, \lambda_{\max}]}{\arg\min} \ \| \boldsymbol{p}(t_k) - \boldsymbol{\sigma}(\lambda) \|, \tag{2-3}$$

such that the path variable can be calculated with

$$\eta_k = \int_0^{\lambda_k^*} \left\| \frac{d\boldsymbol{\sigma}(r)}{dr} \right\| dr. \tag{2-4}$$

Hence, a set of range error e_k and path variable η_k pairs,

$$\mathcal{B} = \{(e_k, \eta_k) \text{ s.t. } k \in \mathcal{Z}^{\text{uwb}}\}, \tag{2-5}$$

is collected. Figure 2-2 shows the subset of $\mathcal{B}$ collected with the dashed anchor in Figure 2-1, and illustrates the spatial range error evolution along the path. It is evident that the range

error is systematic, caused by NLOS conditions, and dependent on the position along the path. When comparing the variance of the range error for positions along the path, the figure further reveals that there is a position-dependent noise component, in addition to the white noise ν_k^{uwb}.

2-2 Basic Framework

In this section a basic framework is presented that exploits the position dependency of the range errors. The basic framework will function as the starting point for the three proposed frameworks in the chapters 4, 5, and 6.

2-2-1 Biased Range Measurement Model

The model in (2-1) is extended with a path-dependent bias $b^{a_k}(\eta_k)$ to capture the position dependency of the range error,

$$z_k = \| \boldsymbol{p}^{a_k} - \boldsymbol{p}(t_k) \| + b^{a_k}(\eta_k) + \nu_k^{\text{uwb}}. \tag{2-6}$$

As the bias evolution is assumed to be a correlated process with respect to the position along the path, the mean and variance of $b^{a_k}(\eta_k)$ are parametrized as functions of the path variable η_k and a set of bias parameters $\boldsymbol{\theta}^{a,\bar{b}}, \boldsymbol{\theta}^{a,\sigma^2} \in \mathbb{R}^{N^b}$. For ease of notation, the anchor superscript a is dropped in $\boldsymbol{\theta}^{a,\bar{b}}, \boldsymbol{\theta}^{a,\sigma^2}$, but do note that each anchor has its own bias profile. We denote the parametrizations of its mean and variance by

$$\mathrm{E}[b(\eta_k)] = \bar{b}(\eta_k, \boldsymbol{\theta}^{\bar{b}}) \qquad \mathrm{Var}[b(\eta_k)] = \sigma^2(\eta_k, \boldsymbol{\theta}^{\sigma^2}), \tag{2-7}$$

and further specify this parametrization in Section 2-2-2. The number of parameters N^b per anchor is chosen such that a satisfactory fit on the underlying bias profile is obtained. See Figure 2-2 for an example of a piece-wise linear fit on the experimental data with a large number of parameters $N^b = 100$.

2-2-2 Bias Parametrization

There exist many different parametrizations that can be used to parametrize the mean and variance of b in (2-7). The parametrizations considered will be given in matrix form

$$y = \boldsymbol{h}^{\mathsf{T}}(\eta)\boldsymbol{\theta}, \tag{2-8}$$

where $\boldsymbol{\theta}$ is the parameter vector, y the output, and $\boldsymbol{h} \in \mathbb{R}^{N^b}$ the regressor vector defined as

$$\boldsymbol{h}^{\mathsf{T}}(\eta) = [h_1(\eta), h_2(\eta), \ldots, h_{N^b}(\eta)], \tag{2-9}$$

where the elements h_i are determined by the type of parametrization, discussed in the following paragraphs.

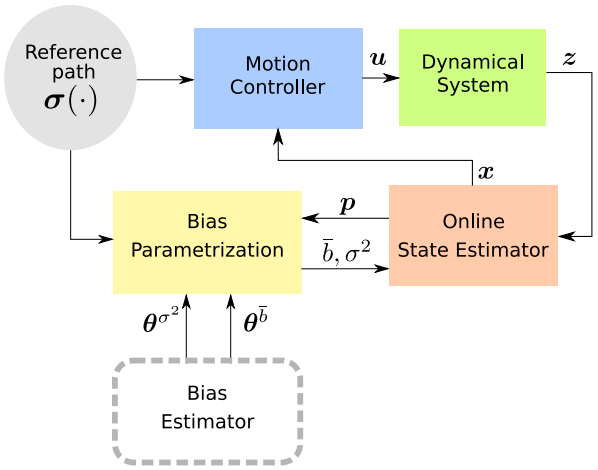

Figure 2-3: The proposed estimation and control architecture for controlling the motion of a dynamical system when the range measurements are corrupted by a position-dependent bias.

Piece-wise Linear Given an open reference path, the elements of vector $\boldsymbol{h}^{\mathsf{T}}(\eta)$ for a piece-wise linear parametrization are zero, except for

$$h_i(\eta) = 1 - \frac{\eta - \tau_i}{\tau_{i+1} - \tau_i} \qquad h_{i+1}(\eta) = \frac{\eta - \tau_i}{\tau_{i+1} - \tau_i}, \tag{2-10}$$

where i is s.t. $\tau_i \leq \eta < \tau_{i+1}$, and where the knots τ_i of the piece-wise linear parameterization are given as

$$\tau_i = \frac{\eta_{\max}}{N^b - 1}(i-1) \text{ for } i = \{1, \ldots, N^b\}, \tag{2-11}$$

where $\eta_{\max}$ is the arc-length of the reference path.

Truncated Fourier Series By truncating the infinite Fourier series up to a certain frequency C, a periodic signal can be approximated with a finite set of parameters $N^b = 2C + 1$. The following parametrization in real-form is used,

$$y = \theta_0 + \sum_{n=1}^{C} \theta_n \cos\left(2\pi n \frac{\eta}{\eta_{\max}}\right) + \theta_{-n} \sin\left(2\pi n \frac{\eta}{\eta_{\max}}\right). \tag{2-12}$$

Subsequently for $i \in \{1, 2, \ldots, N^b\}$, the regressor vector elements h_i can be defined as

$$h_i(\eta) = \begin{cases} \cos(2\pi \left\lfloor \frac{i}{2} \right\rfloor \frac{\eta}{\eta_{\max}}) & \text{, if } i = \text{odd} \\ \sin(2\pi \left\lfloor \frac{i}{2} \right\rfloor \frac{\eta}{\eta_{\max}}) & \text{, if } i = \text{even} \end{cases} \tag{2-13}$$

where $\lfloor \cdot \rfloor$ is the floor operator.

2-2-3 Control and Estimation Architecture

The control and estimation architecture for controlling the motion of a dynamical system in Figure 2-3 is used as the basic framework for the three proposed frameworks in this thesis.

Every timestep, the motion controller receives an estimate of the state $\boldsymbol{x}$ from the online state estimator, which it uses to determine a control input $\boldsymbol{u}$ that forces the agent's position on the path. The agent's sensor measurements are fused in the online state estimator. Given a set of bias parameters $\boldsymbol{\theta}^{\bar{b}}, \boldsymbol{\theta}^{\sigma^2}$, the state estimator can evaluate the bias parametrizations (2-7) for a given position $\boldsymbol{p}$ by calculating the path variable with (2-3), and (2-4). The evaluated bias can subsequently be incorporated into the range measurement update of the online state estimator. The bias estimator provides the bias parametrization block in Figure 2-3 with the parameters $\boldsymbol{\theta}^{\bar{b}}, \boldsymbol{\theta}^{\sigma^2}$ and is the main topic of the upcoming chapters.

Chapter 3

Quadcopter Platform

A quadcopter is used as the experimental platform for real world experiments. The quadcopter is the autonomous robotic agent, tasked to fly a given reference path repeatedly. See Figure 3-1 for the estimation and control architecture that is used to control the quadcopter. The online state estimator, with its components encircled by the dashed line in Figure 3-1, provides the motion controller with an estimate of the state $\boldsymbol{x}$, such that a suitable control input $\boldsymbol{u}$ is calculated that makes the quadcopter fly the reference path $\boldsymbol{\sigma}$. This chapter provides implementation details on the components in Figure 3-1 for the quadcopter platform. For details on the hardware that was used, the reader is referred to Appendix C.

3-1 Online State Estimator

The quadcopter is modeled as a rigid-body, meaning the state consists of the position $\boldsymbol{p}$, velocity $\boldsymbol{v}$, an attitude representation $R(\boldsymbol{\delta}) \in SO(3)$, and the angular velocity $\boldsymbol{\omega}$. Both $\boldsymbol{p}$ and $\boldsymbol{v}$ are defined in the global frame. The used attitude representation $R(\boldsymbol{\delta})$ is with respect to the global frame, and explained in detail in [18]. The online state estimator estimates the agent's state at 200 Hz, by fusing the range, accelerations, angular rates, and altitude measurements. The angular rates measurements are used directly as an estimate of the angular velocity, because the measurement noise covariance Σ^{gyr} is small. The remaining states

$$\boldsymbol{x} = (\boldsymbol{p}, \boldsymbol{v}, R(\boldsymbol{\delta})), \tag{3-1}$$

are estimated in a Kalman filter framework. The framework is initialized with the initial state $\boldsymbol{x}_0$ and initial covariance matrix P_0. Every timestep, the state estimate $\boldsymbol{x}$ and covariance matrix P are updated, as discussed in the following paragraphs.

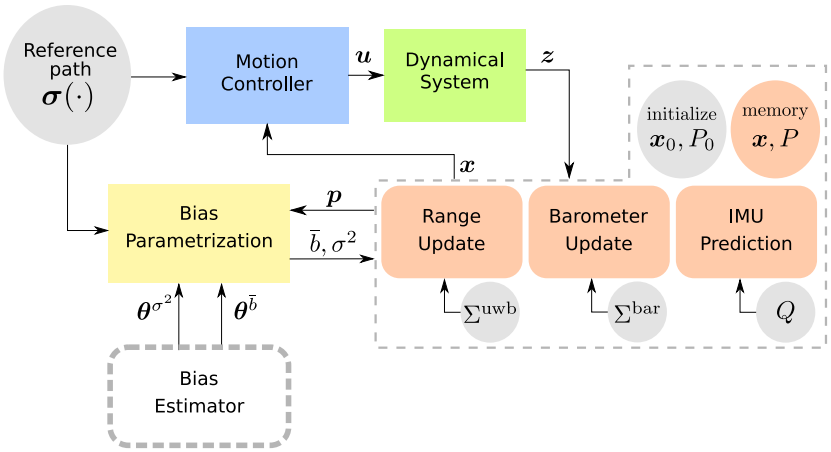

Figure 3-1: The implemented estimation and control architecture for controlling the motion of the quadcopter with the dashed line encircling the components of the online state estimator. The main tuning parameters of the online state estimator are contained in the grey circles. The quadcopter functions as the dynamical system.

IMU Prediction By using the acceleration and angular rates measurements as inputs to the system, the state is assumed to evolve as

$$\begin{aligned} \dot{\boldsymbol{p}} &= \boldsymbol{v}, \\ \dot{\boldsymbol{v}} &= R\boldsymbol{z}^{\text{acc}} + \boldsymbol{g}, \text{ with } \boldsymbol{z}^{\text{acc}} = \boldsymbol{a} + \nu^{\text{acc}} \\ \dot{R} &= R[\![\boldsymbol{z}^{\text{gyr}}]\!]_\times, \text{ with } \boldsymbol{z}^{\text{gyr}} = \boldsymbol{\omega} + \nu^{\text{gyr}} \end{aligned} \tag{3-2}$$

where $\boldsymbol{z}^{\text{acc}}$ and $\boldsymbol{z}^{\text{gyr}}$ are continuous acceleration and angular rates measurements, $\boldsymbol{a}$ is the agent's acceleration in the body frame, $\boldsymbol{g}$ is the gravitational acceleration expressed in the global frame and $[\![\boldsymbol{z}^{\text{gyr}}]\!]_\times$ denotes the matrix form of the cross product, defined such that $[\![\boldsymbol{z}^{\text{gyr}}]\!]_\times \boldsymbol{b} = \boldsymbol{z}^{\text{gyr}} \times \boldsymbol{b}$. The measurement noise ν^{acc} and ν^{gyr} are assumed to be zero mean, normally distributed white noise with noise covariance matrices Σ^{acc} and Σ^{gyr}, respectively.

The evolution model (3-2) is discretized, such that the accelerations $\mathcal{Z}^{\text{acc}}$ and angular rates $\mathcal{Z}^{\text{gyr}}$ can be used in the prediction step of an extended Kalman filter (EKF). The prediction step of an EKF is discussed in detail in [22]. The measurement noise covariance of the inertial measurement unit (IMU) is used to encode the statistics of the process noise, i.e. $Q = \text{diag}(\Sigma^{\text{acc}}, \Sigma^{\text{gyr}})$.

Barometer Update The barometer measures the enviromental pressure, from which the agent's altitude p_z, contained in the agent's state $\boldsymbol{x}$, can be inferred. See Appendix C on how pressure measurements are transformed to altitude measurements with the linearization of the isothermic barometric formula. The model

$$z_l = \underbrace{p_z(\boldsymbol{x})}_{h_l} + \nu^{\text{bar}} \tag{3-3}$$

is used in a linear Kalman filter measurement update [22] for the altitude measurements $l \in \mathcal{Z}^{\text{bar}}$. The measurement noise ν^{bar} is assumed to be zero mean, normally distributed white noise with noise covariance matrix Σ^{bar}.

Range Update The quadcopter communicates with every anchor in a sequential order, providing a range measurement of a different anchor at each timestep $t_k, k \in \mathcal{Z}^{\text{uwb}}$. The model

$$z_k = \underbrace{\| \boldsymbol{p}^{a_k} - \boldsymbol{p}(t_k) \| + b^{a_k}(\eta(\boldsymbol{p}(t_k)), \boldsymbol{\theta}^{a_k,\bar{b}}, \boldsymbol{\theta}^{a_k,\sigma^2})}_{h_k} + \nu_k^{\text{uwb}}. \tag{3-4}$$

is used in an unscented Kalman filter (UKF) measurement update for the range measurements $k \in \mathcal{Z}^{\text{uwb}}$, where $\eta_k(\boldsymbol{p}(t_k))$ is determined with (2-3) and (2-4). The measurement noise ν^{bar} is assumed to be zero mean, normally distributed white noise with covariance matrix Σ^{uwb}. For ease of notation, the anchor superscript a_k is dropped in b^{a_k},$\boldsymbol{\theta}^{a_k,\bar{b}}$, $\boldsymbol{\theta}^{a_k,\sigma^2}$, but do note that each anchor has its own bias profile. In the unscented transformation of the UKF, a set of $2n$ sigma points are chosen that capture the mean and variance of the agent's state $\boldsymbol{x} \in \mathbb{R}^n$. By taking the Cholesky decomposition of the covariance matrix, scaled with the state dimension, i.e., $nP = L^\mathsf{T} L$, the columns of L can be added and subtracted from the prior estimate of the agent's state $\boldsymbol{x}^-$ to form $2n$ sigma points. Subsequently, the sampled sigma points are propagated through the non-linear measurement model (3-4). For every position sigma point $\boldsymbol{p}_k^{(i)}$, the bias parametrization in (2-7) is evaluated and the resulting $(\bar{b}_k, \sigma_k^2)^{(i)}$ are incorporated into the propagation step. Then, a Gaussian distribution is fitted empirically on the propagated sigma points. Herein, the uncertainty in the bias parameters $\boldsymbol{\theta}^{\bar{b}}, \boldsymbol{\theta}^{\sigma^2}$ is not considered. The UKF measurement update step is explained in detail in [22]. To account for the position-dependent noise component, the formulas herein are adjusted as follows. Throughout, the shorthand notations $\bar{b}_k$ and σ_k^2 are used to denote $\bar{b}_k(\eta(\boldsymbol{p}_k), \boldsymbol{\theta}^{\bar{b}})$ and $\sigma_k^2(\eta(\boldsymbol{p}_k), \boldsymbol{\theta}^{\sigma^2})$, respectively. Furthermore, the bias b in (3-4) is split into a deterministic part $\bar{b}_k$ and random part $\nu^{\text{b}} \sim \mathcal{N}(0, \sigma_k^2)$, i.e.

$$z_k = \| \boldsymbol{p}^{a_k} - \boldsymbol{p}_k \| + \bar{b}_k + \nu_k^{\text{b}} + \nu_k^{\text{uwb}} \tag{3-5}$$

Also, use is made of $\mathbb{E}[\nu_k^{\text{b}}] = 0$, $\mathbb{E}[\nu_k^{\text{uwb}}] = 0$, and the assumption that ν_k^{b}, ν_k^{uwb} are independent random variables. The unscented transform approximates the expected measurement $\bar{y}$ with

$$\begin{aligned}
\bar{y}_k &= \mathbb{E}[\| \boldsymbol{p}^{a_k} - \boldsymbol{p}_k \| + \bar{b}_k + \nu_k^{\text{b}} + \nu_k^{\text{uwb}}], \\
&\approx \mathbb{E}[\frac{1}{2n} \sum_{i=1}^{2n} \| \boldsymbol{p}^{a_k} - \boldsymbol{p}_k^{(i)} \| + \bar{b}_k^{(i)} + \nu_k^{\text{b}^{(i)}}] + \mathbb{E}[\nu_k^{\text{uwb}}] = \frac{1}{2n} \sum_{i=1}^{2n} \| \boldsymbol{p}^{a_k} - \boldsymbol{p}_k^{(i)} \| + \bar{b}_k^{(i)}
\end{aligned} \tag{3-6}$$

The measurement covariance $P_{k,\text{y}}$ and cross covariance $P_{k,\text{xy}}$ are approximated with

$$\begin{aligned}
P_{k,\text{y}} &= \mathbb{E}[\big(\| \boldsymbol{p}^{a_k} - \boldsymbol{p}_k \| + \bar{b}_k + \nu_k^{\text{b}} + \nu_k^{\text{uwb}} - \bar{y}_k \big)^2], \\
&\approx \mathbb{E}[\frac{1}{2n} \sum_{i=1}^{2n} \big(\| \boldsymbol{p}^{a_k} - \boldsymbol{p}_k^{(i)} \| + \bar{b}_k^{(i)} + \nu_k^{\text{b}^{(i)}} + \nu_k^{\text{uwb}} - \bar{y}_k \big)^2], \\
&\approx \frac{1}{2n} \sum_{i=1}^{2n} \big(\| \boldsymbol{p}^{a_k} - \boldsymbol{p}_k^{(i)} \| + \bar{b}_k^{(i)} - \bar{y}_k \big)^2 + \sigma_k^{2^{(i)}} + \Sigma^{\text{uwb}},
\end{aligned} \tag{3-7}$$

$$\begin{aligned}
P_{k,\text{xy}} &= \mathbb{E}[(\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k)\big(\| \boldsymbol{p}^{a_k} - \boldsymbol{p}_k \| + \bar{b}_k + \nu_k^{\text{b}} + \nu_k^{\text{uwb}} - \bar{y}_k \big)], \\
&\approx \mathbb{E}[\frac{1}{2n} \sum_{i=1}^{2n} (\boldsymbol{x}_k^{(i)} - \boldsymbol{x}_k^-)\big(\| \boldsymbol{p}^{a_k} - \boldsymbol{p}_k^{(i)} \| + \bar{b}_k^{(i)} + \nu_k^{\text{b}^{(i)}} + \nu_k^{\text{uwb}} - \bar{y}_k \big)], \\
&\approx \frac{1}{2n} \sum_{i=1}^{2n} (\boldsymbol{x}_k^{(i)} - \boldsymbol{x}_k^-)\big(\| \boldsymbol{p}^{a_k} - \boldsymbol{p}_k^{(i)} \| + \bar{b}_k^{(i)} - \bar{y}_k \big)
\end{aligned} \tag{3-8}$$

Then, the prior state $\boldsymbol{x}_k^-$ and covariance matrix P_k^- are updated to obtain the a-posteriori state $\boldsymbol{x}_k^+$ and covariance matrix P_k^+ with

$$K_k = P_{k,\text{xy}} P_{k,y}^{-1} \qquad \boldsymbol{x}_k^+ = \boldsymbol{x}_k^- + K_k(z_k - \bar{y}_k) \qquad P_k^+ = P_k^- - K_k P_y K_k^\mathsf{T} \tag{3-9}$$

3-2 Motion Controller

Every timestep, the motion controller receives an estimate of the state $\boldsymbol{x}$ from the online state estimator, which it uses to determine a control input $\boldsymbol{u}$ that forces the agent's position on the path. A closed curve made from quintic spline-interpolating way-points is used as the reference path $\boldsymbol{\sigma}$. A path following controller (PFC) is implemented as the motion controller of the quadcopter. A PFC is a special type of controller that is parametrized in space instead of time. The control objective is two-fold. Given the position of the quadcopter, the controller minimizes the transversal distance ξ towards the path. Secondly, it wants to traverse along the path with a constant reference speed $\dot{\eta}_{\text{ref}} = 1.5$ m/s. See Appendix D for implementation details and the performance of the PFC. See [11] for a detailed description of the control approach.

3-3 Experimental Set-Up

The experiments were performed in the Flying Machine Arena of ETH Zurich, described in [13]. See Fig. 2-1 and Appendix A for the anchor placement and closed reference path the quadcopter is tasked to fly in the real experiments. In addition, the fictitious experimental set-up in Appendix B is used in simulation to investigate the effect of different experimental set-ups. Objects (e.g. metal trolleys) placed in the room cause some anchors to have both line-of-sight (LOS) and non line-of-sight (NLOS) with parts of the path, inducing a position-dependent bias in the range measurements. See Fig. 2-2 for an example.

3-4 Performance Evaluation

The agent traverses the reference path repeatedly, where every repetition is called an iteration. A motion-capture system tracks the global position and orientation of the quadcopter with an accuracy of the order of millimeters and milliradians at 200 Hz [13]. This system is used to asses the performance of the online state estimator and motion controller over iterations. The performance of the online estimator will be evaluated based on the mean absolute error (MAE) in the position $\boldsymbol{p}$ and velocity $\boldsymbol{v}$ over iterations. Minimizing the transversal distance towards the path is considered to be the most important control objective, because it ensures a safe flight. This transversal distance ξ is also referred to as the tracking error and is defined as

$$\xi = \| \boldsymbol{p} - \boldsymbol{\sigma}(\lambda^*) \| . \tag{3-10}$$

The first time-derivative of the tracking error is denoted by $\dot{\xi}$. A large $\dot{\xi}$ means the tracking error varies rapidly, which expresses itself in an oscillatory flight around the reference path. The flight performance will be evaluated based on the MAE of ξ and $\dot{\xi}$ over iterations.

The performance of the proposed framework is compared with that of two different experiments. In a base-case experiment, no bias estimation is performed. In a motion-capture experiment, the true range error set $\mathcal{B}$ is obtained with a motion-capture system, and subsequently fed to the RLS filter such that the best-possible bias fit is obtained. The performance of the motion-capture experiment is interpreted as the best case. Each experiment is executed multiple times. Only the mean performance μ with one standard deviation σ is plotted for the base-case and motion-capture experiments. See Appendix F-1 and F-2 for performance figures of the aforementioned experiments, and a short explanation on how the mean performances with one standard deviation was obtained.

Chapter 4

Sequential Bias Estimator

In a state augmentation approach, the agent's state is augmented with bias parameters such that both can be estimated simultaneously. State augmentation is a common approach for estimating biases [9]. An observability analysis can show whether it is even possible to estimate the augmented state given the sensor measurements and experimental set-up. In Appendix E an attempt was made to perform such an observability analysis. A thorough analysis proved to be a daunting task for non-trivial scenarios. The main goal of this thesis is to propose an effective bias mitigation framework that works in the real world. Therefore the choice was made to focus on experimentally demonstrating the feasibility of a state augmentation framework, instead of a theoretical analysis.

The sequential bias estimator (SBE) augments the agent's state $\boldsymbol{x}$ with the mean bias parameters $\boldsymbol{\theta}^{\bar{b}}$. The SBE estimates the augmented state $\tilde{\boldsymbol{x}} = (\boldsymbol{x}, \boldsymbol{\theta}^{\bar{b}})$ at every timestep in a Kalman filter framework. To allow for real-time estimation of the augmented state at 200 Hz on a platform with limited computational resources, the variance bias parameters $\boldsymbol{\theta}^{\sigma^2}$ are not included in the augmented state to reduce the augmented state dimension. This means that the range measurement noise is assumed to have constant variance. See (4-1) for the proposed estimation and control architecture. The Kalman filter framework of the SBE is similar to the framework of the online state estimator presented in Section 3-1, except for certain differences in the IMU prediction and range update blocks. The advantage of the SBE is that the sigma points in the range update are sampled from the distribution of the augmented state, as opposed to the range update in Section 3-1 where the sigma points are sampled from the distribution of the agent's state. Hence, the uncertainty in the bias parameters is addressed. In the following paragraphs, the differences in the IMU prediction and range update blocks with respect to Section 3-1 are discussed. Then, the implementation details of the framework on the quadcopter platform are provided, after which simulation and experimental results are presented.

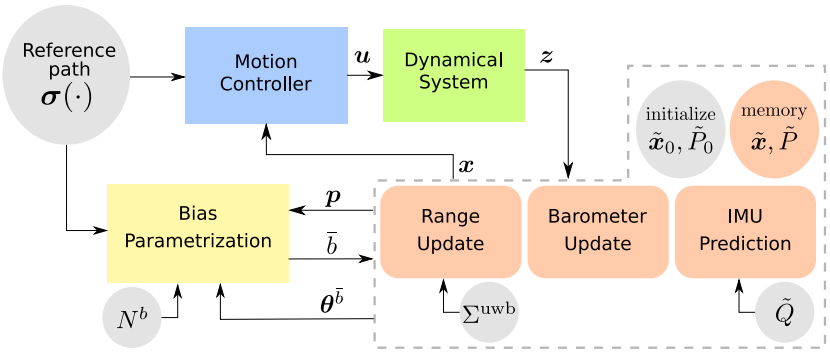

Figure 4-1: The estimation and control architecture for controlling the motion of a dynamical system with the dashed line encircling the components of the SBE. The main tuning parameters of the SBE are contained in the grey circles.

IMU Prediction The state evolution model in (3-2) is extended with a random walk process model for the bias parameters, defined as

$$\dot{\boldsymbol{\theta}}^{\bar{b}} = \nu^{\text{bias}}, \tag{4-1}$$

where the process noise ν^{bias} is assumed to be zero mean, normally distributed white noise with noise covariance matrix Σ^{bias}. The augmented process noise covariance matrix used in the IMU prediction step is then defined as $\tilde{Q} = \text{diag}(\Sigma^{\text{acc}}, \Sigma^{\text{gyr}}, \Sigma^{\text{bias}})$.

Range Update As mentioned in the introduction of this chapter, the position-dependent noise component under non line-of-sight (NLOS) conditions is omitted. Hence, the range measurement noise is assumed to have constant variance. This results in the model

$$z_k = \underbrace{\parallel \boldsymbol{p}^{a_k} - \boldsymbol{p}(t_k) \parallel + \bar{b}(\eta_k(\boldsymbol{p}(t_k)), \boldsymbol{\theta}^{\bar{b}})}_{h_k} + \nu_k^{\text{uwb}}. \tag{4-2}$$

that is used in the measurement update of the SBE for the range measurements $k \in \mathcal{Z}^{\text{uwb}}$. The spherical transformation is used to sample sigma points, which is described in detail in [22]. In the spherical transformation $n+1$ sigma points are sampled, which is the minimum number required to represent the mean and covariance matrix of the state, whereas the transformation discussed in Section 3-1 samples $2n$ sigma points. This reduces the computational load of the range update at the expense of reducing the accuracy of the measurement update. The a-posteriori estimate of the augmented state $\tilde{\boldsymbol{x}}_k^+$ and corresponding covariance matrix $\tilde{P}_k$ are obtained with (3-9) that is adjusted accordingly for the augmented state $\tilde{\boldsymbol{x}}$, and where the expected measurement $\bar{y}_k$, measurement covariance $\tilde{P}_{k,\text{y}}$, and cross covariance $\tilde{P}_{k,\text{xy}}$ are calculated with

$$\bar{y}_k = \frac{1}{n+1} \sum_{i=1}^{n+1} \parallel \boldsymbol{p}^{a_k} - \boldsymbol{p}_k^{(i)} \parallel + \bar{b}(\eta(\boldsymbol{p}_k^{(i)}), \boldsymbol{\theta}^{(i),\bar{b}}), \tag{4-3}$$

$$\tilde{P}_{k,\text{y}} = \frac{1}{n+1} \sum_{i=1}^{n+1} (\parallel \boldsymbol{p}^{a_k} - \boldsymbol{p}_k^{(i)} \parallel + \bar{b}(\eta(\boldsymbol{p}_k^{(i)}), \boldsymbol{\theta}^{(i),\bar{b}}) - \bar{y}_k)^2 + \Sigma^{\text{uwb}}, \tag{4-4}$$

$$\tilde{P}_{k,\text{xy}} = \frac{1}{n+1} \sum_{i=1}^{n+1} (\boldsymbol{x}^{(i)} - \tilde{\boldsymbol{x}}_k^-)(\parallel \boldsymbol{p}^{a_k} - \boldsymbol{p}_k^{(i)} \parallel + \bar{b}(\eta(\boldsymbol{p}_k^{(i)}), \boldsymbol{\theta}^{(i),\bar{b}}) - \bar{y}_k), \tag{4-5}$$

where $\tilde{\boldsymbol{x}}_k^-$ is the prior estimate of the augmented state.

4-1 Evaluation on a Quadcopter Platform

The SBE is evaluated on a quadcopter platform. First, the implementation details are discussed, after which simulation and experimental results are presented.

4-1-1 Implementation

The motion controller and SBE are run on a Snapdragon flight board, located onboard the quadcopter. Both the piece-wise linear and truncated Fourier series parametrizations in Section 2-2-2 were used to parametrize the bias, but no significant performance difference was observed. Therefore, only the results are presented for experiments where the truncated Fourier series was used to parametrize the bias $b^{a_k}(\eta_k(\boldsymbol{p}(t_k)), \boldsymbol{\theta}^{\bar{b}})$ in (4-2). The initial guess of the bias parameters $\boldsymbol{\theta}_0^{\bar{b}}$ is part of $\tilde{\boldsymbol{x}}_0$ and is set to zero, as there is no prior information on the shape of the bias. The measurement noise covariance matrix Σ^{uwb} is set to a value of 0.08^2, which is an overestimation of the range measurement noise under line-of-sight (LOS) conditions, to compensate for the omitted position-dependent noise component present under NLOS conditions. The number of bias parameters per anchor N^b is limited by the available computational resources. For the Snapdragon flight board, this was found to be a maximum of 77 bias parameters in total, so $N^b = 11$ in the case of 7 anchors. See Appendix A for the bias profiles of several anchors, optimally fitted in the least-squares sense with a truncated Fourier series parametrization and $N^b = 11$. In real experiments, the bias noise covariance matrix Σ^{bias} is taken as a diagonal matrix with the diagonal elements set to $(4\text{e}{-6})^2$. Similarly, the initial augmented covariance matrix $\tilde{P}_0$ is a diagonal matrix with the diagonal elements corresponding to the bias parameters set to 0.0012. In simulation, Σ^{bias} and $\tilde{P}_0$ are tuned accordingly to obtain the best performance.

4-1-2 Simulation Results and Discussion

In the simulations, the quadcopter is commanded to fly the reference path accurately using the control and estimation architecture in Figure 4-1. All sensors (IMU, anchors, barometer), the dynamics of the quadcopter, and the cascaded controller structure described in [11] are simulated in a single process. See Figure 4-2 for the results of three types of simulations (*match*, *mismatch*, *real*) that use the experimental set-up in Appendix A. In addition, the *real* simulation is run for the fictitious experimental set-up in Appendix B. In the following paragraphs the different simulations are explained, after which the results are presented and discussed.

Match In this simulation a deterministic scenario is considered, i.e. no measurement noise. The experimentally obtained range errors in Appendix A are fit with truncated Fourier series parametrizations with $N^b = 11$. The resulting mean parametrizations are used to simulated the mean bias for each anchor. The bias parametrization in the SBE also uses a truncated Fourier series with $N^b = 11$, meaning the models that are used to simulate and estimate the

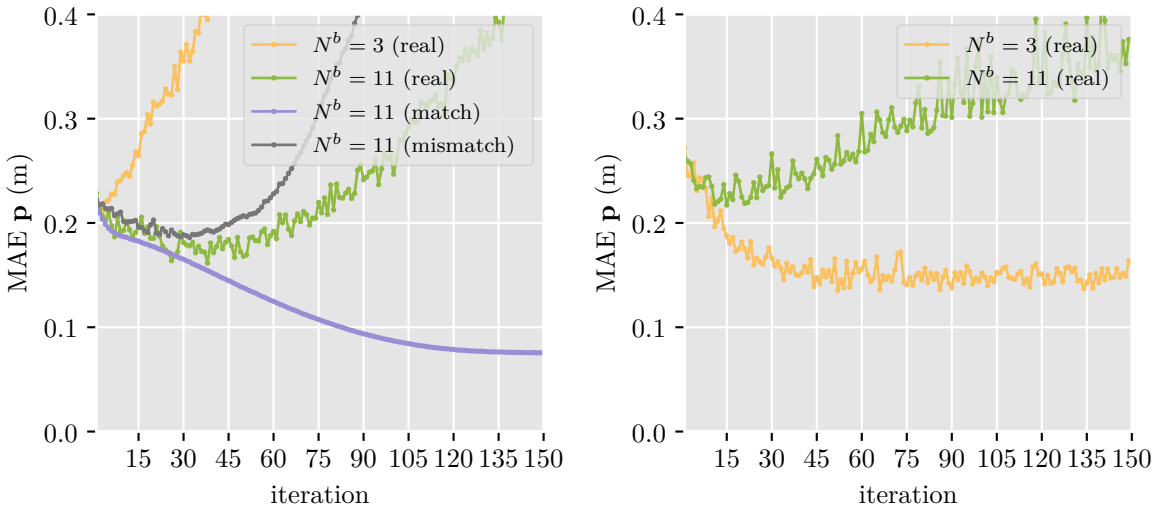

Figure 4-2: The estimation error in the position over iterations of the SBE in three different types of simulations, where the architecture in Figure 4-1 was used. Either the experimental set-up in Appendix A (left), or the experimental set-up in Appendix B (right) was used.

bias, match exactly. See the left plot in Figure 4-2 for the estimation error in the position indicated by *match*. Only the first 150 iterations are shown in Figure 4-2, but the simulation continued to run for over 10000 iterations. The estimation error in the position decreased rapidly in the first 8 iterations. Then, the error linearly decreased until iteration 150, after which the error increased only slightly and then remained stable for the rest of the simulation. A visual inspection of the estimated bias parametrizations after 100000 iterations showed that the estimated mean bias matches the simulated bias. It is therefore concluded that the augmented state $\tilde{x}$ is observable for this particular set-up in Appendix A, and the estimation of the bias parameters $\boldsymbol{\theta}^{\bar{b}}$ in a state augmentation framework is possible.

Mismatch To investigate the robustness of the SBE to model mismatch, the piece-wise linear parametrizations with $N^b = 100$ shown in Appendix A are used to simulate the mean bias, while the estimator still uses truncated Fourier series parametrizations with $N^b = 11$. This means that the simulated bias is closer to the experimentally observed bias and there exists a certain degree of model mismatch between the bias parametrization and simulated bias. See the left plot in Figure 4-2 for the estimation error in the position indicated by *mismatch*. The estimation error initially improves up until iteration 40 and then starts to diverge rapidly over iterations. The divergent behavior is attributed to the augmented state being weakly observable, as observability is generally a local property for non-linear systems [19]. This means the estimator is not robust against mismatches between the bias parametrization and simulated bias and can cause the estimator to diverge.

Real In the *mismatch* simulation, a mismatch between the bias parametrization and simulated bias was introduced. However, more factors can cause the estimator to diverge. In the *match* and *mismatch* simulations, the bias was simulated to vary with respect to the path variable η which was determined with ground-truth data. This simulated the variation of the bias in the tangential direction of the path. However, in practice, the bias is just as likely to

vary in the transversal direction, i.e. in the direction away from the path. Especially the first few iterations when the quality of the agent's state estimates is of lesser quality (because the bias is not yet compensated for) this effect plays a role because the agent does not track the path accurately. Therefore in this simulation, the bias is simulated to vary in both the tangential and transversal directions. Also, Gaussian noise is simulated to corrupt the sensors. The variance of the position-dependent noise component is also simulated to vary in both the tangential and transversal directions. See Figure 4-2 for the results of this simulation indicated by *real*. The simulation was run for $N^b \in \{3, 11\}$ in two different experimental set-ups, i.e. different reference path, anchor placement, number of anchors and bias profiles.

Consider the simulations in the left plot of Figure 4-2, where the experimental set-up in Appendix A was used. As the model mismatch in this simulation was increased even further with respect to the *mismatch* simulation, it is no surprise that the simulation for $N^b = 11$ diverged again. The simulation was repeated with $N^b = 3$ which reduced the state dimension in an attempt to increase the observability of the augmented state $\boldsymbol{x}$. However, this had the opposite effect and made the estimator diverge even faster.

Now consider the simulations in the right plot of Figure 4-2, where the experimental set-up in Appendix B was used. For $N^b = 11$ the estimation error initially improves and then starts to diverge again, which is similar to the simulation in the right plot. For $N^b = 3$ we see a different result compared to the simulations in the left plot because the estimator improves up to iteration 45 without diverging afterwards. The difference between the left and right plot are the experimental set-ups used in the simulation. These simulations show that the experimental set-up influences the observability of the augmented state. An observability analysis can give insight into the influence of the reference path and anchor placement on the observability of the augmented state. These insights could then be used to optimize the experimental set-up for increased observability. The observability analysis is left for future work, where Appendix E can function as a starting point.

4-1-3 Experimental Results and Discussion

The SBE is experimentally evaluated with two real experiments in the lab, where the experimental set-up in Appendix A was used. The performance of the proposed framework is evaluated as explained in Appendix F by comparing the performance of the SBE with that of the base-case and motion-capture experiments. See Figure 4-3 for the results of the two experiments, where different numbers of bias parameters N^b are used. In both experiments, the estimation error in the position initially improved with respect to the base-case, but then slowly increased again after iteration 8. It was not possible to run the experiment for more than 20 iterations due to the limited capacity of the flight battery. The two experiments show similar divergent behavior, as was observed in the simulations shown in the left plot of Figure 4-2.

No improvements are observed in the velocity estimate $\boldsymbol{v}$ and the first derivative of the tracking error $\dot{\xi}$, which expressed itself in a heavily oscillatory flight around the reference path. This is ascribed to the fact that the position-dependent noise component in the range measurements is not accounted for in this framework. The variance of the position-dependent noise component is large in some parts of the path as can be observed in Figure 2-2. As a result of underestimating the measurement noise in parts of the path, large biases occur

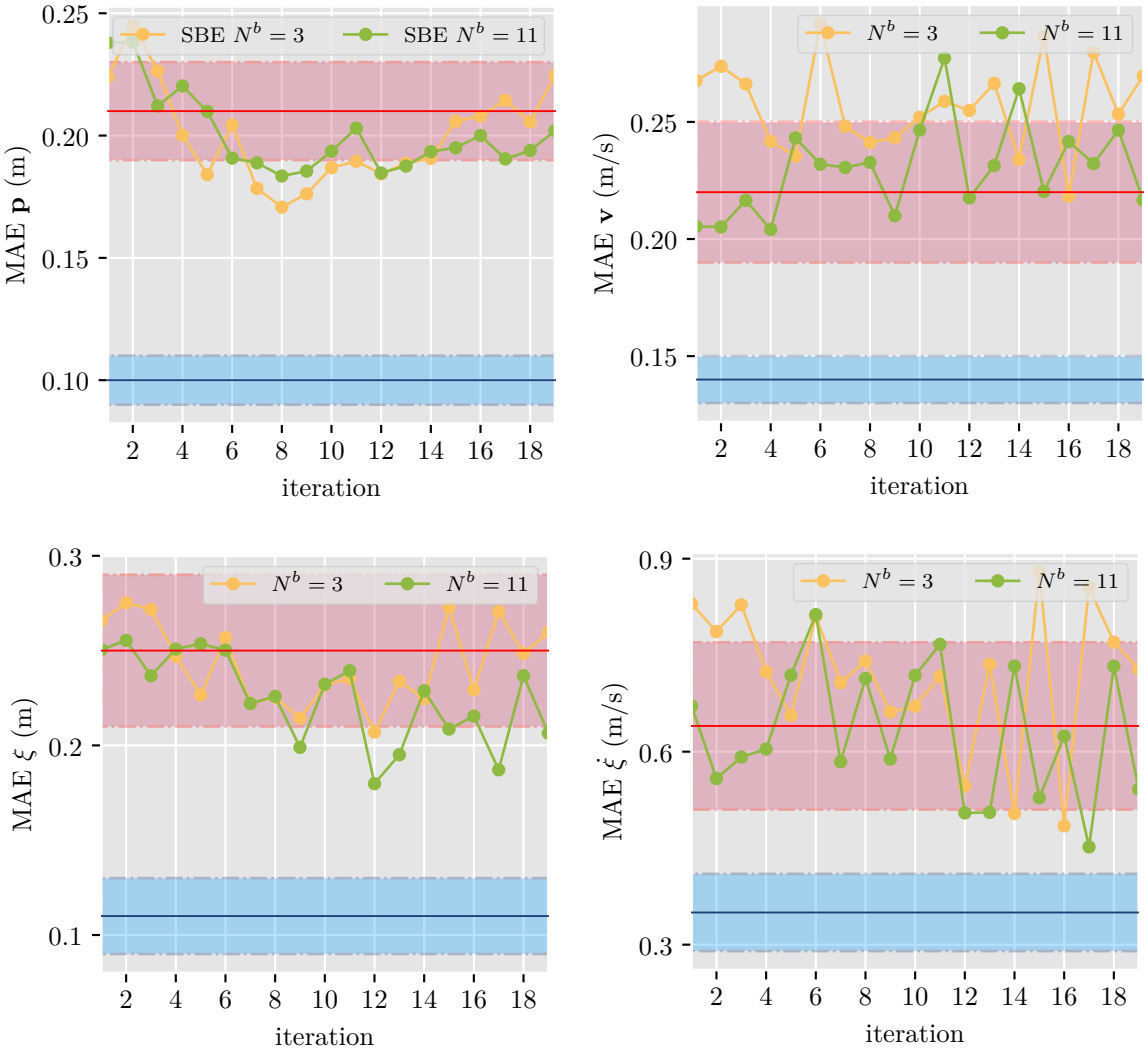

Figure 4-3: The estimator performance (top) and path tracking performance (bottom) are plotted for two real experiments, where the architecture in Figure 4-1 was used. The mean performance (line) with one standard deviation (colored area) of the base-case (red) and motion-capture (blue) experiments are plotted for comparison.

more frequently than accounted for in the measurement noise covariance Σ^{uwb}. This induces oscillations in the agent's state estimation error, which in turn has a similar effect on the path tracking performance.

The SBE was tough to tune. Striking a balance between bias process noise covariance Σ^{bias} and the initial covariance $\tilde{P}_0$ was hard, as the interplay between both tuning parameters is difficult to interpret when performing sequential updates. An iterative framework where the updates are performed only after completing a full iteration is possibly easier to tune. Moreover, an iterative framework allows for non-causal filtering. This motivates research into an iterative framework where the bias parameters are estimated over iterations instead of sequentially.

Chapter 5

Iterative Bias Estimator

In this chapter, the control and estimation architecture in Figure 5-1 is proposed. Every timestep, the motion controller receives an estimate of the state $\boldsymbol{x}$ from the online state estimator, which it uses to determine a control input $\boldsymbol{u}$ that forces the agent's position on the path. The agent's sensor measurements are fused in the online state estimator and logged over one iteration j, after which they are sent to the iterative bias estimator (IBE), with its components encircled by the dashed line in Fig. 5-1. The iterative framework enables the logged measurements to be fused in a non-causal state estimator that estimates the agent's state trajectory $\boldsymbol{x}(t, \boldsymbol{c}_x)$ and the bias trajectories $b^a(t, \boldsymbol{c}_b^a)$ of iteration j. The trajectories are expressed as the weighted sum of a finite set of temporal basis functions with parameters $\boldsymbol{c}_x \in \mathbb{R}^{N^x}, \boldsymbol{c}_b^a \in \mathbb{R}^{N^c}$. The variables N^x and N^c denote the number of parameters used to express the state trajectory and bias trajectory per anchor, respectively. Subsequently, the estimated state trajectory $\boldsymbol{x}(t, \boldsymbol{c}_x)$ is used to infer the range error $\mathcal{B}_{(j)}$ in the logged range measurements. The range errors are combined over iterations in a recursive least squares (RLS) filter, allowing the estimation of the bias' variance, in addition to its mean. The range errors are used instead of directly using the bias trajectories $b^a(t, \boldsymbol{c}_b^a)$ because the bias trajectories are a smoothened representation of the bias which hampers the estimation of the bias' variance. A forgetting factor in the RLS filter allows the framework to deal with non-static environments. The updated bias parameters $\boldsymbol{\theta}^{\bar{b}}_{(j)}, \boldsymbol{\theta}^{\sigma^2}_{(j)}$ are subsequently used by the online state estimator to evaluate the bias of an anchor for a given position $\boldsymbol{p}$ as described in Section 3-1.

It is important to note that the IBE also uses state augmentation, similar to the sequential bias estimator (SBE) in Chapter 4. However, in this framework the agent's state trajectory $\boldsymbol{x}(t, \boldsymbol{c}_x)$ is augmented with the bias parameters $\boldsymbol{c}_b^a$ which are estimated over iterations, as opposed to the SBE where the agent's state $\boldsymbol{x}$ is augmented with the bias parameters $\boldsymbol{\theta}^{\bar{b}}$ which are estimated at every timestep. Nevertheless, the bias parameters $\boldsymbol{\theta}^{\bar{b}}_{(j)}, \boldsymbol{\theta}^{\sigma^2}_{(j)}$, located in the bias parametrization block in Figure 5-1, remain necessary for combining the estimated range errors over iterations, for providing a continuous path-dependent parametrization, and for dealing with non-static environments.

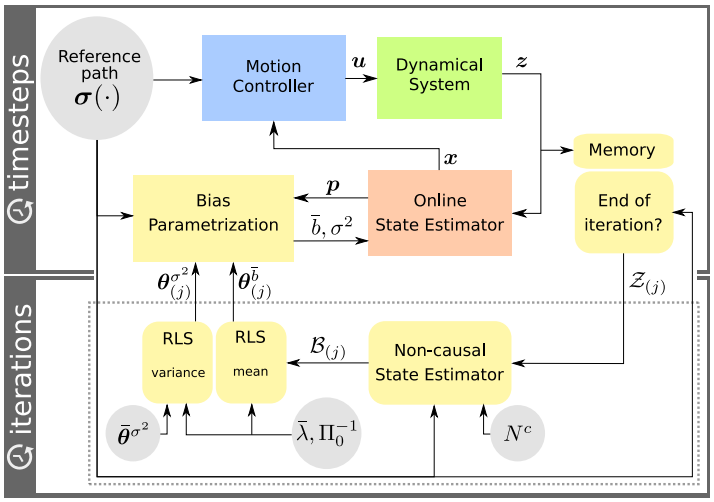

Figure 5-1: The estimation and control architecture for controlling the motion of a dynamical system with the dashed line encircling the components of the IBE. The main tuning parameters of the IBE are contained in the grey circles.

The set of logged measurements in iteration j is defined as the union

$$\mathcal{Z}_{(j)} = \mathcal{Z}^{\text{uwb}}_{(j)} \cup \mathcal{Z}^{\text{acc}}_{(j)} \cup \mathcal{Z}^{\text{gyr}}_{(j)} \cup \mathcal{Z}^{\text{bar}}_{(j)}, \tag{5-1}$$

of the range, acceleration, angular rate, and altitude measurement sets, respectively. For ease of notation, the anchor superscript a is dropped in $\boldsymbol{c}_b^a$, but do note that each anchor has its own bias trajectory. The components of the IBE are specified in the following paragraphs. Then, the implementation details of the framework on the quadcopter platform are provided, after which experimental results are presented.

Non-causal State Estimator For each iteration j, the non-causal estimator estimates the range error set $\mathcal{B}_{(j)}$ defined in (2-5). For this, it requires estimates of the position at every time $t_k, k \in \mathcal{Z}^{\text{uwb}}_{(j)}$. An adapted version of the continuous-time batch optimization framework presented in [8] is used, as follows. The measurement set $\mathcal{Z}_{(j)}$ is used to find the maximum a-posteriori estimate of the state trajectory $\boldsymbol{x}(t, \boldsymbol{c}_x)$ and range bias trajectory $b^a(t, \boldsymbol{c}_b)$. It is assumed that the position trajectory $\boldsymbol{p}(t)$ and the unit quaternion trajectory $\boldsymbol{q}(t)$, describing the rotation $R(t)$ from the inertial to the body frame, can both be inferred from the state trajectory $\boldsymbol{x}(t, \boldsymbol{c}_x)$. An augmented state trajectory is defined as $\tilde{\boldsymbol{x}}(t, \boldsymbol{c}_x, \boldsymbol{c}_b) = (\boldsymbol{x}(t, \boldsymbol{c}_x), b^a(t, \boldsymbol{c}_b))$, such that Bayes' law can be used to write the posterior estimate as

$$p(\tilde{\boldsymbol{x}}(t, \boldsymbol{c}_x, \boldsymbol{c}_b)|\mathcal{Z}_{(j)}) = \frac{p(\tilde{\boldsymbol{x}}(t, \boldsymbol{c}_x, \boldsymbol{c}_b))}{p(\mathcal{Z}_{(j)})} p(\mathcal{Z}_{(j)}|\tilde{\boldsymbol{x}}(t, \boldsymbol{c}_x, \boldsymbol{c}_b)). \tag{5-2}$$

Assuming that the sensor measurements are conditionally independent given $\tilde{\boldsymbol{x}}(t, \boldsymbol{c}_x, \boldsymbol{c}_b)$, the posterior is rewritten as

$$p(\tilde{\boldsymbol{x}}(t, \boldsymbol{c}_x, \boldsymbol{c}_b)|\mathcal{Z}_{(j)}) = \frac{p(\tilde{\boldsymbol{x}}(t, \boldsymbol{c}_x, \boldsymbol{c}_b))}{p(\mathcal{Z}_{(j)})} \prod_{m \in \mathcal{Z}_{(j)}} p(z_m|\tilde{\boldsymbol{x}}(t_m, \boldsymbol{c}_x, \boldsymbol{c}_b)). \tag{5-3}$$

The measurement likelihood for an arbitrary sensor measurement m with a Gaussian distribution is given by

$$p(z_m|\tilde{\boldsymbol{x}}(t_m, \boldsymbol{c}_x, \boldsymbol{c}_b)) = \mathcal{N}(h_m(\tilde{\boldsymbol{x}}(t_m, \boldsymbol{c}_x, \boldsymbol{c}_b)), \Sigma_m), \tag{5-4}$$

where the measurement model $h_m(\tilde{\boldsymbol{x}}(t_m, \boldsymbol{c}_x, \boldsymbol{c}_b))$ denotes the mean of z_m, and Σ_m denotes the noise covariance matrix. The model

$$h_k(\tilde{\boldsymbol{x}}(t_k, \boldsymbol{c}_x, \boldsymbol{c}_b)) = \parallel \boldsymbol{p}^{a_k} - \boldsymbol{p}(\tilde{\boldsymbol{x}}(t_k, \boldsymbol{c}_x, \boldsymbol{c}_b)) \parallel + b^{a_k}(\tilde{\boldsymbol{x}}(t_k, \boldsymbol{c}_x, \boldsymbol{c}_b)) \tag{5-5}$$

is used for the range measurements $k \in \mathcal{Z}_{(j)}^{\text{uwb}}$. Slow drifting behavior in the barometer can occur, and causes an offset o^{bar} in the altitude measurements that is approximately constant over one iteration as can be seen in Appendix C. Therefore, a constant offset o^{bar} is estimated each iteration by assuming the barometer measurement model

$$h_l(\tilde{\boldsymbol{x}}(t_l, \boldsymbol{c}_x, \boldsymbol{c}_b), o^{\text{bar}}) = p_z(\tilde{\boldsymbol{x}}(t_l, \boldsymbol{c}_x, \boldsymbol{c}_b)) + o^{\text{bar}}. \tag{5-6}$$

for the altitude measurements $l \in \mathcal{Z}_{(j)}^{\text{bar}}$. By differentiating the position trajectory $\boldsymbol{p}(\tilde{\boldsymbol{x}}(t, \boldsymbol{c}_x, \boldsymbol{c}_b))$ twice with respect to time, the acceleration trajectory $\ddot{\boldsymbol{p}}(\tilde{\boldsymbol{x}}(t, \boldsymbol{c}_x, \boldsymbol{c}_b))$ is obtained. The measurement models for the inertial measurement unit (IMU) are based on the models in [12]. The acceleration trajectory together with the rotation matrix $R(\boldsymbol{q}(\tilde{\boldsymbol{x}}(t, \boldsymbol{c}_x, \boldsymbol{c}_b)))$ are used to model the acceleration measurements $n \in \mathcal{Z}_{(j)}^{\text{acc}}$ as

$$h_n(\tilde{\boldsymbol{x}}(t_n, \boldsymbol{c}_x, \boldsymbol{c}_b)) = R(\boldsymbol{q}(\tilde{\boldsymbol{x}}(t_n, \boldsymbol{c}_x, \boldsymbol{c}_b)))(\ddot{\boldsymbol{p}}(\tilde{\boldsymbol{x}}(t_n, \boldsymbol{c}_x, \boldsymbol{c}_b)) - \boldsymbol{g}), \tag{5-7}$$

where $\boldsymbol{g}$ is the gravitational acceleration. Likewise, the first time derivative of the unit quaternion $\dot{\boldsymbol{q}}(\tilde{\boldsymbol{x}}(t, \boldsymbol{c}_x, \boldsymbol{c}_b))$ can be obtained by differentiating the continuous-time trajectory $\boldsymbol{q}(\tilde{\boldsymbol{x}}(t, \boldsymbol{c}_x, \boldsymbol{c}_b))$. By defining q_w and $\tilde{\boldsymbol{q}}$ to be the scalar and vector part of the unit quaternion $\boldsymbol{q}$, respectively, the model for the angular rate measurements $w \in \mathcal{Z}_{(j)}^{\text{gyr}}$ can be defined as

$$\begin{aligned} h_w(\tilde{\boldsymbol{x}}(t_w, \boldsymbol{c}_x, \boldsymbol{c}_b)) = 2\big(q_w(\tilde{\boldsymbol{x}}(t_w, \boldsymbol{c}_x, \boldsymbol{c}_b))\dot{\tilde{\boldsymbol{q}}}(\tilde{\boldsymbol{x}}(t_w, \boldsymbol{c}_x, \boldsymbol{c}_b)) - \dot{q}_w(\tilde{\boldsymbol{x}}(t_w, \boldsymbol{c}_x, \boldsymbol{c}_b))\tilde{\boldsymbol{q}}(\tilde{\boldsymbol{x}}(t_w, \boldsymbol{c}_x, \boldsymbol{c}_b)) \\ + \tilde{\boldsymbol{q}}(\tilde{\boldsymbol{x}}(t_w, \boldsymbol{c}_x, \boldsymbol{c}_b)) \times \dot{\tilde{\boldsymbol{q}}}(\tilde{\boldsymbol{x}}(t_w, \boldsymbol{c}_x, \boldsymbol{c}_b))\big). \end{aligned} \tag{5-8}$$

In case the IMU is biased, additive bias trajectories can be added to the models in (5-7) and (5-8) as described in [12]. The maximum a-posteriori estimates $({}^*\boldsymbol{c}_x, {}^*\boldsymbol{c}_b, {}^*o^{\text{bar}})$ are found by minimizing the negative logarithm of the posterior (5-3),

$$\min_{\boldsymbol{c}_x, \boldsymbol{c}_b, o^{\text{bar}}} \sum_{m \in \mathcal{Z}_{(j)}} -\log(p(z_m|\tilde{\boldsymbol{x}}(t_m, \boldsymbol{c}_x, \boldsymbol{c}_b), o^{\text{bar}})), \tag{5-9}$$

where the term $p(\mathcal{Z}_{(j)})$ is omitted, as it does not influence the optimized solution, and the prior $p(\tilde{\boldsymbol{x}}(t, \boldsymbol{c}_x, \boldsymbol{c}_b))$ is omitted because the ratio of measurements to optimization variables is large. This reduces (5-9) to a maximum-likelihood problem [8]. A local optimization is performed, where the estimates of the online state estimator serve as the initial guess for $\boldsymbol{c}_x$ in the optimization. The bias parameters $\boldsymbol{c}_b$ are initialized to zero. The estimated barometer offset of the previous cycle is used to initialize o^{bar}. With the resulting position trajectory $\boldsymbol{p}(t, {}^*\boldsymbol{c}_x)$, an estimate of the range error set $\mathcal{B}_{(j)}$ is calculated with (2-3), (2-4), and (2-5). The barometer offset ${}^*o^{\text{bar}}$ is provided to the online state estimator, such that the drift is compensated for in the altitude measurements $z_l, l \in \mathcal{Z}^{\text{bar}}$ before being fused in the online state estimator.

Recursive Least Squares Filter The bias parameters $\boldsymbol{\theta}^{\bar{b}}_{(j)}, \boldsymbol{\theta}^{\sigma^2}_{(j)}$ are estimated recursively by combining the range error estimates $\mathcal{B}_{(j)}$ with the sets of range error estimates from all previous iterations, i.e. $\mathcal{B}_{(1..j)} = \cup_{i=1}^{j} \mathcal{B}_{(i)}$, yielding the two bias parametrizations introduced in (2-7) and defined as

$$\bar{b}(\eta_k, \boldsymbol{\theta}^{\bar{b}}_{(j)}) = \boldsymbol{h}^{\mathsf{T}}(\eta_k)\boldsymbol{\theta}^{\bar{b}}_{(j)} \tag{5-10}$$

$$\sigma^2(\eta_k, \boldsymbol{\theta}^{\sigma^2}_{(j)}) = \boldsymbol{h}^{\mathsf{T}}(\eta_k)\boldsymbol{\theta}^{\sigma^2}_{(j)}. \tag{5-11}$$

The vector $\boldsymbol{h}^{\mathsf{T}}$ is constructed according to Section 2-2-2 for the chosen bias parametrization. To solve for the mean bias parameters $\boldsymbol{\theta}^{\bar{b}}_{(j)}$, a regressor matrix $H \in \mathbb{R}^{N^b \times |\mathcal{B}_{(1..j)}|}$ and range error vector $\boldsymbol{e} \in \mathbb{R}^{|\mathcal{B}_{(1..j)}|}$ are constructed with the set $\mathcal{B}_{(1..j)}$ of all range error, path variable pairs up to iteration j, i.e.,

$$H(\mathcal{B}_{(1..j)}) = [\boldsymbol{h}(\eta_1), \boldsymbol{h}(\eta_2), ..., \boldsymbol{h}(\eta_{|\mathcal{B}_{(1..j)}|})] \tag{5-12}$$

$$\boldsymbol{e}(\mathcal{B}_{(1..j)}) = [e_1, e_2, ..., e_{|\mathcal{B}_{(1..j)}|}]^{\mathsf{T}}. \tag{5-13}$$

Subsequently, the problem of finding the mean bias parameters $\boldsymbol{\theta}^{\bar{b}}_{(j)}$ can be formulated as a least squares problem,

$$\min_{\boldsymbol{\theta}^{\bar{b}}_{(j)}} [(\boldsymbol{\theta}^{\bar{b}}_{(j)} - \bar{\boldsymbol{\theta}}^{\bar{b}})^{\mathsf{T}} \Pi_0^{-1} (\boldsymbol{\theta}^{\bar{b}}_{(j)} - \bar{\boldsymbol{\theta}}^{\bar{b}}) + \| \boldsymbol{e} - H^{\mathsf{T}} \boldsymbol{\theta}^{\bar{b}}_{(j)} \|^2], \tag{5-14}$$

where the initial guess $\bar{\boldsymbol{\theta}}^{\bar{b}}$ and the weighting matrix Π_0^{-1} represent the prior knowledge. Each iteration, $\boldsymbol{\theta}^{\bar{b}}_{(j)}$ is estimated by solving (5-14) recursively with the regularized RLS algorithm described in Section 21.4 of [20]. A forgetting factor $\bar{\lambda}$ is included in the algorithm, so that the framework is adaptive to non-static environments. The regularized RLS iteration-update with exponential forgetting is given as

$$\Phi_{(j)} = \bar{\lambda}\Phi_{(j-1)} + H_{(j)}H^{\mathsf{T}}_{(j)} \qquad \boldsymbol{S}_{(j)} = \bar{\lambda}\boldsymbol{S}_{(j-1)} + H_{(j)}(\boldsymbol{e}_{(j)} - H^{\mathsf{T}}_{(j)}\boldsymbol{\theta}^{\bar{b}}_{(j)}), \tag{5-15}$$

with initial conditions $\boldsymbol{S}_{(0)} = \boldsymbol{0}$ and $\Phi_{(0)} = \Pi_0^{-1}$. Error vector $\boldsymbol{e}_{(j)}$ and regressor matrix $H_{(j)}$ are constructed with (5-13) and (5-12), where $\mathcal{B}_{(j)}$ is used instead of $\mathcal{B}_{(1..j)}$. Then, the mean bias parameters $\boldsymbol{\theta}^{\bar{b}}_{(j)}$ can be obtained with

$$\boldsymbol{\theta}^{\bar{b}}_{(j)} = \boldsymbol{\Phi}^{-1}_{(j)}\boldsymbol{S}_{(j)} + \bar{\boldsymbol{\theta}}^{\bar{b}} \tag{5-16}$$

Similarly, the variance bias parameters $\boldsymbol{\theta}^{\sigma^2}_{(j)}$ are estimated by recursively solving

$$\min_{\boldsymbol{\theta}^{\sigma^2}_{(j)}} [(\boldsymbol{\theta}^{\sigma^2}_{(j)} - \bar{\boldsymbol{\theta}}^{\sigma^2})^{\mathsf{T}} \Pi_0^{-1} (\boldsymbol{\theta}^{\sigma^2}_{(j)} - \bar{\boldsymbol{\theta}}^{\sigma^2}) + \| \boldsymbol{r} - H^{\mathsf{T}} \boldsymbol{\theta}^{\sigma^2}_{(j)} \|^2], \tag{5-17}$$

where $\bar{\boldsymbol{\theta}}^{\sigma^2}$ is the initial guess for the variance. In practice, $\bar{\boldsymbol{\theta}}^{\sigma^2}$ is set to a large value, while $\bar{\boldsymbol{\theta}}^{\bar{b}}$ is initialized to zero, as the shape of the bias profile is unknown. Vector $\boldsymbol{r}$ denotes the squared residual between the range error data and mean bias fit. The squared residual is a common metric for estimating variances, as described in [4], and is defined as

$$\boldsymbol{r} = (\boldsymbol{e} - H^{\mathsf{T}}\boldsymbol{\theta}^{\bar{b}}_{(j)}) \odot (\boldsymbol{e} - H^{\mathsf{T}}\boldsymbol{\theta}^{\bar{b}}_{(j)}), \tag{5-18}$$

where $\odot$ is the element-wise multiplication operator. The estimated parameters $\boldsymbol{\theta}^{\bar{b}}_{(j)}, \boldsymbol{\theta}^{\sigma^2}_{(j)}$ are provided to the bias parametrization block in Figure 5-1, such that the online state estimator can evaluate the mean and variance bias parametrizations in (5-10), and (5-11) for a given position $\boldsymbol{p}$, using (2-3), and (2-4).

5-1 Evaluation on a Quadcopter Platform

The IBE is evaluated on a quadcopter platform. First, the implementation details are discussed, after which simulation and experimental results are presented.

5-1-1 Implementation

The motion controller and online state estimator are run on a Snapdragon flight board, located onboard the quadcopter. The IBE is run over iterations in a separate process, offboard on a laptop with an i7-Intel processor. The onboard and offboard processes communicate with a TCP client/server set-up.

The implementation of the non-causal state estimator is similar to the implementation in [12]. The state trajectory $\boldsymbol{x}(t, \boldsymbol{c}_x)$ is assumed to consist of a position trajectory $\boldsymbol{p}(t, \boldsymbol{c}_x)$ and unit quaternion trajectory $\boldsymbol{q}(t, \boldsymbol{c}_x)$ resulting in 7 state trajectories. Each trajectory is parametrized by 33 uniform cubic B-splines, resulting in a total of $N^x = 7 * 33 = 231$ parameters. For an iteration j with a duration of 15 seconds, this translates to a knot spacing for the splines of roughly 500 ms. The bias trajectory $b^a(t, \boldsymbol{c}_b)$ is also parametrized by a set of uniform cubic B-splines for which the number of parameters per anchor N^c is specified in Section 5-1-2 and Section 5-1-3 for different experiments. The IMU is assumed to be unbiased because it showed no drifting behavior after calibrating prior to flying. The optimization routine to solve (5-9) is implemented in Tensorflow [1], such that the gradient can be computed efficiently with Tensorflow's automatic differentiation. Tensorflow also permits easy use of the computational power of a GPU. However, this functionality was not used in this thesis.

The RLS filter has a forgetting factor $\bar{\lambda} = 0.9$, which means that the estimated parameters $\boldsymbol{\theta}^{\bar{b}}_{(j)}, \boldsymbol{\theta}^{\sigma^2}_{(j)}$ depend on roughly the last 10 iterations. See Appendix F-1 for an experiment which demonstrates that the influences of an erroneous update in the RLS filter diminishes after 10 iterations. A large uncertainty in the bias is encoded by setting each element in the initial variance $\bar{\boldsymbol{\theta}}^{\sigma^2}$ to a large value of 0.15^2. The weighting matrix is set to be $\Pi_0^{-1} = HH^{\mathsf{T}}$, where H is constructed with (5-12) using N^{sub} uniformly sampled path variables η along the path. Variable N^{sub} denotes the expected number of range measurements collected during one iteration. If the number of anchors is denoted by N^a, then for an iteration j with a duration of 15 seconds, roughly $|\mathcal{Z}^{\text{uwb}}_{(j)}| \approx 3000$ range measurements are logged, resulting in $N^{\text{sub}} \approx |\mathcal{Z}^{\text{uwb}}_{(j)}|/N^a \approx 429$ measurements per anchor. Two piece-wise linear parametrization with $N^b = 50$, introduced in Section 2-2-2, are used to parametrize the mean and variance of the bias. The reference path is a closed path, so the parametrization in (2-9), (2-10), and (2-11) is adjusted accordingly.

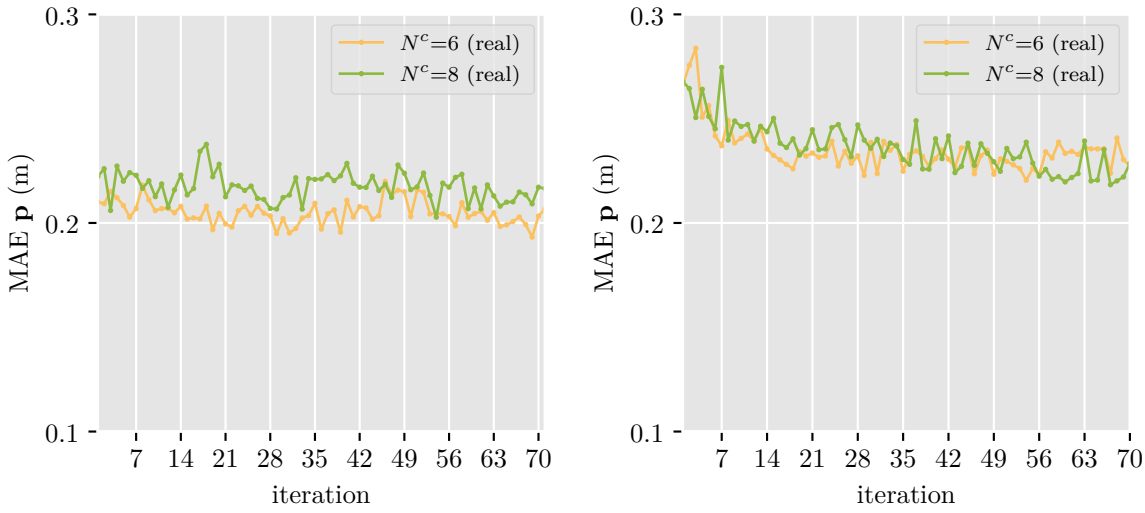

Figure 5-2: The estimation error in the position over iterations of the IBE, where the architecture in Figure 5-1 was used. Either the experimental set-up in Appendix A (left), or the experimental set-up in Appendix B (right) was used.

5-1-2 Simulation Results and Discussion

In the simulations, the quadcopter is commanded to fly the reference path accurately using the control and estimation architecture in Figure 5-1. All sensors (IMU, anchors, barometer), the dynamics of the quadcopter, and the cascaded controller structure described in [11] are simulated in a single process. See Figure 5-2 for the results of the simulations. Either the experimental set-up in Appendix A, or the fictitious experimental set-up in Appendix B was used.

Real The *real* simulation simulates measurement noise and assumes the bias to vary in both tangential and transversal direction with respect to the path. It is the simulation that most accurately simulates the real experiments in the lab. For a detailed description of this simulation the reader is referred to the *real* paragraph in Section 4-1. For the experimental set-up in Appendix A no significant improvements in the position error is observed in the left plot of Figure 5-2 for both $N^c = 6$ and $N^c = 8$. On the other hand, the results in the right plot of Figure 5-2 for the experimental set-up in Appendix B do show improvements. The difference in performance between the left and right plot in Figure 5-2 must be caused by the experimental set-up (i.e. the reference path, anchor placement, number of anchors and bias profiles), as it is the only difference between the two experiments. The simulation results for the SBE in Figure 4-2 showed a similar difference, so it is concluded that the experimental set-up significantly influences the effectiveness of strategies that use state augmentation. As suggested in Chapter 4, an observability analysis could provide insight into the effect of the experimental set-up on the estimator performance. The observability analysis is left for future work, where Appendix E can function as a starting point.

The right plot of Figure 4-2 and Figure 5-2 both used the exact same experimental set-up, but the SBE diverged, while the IBE did not. The interaction between the different components (motion controller, online estimator, quadcopter dynamics) is very convoluted. Therefore, it

is hard to make a definitive statement that explains the observed difference. It is believed that the IBE did not diverge, because of the iterative framework that allowed for non-causal filtering and the estimation of the bias' variance, in addition to its mean. The non-causal filtering should be able to estimate the bias parameters more accurately, and the incorporation of the bias' variance reduced model mismatch as the position-dependent noise component was accounted for.

5-1-3 Experimental Results and Discussion

See Figure 5-3 for the experimental results of two experiments with a varying number of bias parameters N^c that used the experimental set-up in Appendix A. In both of these experiments, the motion controller uses the estimate of the agent's state provided for by the online state estimator. The performance is evaluated as explained in Appendix F by comparing the performance of the IBE with that of the base-case and motion-capture experiments.

Concerning $\dot{\xi}$, a similar performance level with respect to the motion-capture experiment is obtained. The improvements in $\dot{\xi}$ are ascribed to the estimation of the position-dependent noise component, in addition to the mean bias and result in reduced oscillations during flight. While the reduction of the estimation error in the position $\boldsymbol{p}$ is minimal, there is a significant reduction in the path tracking error ξ. It is believed that this reduction is also obtained because of the inclusion of the estimated bias' variance σ^2 in (3-4) to account for the position-dependent noise component in the online state estimator.

The main tuning parameters of the IBE are contained in the grey circles in Figure 5-1. Each tuning parameter has a clear physical interpretation, which facilitated the tuning process. This makes the IBE more intuitive to tune compared to the SBE in Chapter 4, where it was hard to set the process noise and initial covariance matrix that corresponded to the bias parameters.

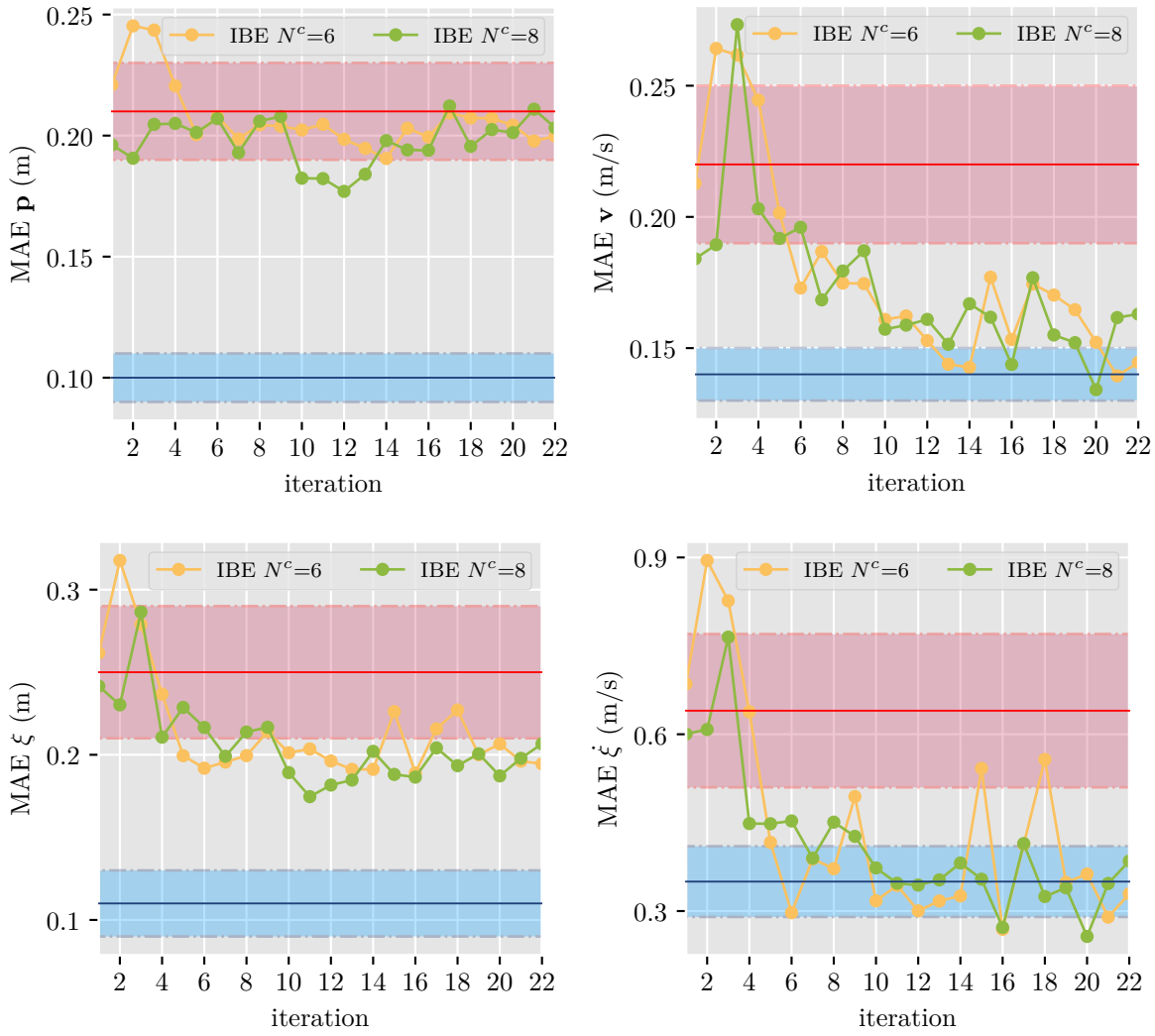

Figure 5-3: The estimator performance (top) and path tracking performance (bottom) of the IBE (IBE) are plotted for four runs, where the architecture in Figure 5-1 was used. The mean performance (line) with one standard deviation (colored area) of the base-case (red) and motion-capture (blue) experiments are plotted for comparison.

Chapter 6

Reliable Iterative Bias Estimator

In this chapter, the control and estimation architecture in Figure 6-1 is proposed. Every timestep, the motion controller receives an estimate of the state $\boldsymbol{x}$ from the online state estimator, which it uses to determine a control input $\boldsymbol{u}$ that forces the agent's position on the path. The agent's sensor measurements are fused in the online state estimator and logged over one iteration j, after which they are sent to the reliable iterative bias estimator (RIBE), with its components encircled by the dashed line in Figure 6-1. The RIBE relies on a classifier to distinguish reliable (i.e. unbiased) from unreliable range measurements. The iterative framework enables the reliable measurements to be fused in a non-causal state estimator whose state trajectory estimate is used to infer the range error in both the reliable, and unreliable range measurements. The range errors are combined over iterations in a recursive least squares (RLS) filter, allowing the estimation of the bias' variance, in addition to its mean. A forgetting factor in the RLS filter allows the framework to deal with non-static environments. The updated bias parameters $\boldsymbol{\theta}^{\bar{b}}_{(j)}, \boldsymbol{\theta}^{\sigma^2}_{(j)}$ are subsequently used by the online state estimator to evaluate the bias of an anchor for a given position $\boldsymbol{p}$ as described in Section 3-1. The set of logged measurements in iteration j is defined as the union

$$\mathcal{Z}_{(j)} = \mathcal{Z}^{\text{uwb}}_{(j)} \cup \mathcal{Z}^{\text{acc}}_{(j)} \cup \mathcal{Z}^{\text{gyr}}_{(j)} \cup \mathcal{Z}^{\text{bar}}_{(j)}, \tag{6-1}$$

of the range, acceleration, angular rate, and altitude measurement set, respectively. The components of the RIBE are specified in the following paragraph. Then, the implementation details of the framework on the quadcopter platform are provided, after which experimental results are presented.

Classifier The logged measurements are sent to a classifier. The classifier must partition range measurements $\mathcal{Z}^{\text{uwb}}_{(j)}$ into a reliable partition $\mathcal{Z}^{\text{uwb,rel}}_{(j)}$ and an unreliable partition $\mathcal{Z}^{\text{uwb,unrel}}_{(j)}$. The classification algorithm is an adapted version of the Residual Weighting algorithm presented in [5], which is based on a least squares (LS) algorithm. The algorithm herein differs in that its task is to classify reliable measurements, instead of estimating the position. Furthermore, the LS algorithm uses the altitude measurements in addition to the range measurements.

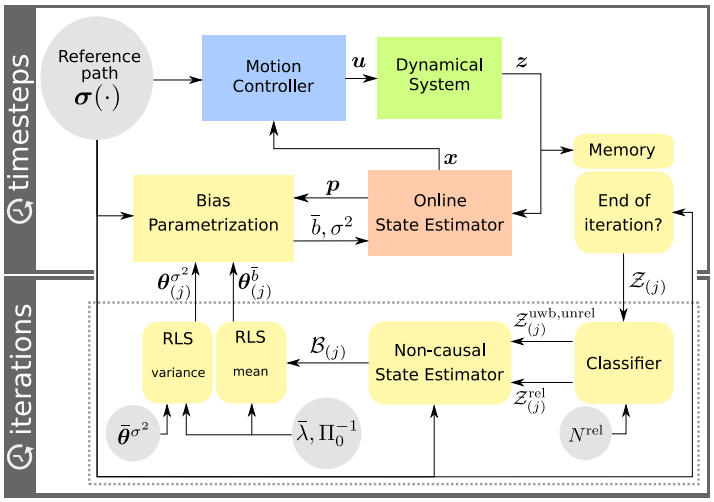

Figure 6-1: The estimation and control architecture for controlling the motion of a dynamical system with the dashed line encircling the components of the RIBE. The main tuning parameters of the RIBE are contained in the grey circles.

When the measurements $\mathcal{Z}_{(j)}$ are received, the range measurements are first partitioned into N^{sub} subsequences, i.e.

$$\mathcal{Z}^{\text{uwb}}_{(j)} = \{\mathcal{Z}^{\text{uwb}}_{(j),1}, \mathcal{Z}^{\text{uwb}}_{(j),2}, \dots, \mathcal{Z}^{\text{uwb}}_{(j),N^{\text{sub}}}\}. \tag{6-2}$$

Each subsequence $\mathcal{Z}^{\text{uwb}}_{(j),n} \in \mathcal{Z}^{\text{uwb}}_{(j)}$ with $n \in \{1, 2, ..., N^{\text{sub}}\}$ contains consecutive range measurements obtained with different anchors. Hence, if N^a is the number of anchors from which sequential range measurements are obtained, such a subsequence is of size $|\mathcal{Z}^{\text{uwb}}_{(j),n}| = N^a$. The classifier labels N^{rel} range measurements per subsequence $\mathcal{Z}^{\text{uwb}}_{(j),n}$ as reliable, where N^{rel} is set to the expected number of anchors with line-of-sight (LOS) at any given time. Reliable range measurements are consistent with each other, which is reflected in a low residual of a LS position estimate given reliable range measurements. Therefore, a LS position estimate is calculated for different range measurement combinations $\mathcal{C}^i_{(j),n} \subset \mathcal{Z}^{\text{uwb}}_{(j),n}$, $|\mathcal{C}^i_{(j),n}| = N^{\text{rel}}$. These combination are obtained by taking all $|\mathcal{Z}^{\text{uwb}}_{(j),n}|$ choose N^{rel} different combinations, resulting in N^{c} combinations per subsequence. The combination $\mathcal{C}^i_{(j),n}$ that results in the lowest LS position estimate residual $r^i_{(j),n}$ is classified as reliable, i.e.

$$\mathcal{Z}^{\text{uwb,rel}}_{(j),n} = \mathcal{C}^i_{(j),n} \text{ s.t. } i = \underset{i \in 1,2,\dots,N^{\text{c}}}{\arg\min}\ r^i_{(j),n}. \tag{6-3}$$

The altitude measurements $l \in \mathcal{Z}^{\text{bar}}_{(j),n}$, measured within the time spanned by the range measurement timestamps of $\mathcal{Z}^{\text{uwb}}_{(j),n}$, are additionally fused into the LS estimate. The residual of such a LS estimate is given by

$$r^i_{(j),n} = \min_{\boldsymbol{p} \in \mathbb{R}^3} \left(r^{i,\text{uwb}}_{(j),n}(\boldsymbol{p}) + r^{\text{bar}}_{(j),n}(p_{\text{z}}) \right), \tag{6-4}$$

where p_{z} is the z-coordinate of the position $\boldsymbol{p}$, and

$$r^{i,\text{uwb}}_{(j),n}(\boldsymbol{p}) = \frac{1}{\Sigma^{\text{uwb}}} \sum_{k \in \mathcal{C}^i_{(j),n}} (z_k - \| \boldsymbol{p}^{a_k} - \boldsymbol{p} \|)^2, \qquad r^{\text{bar}}_{(j),n}(p_z) = \frac{1}{\Sigma^{\text{bar}}} \sum_{l \in \mathcal{Z}^{\text{bar}}_{(j),n}} (z_l - p_z)^2. \tag{6-5}$$

where Σ^{uwb} and Σ^{bar} are the corresponding measurement noise variances. Then, the set of reliable range measurements $\mathcal{Z}_{(j)}^{\text{uwb,rel}}$ over a complete iteration is given as the union of the combinations with the lowest residuals, and the unreliable set $\mathcal{Z}_{(j)}^{\text{uwb,unrel}}$ as the complement to $\mathcal{Z}_{(j)}^{\text{uwb,rel}}$, i.e.

$$\mathcal{Z}_{(j)}^{\text{uwb,rel}} = \{\mathcal{Z}_{(j),1}^{\text{uwb,rel}}, \mathcal{Z}_{(j),2}^{\text{uwb,rel}}, \ldots, \mathcal{Z}_{(j),N^{\text{sub}}}^{\text{uwb,rel}}\}, \qquad \mathcal{Z}_{(j)}^{\text{uwb,unrel}} = \mathcal{Z}_{(j)}^{\text{uwb}} \setminus \mathcal{Z}_{(j)}^{\text{uwb,rel}}. \tag{6-6}$$

Non-causal State Estimator For each iteration j, the non-causal estimator estimates the range error set $\mathcal{B}_{(j)}$ defined in (2-5). For this, it requires estimates of the position at every time $t_k, k \in \mathcal{Z}_{(j)}^{\text{uwb}}$. The non-causal state estimator is similar to the one used in the iterative bias estimator (IBE). Only the differences are highlighted between the non-causal estimator implemented in this framework and the one presented in Chapter 5 to avoid repeating much of the derivations. The non-causal estimator in this framework uses the reliable set of measurements

$$\mathcal{Z}_{(j)}^{\text{rel}} = \mathcal{Z}_{(j)}^{\text{uwb,rel}} \cup \mathcal{Z}_{(j)}^{\text{acc}} \cup \mathcal{Z}_{(j)}^{\text{gyr}} \cup \mathcal{Z}_{(j)}^{\text{bar}}, \tag{6-7}$$

to find the maximum a-posteriori estimate of the state trajectory $\boldsymbol{x}(t, \boldsymbol{c}_x)$, whereas in Chapter 5 all measurements $\mathcal{Z}_{(j)}$ are used to find the maximum a-posteriori estimate of the augmented state $\tilde{\boldsymbol{x}}(t, \boldsymbol{c}_x, \boldsymbol{c}_b)$. Only reliable range measurements are used, so the bias trajectory $b^a(t, \boldsymbol{c}_b)$ is omitted in this non-causal estimator. Nevertheless, small differences in the antennas cause a constant offset o_a^{uwb}, slightly different for each anchor. To compensate for o_a^{uwb}, the model

$$h_k(\boldsymbol{x}(t_k, \boldsymbol{c}_x), o_{a_k}^{\text{uwb}}) = \| \boldsymbol{p}^{a_k} - \boldsymbol{p}(\boldsymbol{x}(t_k, \boldsymbol{c}_x)) \| + o_{a_k}^{\text{uwb}} \tag{6-8}$$

is used for the reliable range measurements $k \in \mathcal{Z}_{(j)}^{\text{uwb,rel}}$. By replacing $\tilde{\boldsymbol{x}}(t, \boldsymbol{c}_x, \boldsymbol{c}_b)$ with $\boldsymbol{x}(t, \boldsymbol{c}_x)$ in (5-6), (5-7), and (5-8) the measurement models for the altitude, acceleration, and angular rate measurements are obtained, respectively. Then by following the same derivations as in Chapter 5, the maximum a-posteriori estimates $({}^*\boldsymbol{c}_x, {}^*o_a^{\text{uwb}}, {}^*o^{\text{bar}})$ are found by minimizing

$$\min_{\boldsymbol{c}_x, o_a^{\text{uwb}}, o^{\text{bar}}} \sum_{m \in \mathcal{Z}_{(j)}^{\text{rel}}} -\log(p(z_m | \boldsymbol{x}(t, \boldsymbol{c}_x), o_a^{\text{uwb}}, o^{\text{bar}})). \tag{6-9}$$

A local optimization is performed, where the estimates of the online state estimator serve as the initial guess for $\boldsymbol{c}_x$ in the optimization. The estimated barometer offset of the previous cycle is used to initialize o^{bar}, while the initial guess for o_a^{uwb} is set to zero. With the resulting position trajectory $\boldsymbol{p}(t, {}^*\boldsymbol{c}_x)$, an estimate of the range error set $\mathcal{B}_{(j)}$ is calculated with (2-3), (2-4), and (2-5). It is important to note here, that the range error is calculated for all range measurements, i.e. for measurements classified as reliable and for measurements classified as unreliable. The barometer offset ${}^*o^{\text{bar}}$ is provided to the classifier and online state estimator, such that the drift is compensated for in the altitude measurements $z_l, l \in \mathcal{Z}^{\text{bar}}$ before being incorporated in (6-5) and before being fused in the online state estimator.

Recursive Least Squares Filter The RLS filter is the same filter used in the IBE. Therefore, the reader is referred to Chapter 5 for a detailed description.

6-1 Evaluation on a Quadcopter Platform

The RIBE is evaluated on a quadcopter platform. First, the implementation details are discussed, after which simulation and experimental results are presented.

6-1-1 Implementation

The motion controller and online state estimator are run on a Snapdragon flight board, located onboard the quadcopter. The RIBE is run over iterations in a separate process, offboard on a laptop with an i7-Intel processor. The onboard and offboard processes communicate with a TCP client/server set-up.

It is assumed that the quadcopter has LOS with 5 out of 7 anchors at any given time. Therefore, N^{rel} is set to 5, and $N^c = 21$ for $|\mathcal{Z}^{\text{uwb}}_{(j),n}| = 7$. For an iteration j with a duration of 15 seconds, roughly $|\mathcal{Z}^{\text{uwb}}_{(j)}| \approx 3000$ range measurements are logged, resulting in $N^{\text{sub}} \approx |\mathcal{Z}^{\text{uwb}}_{(j)}|/N^a \approx 429$ partitions, as defined in (6-2). For every partition, (6-4) must be evaluated for all $N^c = 21$ combinations, resulting in a total of $286 * 21 = 9000$ computations of (6-4). All computations are independent from one another, so that the computations can be parallelized.

The implementation of the non-causal state estimator is similar to the implementation in [12]. The state trajectory $\boldsymbol{x}(t, \boldsymbol{c}_x)$ is assumed to consist of a position trajectory $\boldsymbol{p}(t, \boldsymbol{c}_x)$ and unit quaternion trajectory $\boldsymbol{q}(t, \boldsymbol{c}_x)$ resulting in 7 trajectories. Each trajectory is parametrized by 33 uniform cubic B-splines, resulting in a total of $N^x = 7 * 33 = 231$ parameters. For an iteration j with a duration of 15 seconds, this translates to a knot spacing for the splines of roughly 500 ms. The inertial measurement unit (IMU) is assumed to be unbiased because it showed no drifting behavior after calibrating prior to flying. The optimization routine to solve (6-9) is implemented in Tensorflow [1].

The RLS filter is configured similar to the filter used in the IBE which was discussed in Chapter 5. Therefore, the reader is referred to Section 5-1 for the implementation details of the RLS filter.

6-1-2 Simulation Results and Discussion

In the simulations, the quadcopter is commanded to fly the reference path accurately using the control and estimation architecture in Figure 6-1. All sensors (IMU, anchors, barometer), the dynamics of the quadcopter, and the cascaded controller structure described in [11] are simulated in a single process. See Figure 6-2 for the results of the simulations. Either the experimental set-up in Appendix A, or the fictitious experimental set-up in Appendix B was used.

Real The *real* simulation simulates measurement noise and assumes the bias to vary in both tangential and transversal direction with respect to the path. It is the simulation that most accurately simulates the real experiments in the lab. For a detailed description of this simulation the reader is referred to the *real* paragraph in Section 4-1. See Figure 6-2 for the

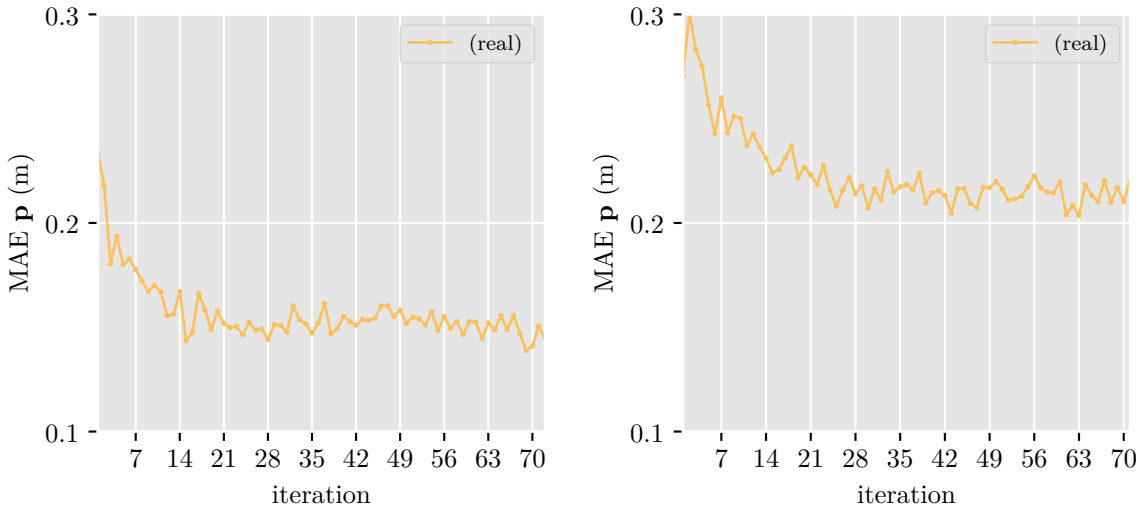

Figure 6-2: The estimation error in the position over iterations of the RIBE, where the architecture in Figure 6-1 was used. Either the experimental set-up in Appendix A (left), or the experimental set-up in Appendix B (right) was used.

results of this simulation. In both experimental set-ups, the RIBE reduces the error in the position estimate and does not show any divergent behavior.

If the performance of the real simulations in Figure 6-2 is compared with the performance of the sequential bias estimator (SBE) in Figure 4-2, this framework is outperformed by the SBE with $N^b = 3$ and the experimental set-up in Appendix B. However, the SBE diverged in the real experimental set-up in Appendix A, whereas this framework showed improvements and did not diverge. If the performance of the real simulations in Figure 6-2 is compared with the performance of the IBE in Figure 5-2, this framework performs better in both experimental set-ups.

6-1-3 Experimental Results

See Figure 6-3 for the experimental results of four runs that used the experimental set-up in Appendix A. The performance initially decreases in the first 4 iterations. This is attributed to the sub-optimally set weighting matrix Π_0^{-1} in the RLS filter. See Section 5-1 for a description on how Π_0^{-1} was obtained. Herein, N^{sub} path variables η were sampled uniformly along the path. The variable N^{sub} represents the expected number of collected range measurements per anchor in one iteration. By constructing Π_0^{-1} with twice or triple the number of sampled path variables, the initial performance decrease in the first few iterations can be avoided, because the overestimated initial guess for the variance $\bar{\boldsymbol{\theta}}^{\sigma^2}$ is given a stronger weight. Though, this might lead to a slower convergence.

The estimator performance consistently improved in all four runs with significant reductions in the position $\boldsymbol{p}$ and velocity $\boldsymbol{v}$ errors with respect to the base-case experiments. However, a similar level of performance as the motion-capture experiments is not achieved. The path tracking performance plot in Figure 6-3 shows that the proposed framework consistently improves performance in different runs in both ξ, and $\dot{\xi}$, with respect to the base-case experiments. Concerning $\dot{\xi}$, a similar performance level with respect to the motion-capture

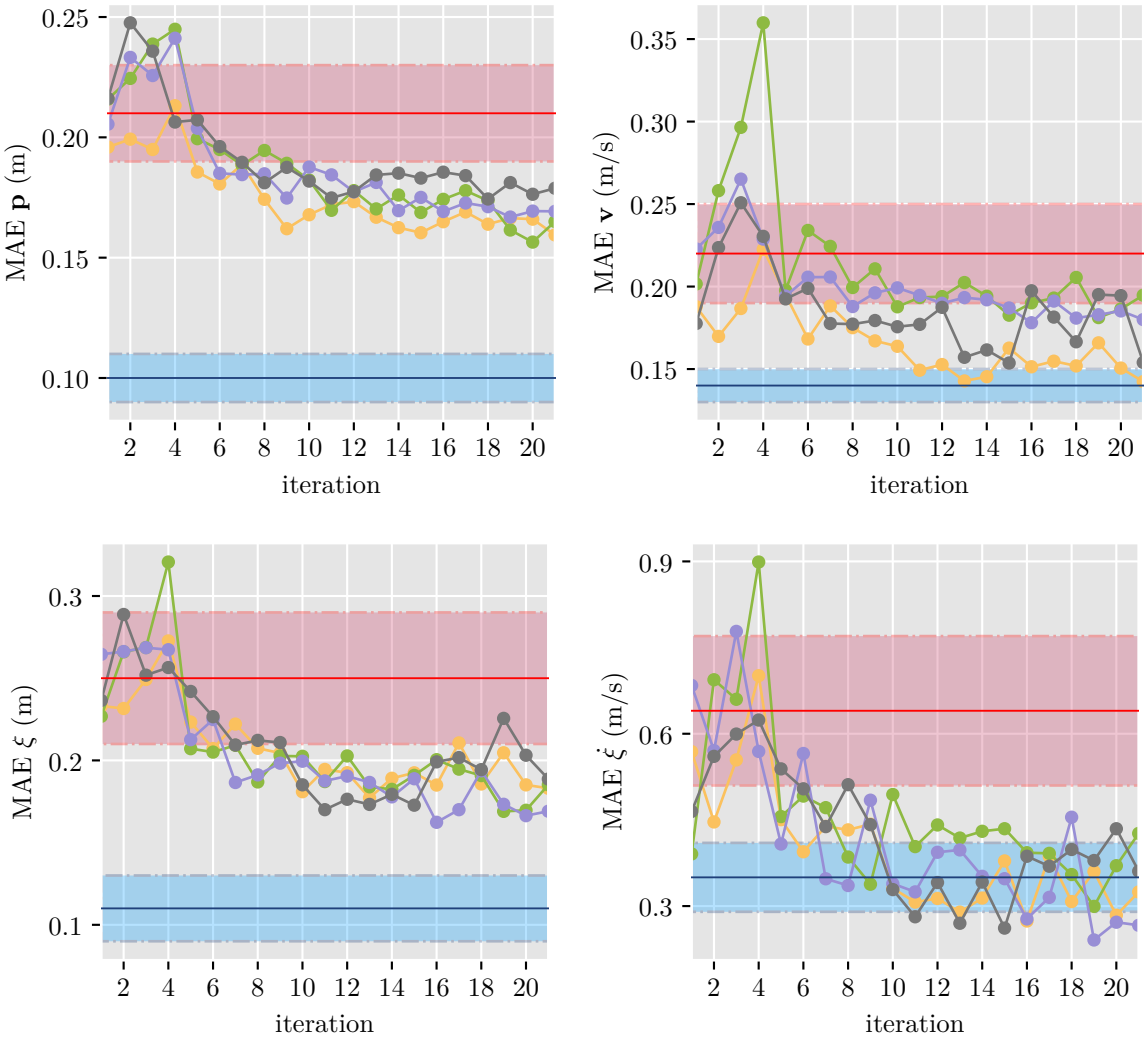

Figure 6-3: The estimator performance (top) and path tracking performance (bottom) of the RIBE are plotted for four runs, where the architecture in Figure 6-1 was used. The mean performance (line) with one standard deviation (colored area) of the base-case (red) and motion-capture (blue) experiments are plotted for comparison.

experiment is obtained. The improvements in $\dot{\xi}$ are ascribed to the inclusion of a position-dependent noise component, in addition to the mean bias and result in reduced oscillations during flight. See `https://youtu.be/J-htfbzf40U` for a video, showcasing the improved flight performance. The performance of the RIBE framework largely depends on the ability of the classifier to correctly distinguish reliable from unreliable measurements. Therefore, the difference in path tracking performance improvement between the framework (26%), and best case (56%), is primarily attributed to the sub-optimal performance of the classifier.

The comparison between the performance in Figure 6-3 with the performance of the two state augmentation frameworks in Figure 4-3 and Figure 5-3 shows that this framework has the lowest path tracking error ξ. Both the IBE and this framework achieve the same performance in $\dot{\xi}$ as the motion-capture experiments. This is attributed to the accommodation of the position-dependent noise component.

In Appendix F-4 the results of several experiments are presented where the architectures of the SBE in Figure 4-1 and the IBE in Figure 5-1 are modified, such that a constant bias could be estimated for each anchor. The mitigation of systematic errors in sensor measurements by estimating a constant bias is a simple and commonly used strategy [17]. In addition to the constant bias estimation, one experiment also uses range outlier rejection based on the squared Mahalanobis distance (MD) to reject range measurements with large errors. This is a common method to mitigate outliers [18],[14]. A comparison of the results clearly shows that the RIBE performs better than the constant bias experiments in Appendix F-4.

Chapter 7

Conclusion

In Chapters 4 and 5 attempts were made to augment the agent's state with bias parameters and estimate the augmented state either sequentially or with updates over iterations. Though stable configurations were found in simulation, the sequential bias estimator (SBE) was prone to diverge in practice. The divergent behavior was attributed to the augmented state being weakly observable given the experimental set-up, and to the model mismatch between the bias parametrization and the actual bias profile.

The iterative bias estimator (IBE) used non-causal filtering and accounted for the position-dependent noise component, therefore reducing the degree of model mismatch. Presumably, this prevented the IBE from diverging. Nonetheless, it also did not provide any significant estimator performance improvements in the position for the used experimental set-up. Various simulations demonstrated that the experimental set-up influences the observability of the augmented state. An extensive observability analysis is left for future work, as it could provide guidelines for anchor placement, such that the observability of the augmented state can be maximized for a given reference path.

Overall, the reliable iterative bias estimator (RIBE) in Chapter 6 performed best compared to the two state augmentation frameworks, as it consistently improved the path tracking performance in different experimental set-ups. The framework did not show any divergent behavior, as any observability issues were circumvented with the use of a classifier. The results demonstrated that position-dependent mitigation of range errors improves performance compared to conventional methods such as constant bias estimation. Especially the mitigation of the position-dependent noise component in addition to the mean bias smoothens the overall flight. See `https://youtu.be/J-htfbzf40U` for a video, showcasing the improved flight performance.

Position-dependent errors are not unique to ultra-wideband (UWB) range measurements. For instance, the air that is thrust down by the quadcopter's propellers bounces back when the quadcopter flies close to the ground, inducing a position-dependent error in the barometer's pressure measurements. For future work, it would be interesting to apply the proposed framework on other sensors such as a barometer to mitigate the position-dependent errors.

The performance of the iterative bias estimation largely depends on the ability of the classifier to correctly distinguish reliable from unreliable measurements. Therefore, improvements in the classifier are expected to be most effective in improving the overall performance of the framework. For example, the additional inclusion of inertial measurement unit (IMU) measurements into the classification scheme is worth investigating. Also, completely different classification schemes based on channel impulse response data should be tested and could lead to a reduction of the classifier's computational complexity. Finally, it would be interesting to apply iterative learning control on top of the presented framework for high-performance maneuvering under UWB.

Appendix A

Experimental Set-Up 1

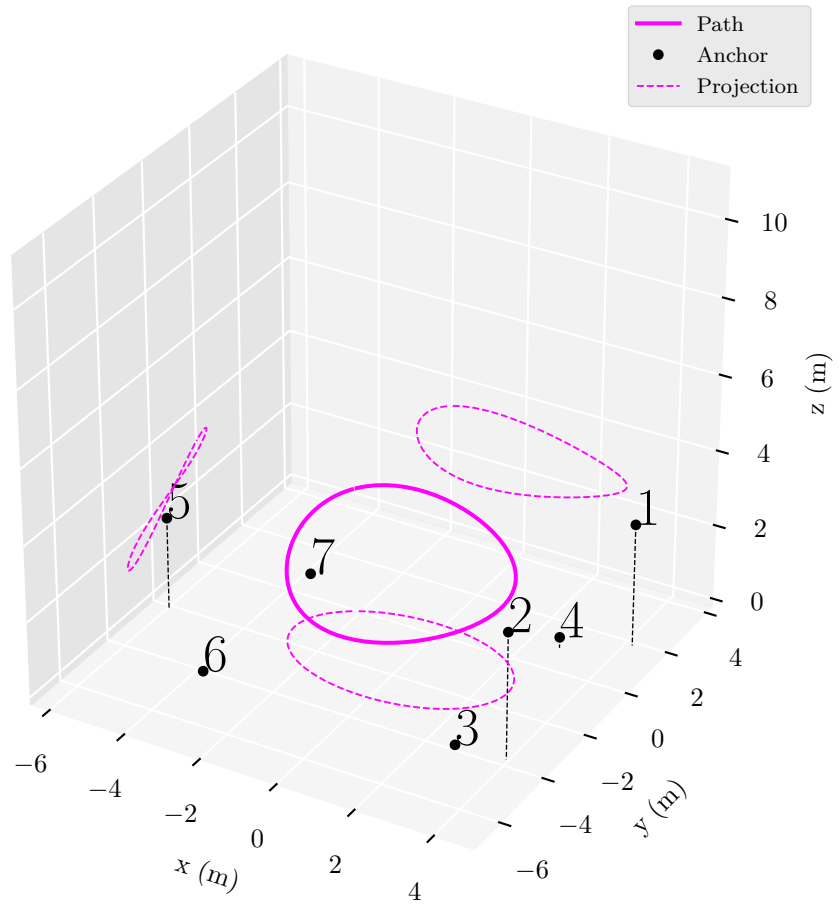

Figure A-1: The reference path and anchor placement. The path is also projected on the xy-, xz-, and yz-plane to improve the visualization.

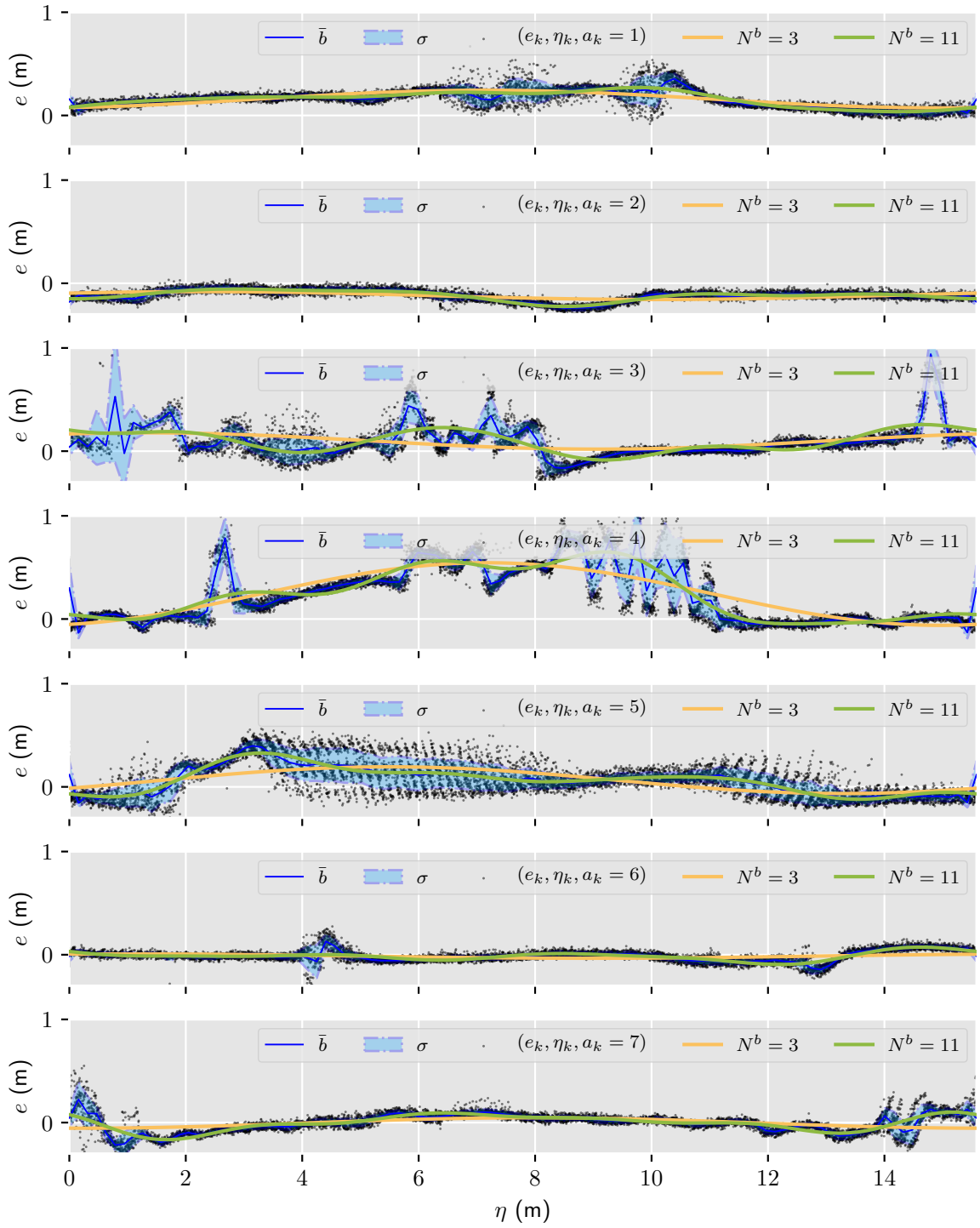

Figure A-2: The bias profiles of the anchors used in the real experiments and shown in Figure A-1. The top-left, top-right, lower-right, lower-left anchors in Figure 2-1 correspond to the bias profiles of anchor $a_k \in \{1, 2, 3, 4\}$ in this figure, respectively. Two piece-wise linear parametrizations (see Section 2-2-2) are fitted on the range error set (see (2-5)) to obtain $\bar{b}, \sigma$, using a least-squares approach with $N^b = 100$. Two truncated Fourier parametrizations are used to fit the mean $\bar{b}$ with $N^b = 3$ (orange) and $N^b = 11$ (green).

Appendix B

Experimental Set-Up 2

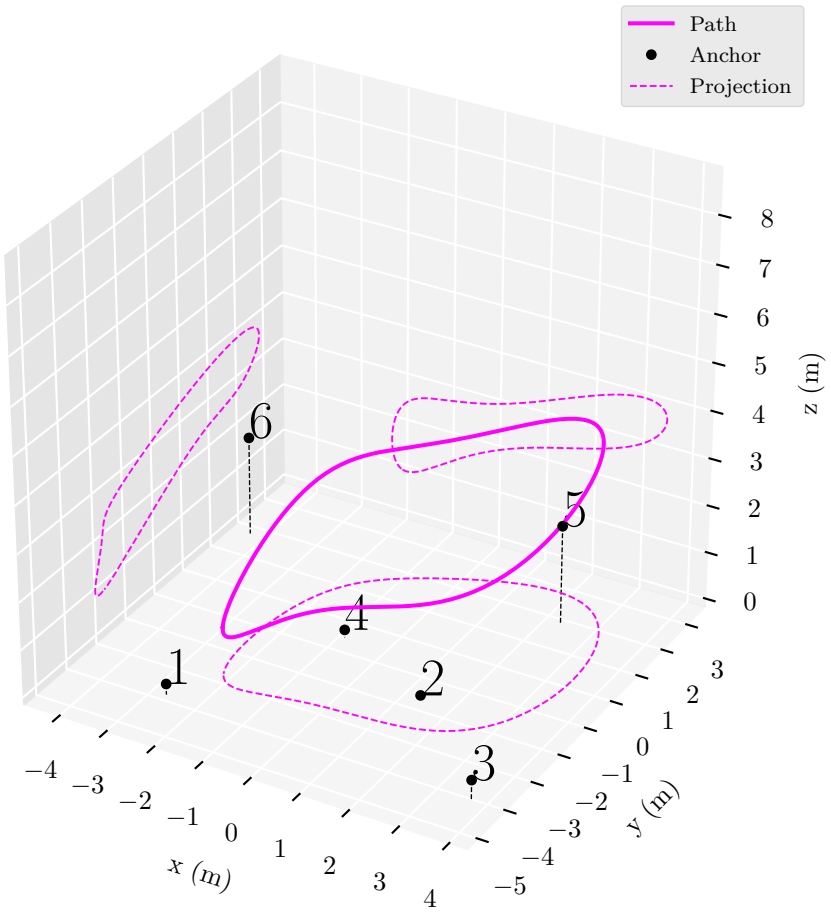

Figure B-1: The reference path and anchor placement. The path is also projected on the xy-, xz-, and yz-plane to improve the visualization.

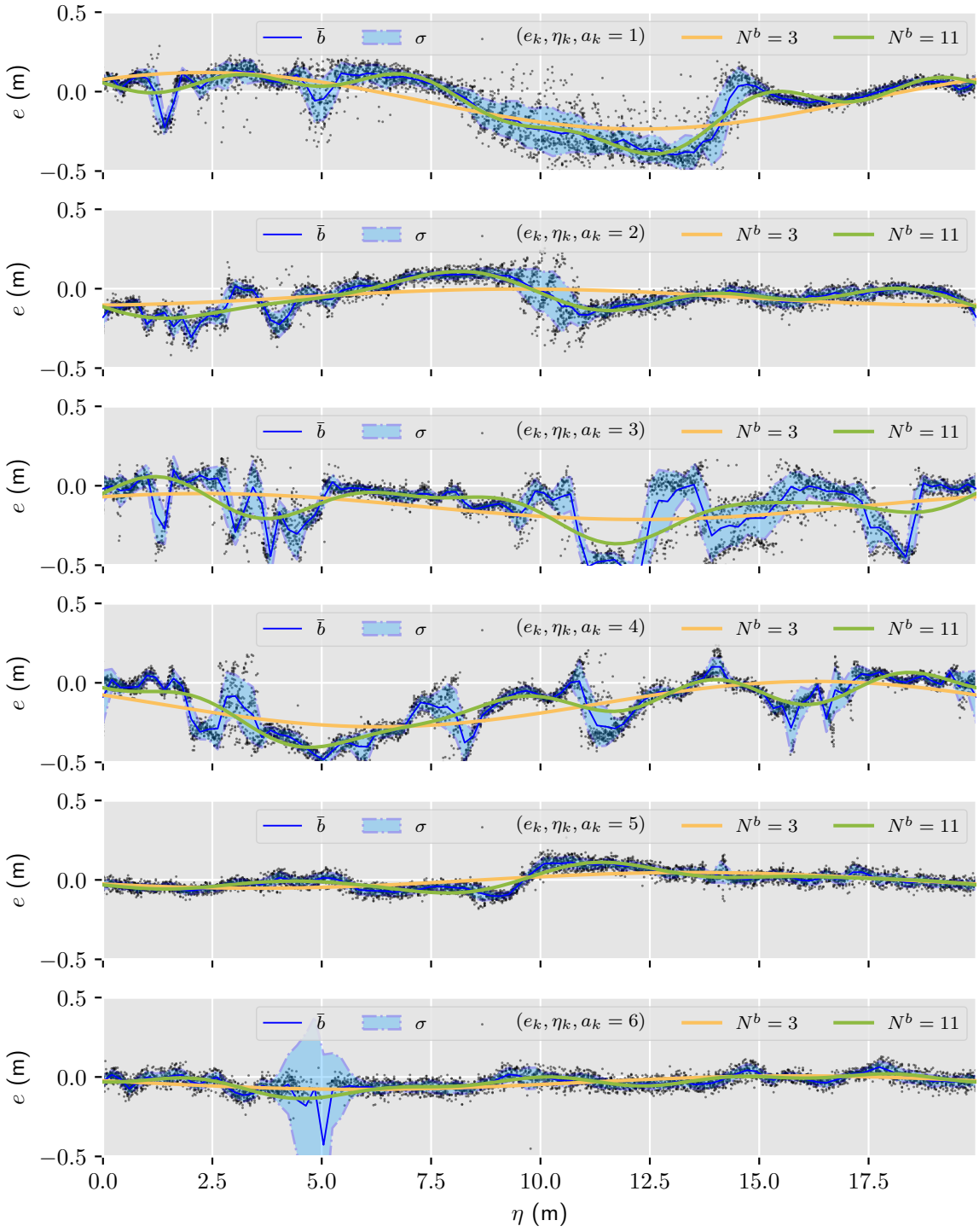

Figure B-2: The bias profiles of the anchors in Figure B-1. Two piece-wise linear parametrizations (see Section 2-2-2) are fitted on the range error set (see (2-5)) to obtain $\bar{b}, \sigma$, using a least-squares approach with $N^b = 100$. Two truncated Fourier parametrizations are used to fit the mean $\bar{b}$ with $N^b = 3$ (orange) and $N^b = 11$ (green).

Appendix C

Hardware

The quadcopter uses the frame, motors, and rotor controllers of an Ascending Technology Hummingbird. The motion controller and online state estimator are run at 200 Hz on a Snapdragon flight board, located onboard the quadcopter. Both the anchors and quadcopter are equipped with DWM 1000 modules and communicate using the two-way ranging algorithm with repeated reply described in [18]. The quadcopter communicates with every anchor in sequential order at a frequency of 200 Hz. The MPU9250 IMU and BMP280 barometer modules are integrated on the Snapdragon flight board, and provide measurements with a frequency of 1 kHz and 50 Hz, respectively. See Figure C-1 for the acceleration, angular rate, and barometer measurements that were collected while flying the path in Figure 2-1 and Appendix A using motion-capture data.

IMU Figure C-1 shows that the acceleration measurements are very noisy, while the angular rate measurements have little noise. The noisy acceleration measurements are in part caused by vibrations in the frame, induced by the spinning of the rotors. The measurement noise covariance matrices for the acceleration and angular rate measurements are set to $\Sigma^{\text{acc}} = \text{diag}(3.4^2, 3.4^2, 3.4^2)$ and $\Sigma^{\text{gyr}} = \text{diag}(0.14^2, 0.14^2, 0.14^2)$, respectively.

Barometer The barometer measures the air pressure of the environment. The relation between pressure and altitude is given by the isothermic barometric formula [3]

$$P(p_z) = P(p_z^{\text{ref}}) \exp\Big(- \frac{Mg}{RT}(p_z - p_z^{\text{ref}})\Big), \tag{C-1}$$

where P denotes pressure, p_z^{ref} is the reference height, M is the molar mass of air, g is the gravitational acceleration, R is the universal gas constant, and T is the temperature which is assumed to be constant. By linearizing (C-1) around p_z^{ref} and rewriting for the altitude, the linear relation

$$z_l^{\text{bar}} = \frac{RT}{Mg}\Big(1 - \frac{P_l}{P(p_z^{\text{ref}})}\Big) + p_z^{\text{ref}} \tag{C-2}$$

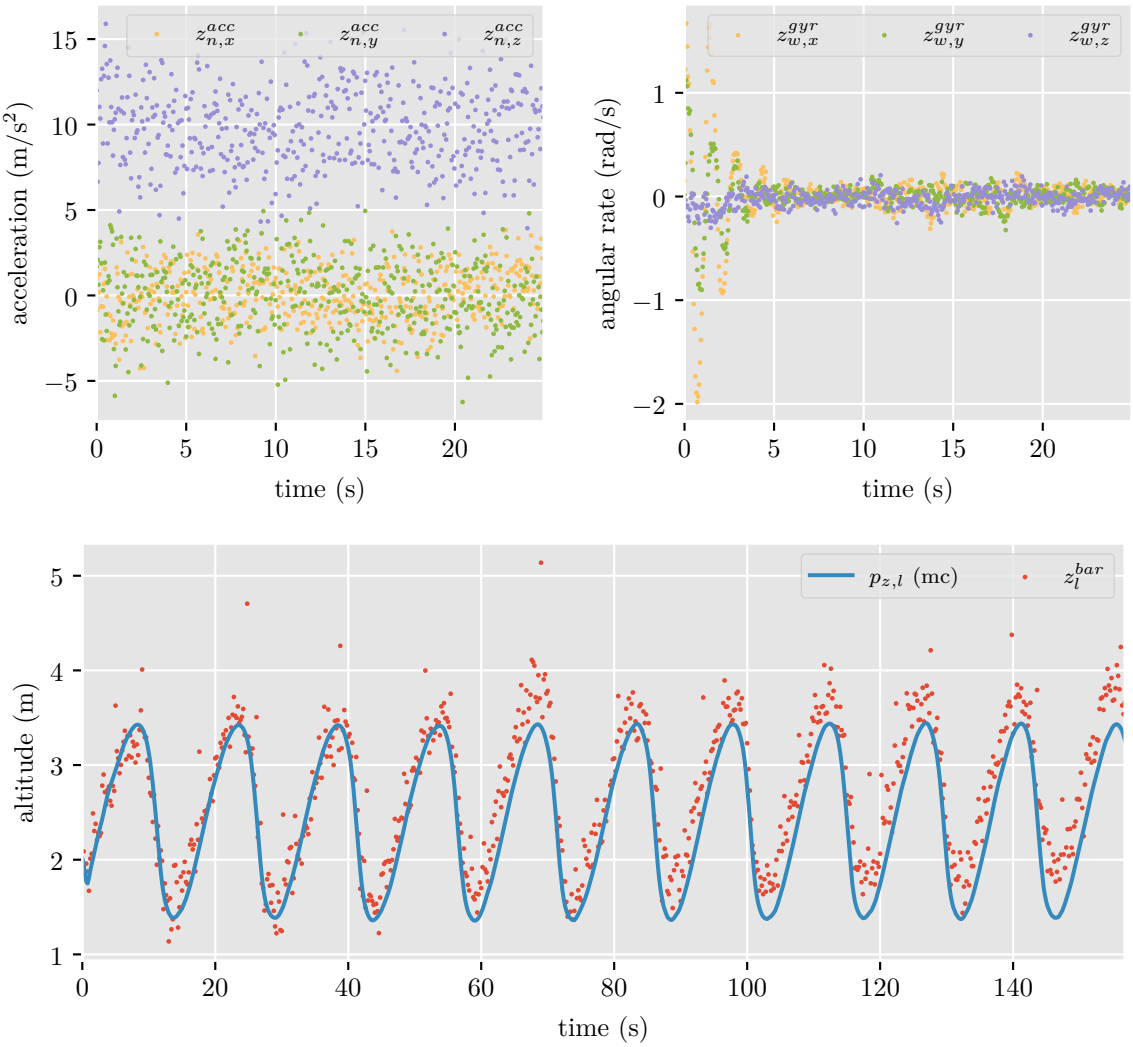

Figure C-1: The acceleration measurements $\boldsymbol{z}_n^{\text{acc}}, n \in \mathcal{Z}^{\text{acc}}$ (top left), angular rate measurements $\boldsymbol{z}_w^{\text{gyr}}, w \in \mathcal{Z}^{\text{gyr}}$ (top right), and altitude measurements $\boldsymbol{z}_l^{\text{bar}}, l \in \mathcal{Z}^{\text{bar}}$. The figures only show every 50^{th} acceleration, 50^{th} angular rate, and 10^{th} altitude measurement.

is obtained and used to transform the pressure measurements P_l to altitude measurements z_l^{bar} with $l \in \mathcal{Z}^{\text{bar}}$. Prior to flying, the reference pressure $P(p_z^{\text{ref}})$ is measured at the known reference height p_z^{ref}. See Figure C-1 for the transformed altitude measurements of ten full iterations. The figure illustrates the slow drifting behavior over iterations. The measurement noise covariance of the altitude measurements is set to $\Sigma^{\text{bar}} = 0.44^2$.

Range In Appendix A the bias profiles of seven anchors are shown for the given experimental set-up. In case the anchor has line-of-sight (LOS) with the agent, the range measurements are assumed to only be corrupted by white noise ν^{uwb} with measurement covariance $\Sigma^{\text{uwb}} = 0.05^2$.

Appendix D

Path Following Controller

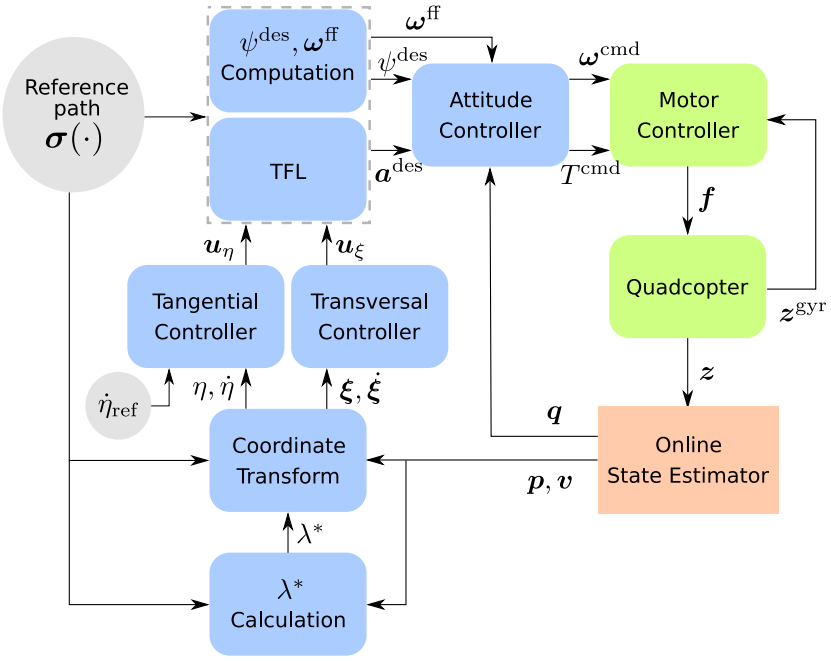

Figure D-1: The architecture of the motion controller (blue) and dynamical system (green). A modified version of a Figure in [11] was taken with permission of the authors.

The motion of a quadcopter is ultimately determined by the thrust $\boldsymbol{f} = (f_1, f_2, f_3, f_4)$ generated by each of the four propellers. All propellers point in the same direction, so to change the flight direction, a quadcopter has to change its attitude $\boldsymbol{q}$. By differences in propeller thrust, the quadcopter can do so. The combined thrust $F = \sum_{i=1}^{4} f_i$ generated by the four propellers determines the acceleration of the quadcopter.

See Figure D-1 for the control architecture that is used to control the motion of the quadcopter [11]. The attitude controller determines the required body rates to change the quadcopter's

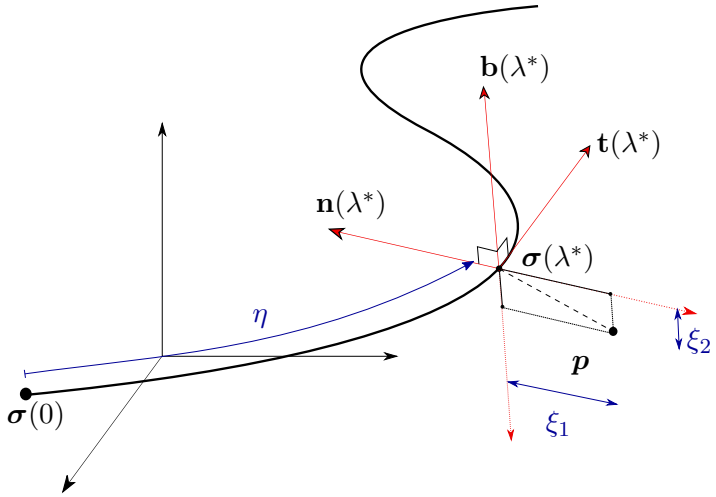

Figure D-2: The FSF state transformation. A modified version of a Figure in [11] was taken with permission of the authors.

attitude to the desired pose. It takes the desired acceleration $\boldsymbol{a}^{\text{des}}$, desired yaw orientation ψ^{des}, the attitude $\boldsymbol{q}$ and a body rate feedforward term $\boldsymbol{\omega}^{\text{ff}}$ as inputs. The attitude controller outputs the desired mass normalized combined thrust of the four propellers T^{cmd} and body rate commands $\boldsymbol{\omega}^{\text{cmd}}$ to the motor controller. The motor controller is tasked with following these commands using the angular rate measurements $\boldsymbol{z}^{\text{gyr}}$ in feedback, and results in the thrust vector $\boldsymbol{f}$. It is assumed that the inner loops containing the attitude controller, motor controller and quadcopter have sufficiently fast dynamics, such that they can be modeled as a double integrator, i.e. only the position and velocity have to be considered in the design of the path following controller (PFC).

The goal of the PFC is to find a time-invariant control law for $\boldsymbol{a}^{\text{des}}$, $\boldsymbol{\omega}^{\text{ff}}$ and $\boldsymbol{\psi}^{\text{des}}$, such that the quadcopter's position $\boldsymbol{p}$ exponentially approaches a given path $\boldsymbol{\sigma}$, and traverses along this path with a user-defined constant reference speed $\dot{\eta}_{\text{ref}}$. The control law must also ensure path invariance and path attractiveness. Path invariance means that the quadcopter will not leave the path once it is has reached the path. Path attractiveness means that the quadcopter can reach the path from any arbitrary initial position. The main components in Figure D-1 briefly discussed. For a detailed description of the implemented PFC the reader is referred to [11]. See Figure D-3 for the performance of the implemented PFC for four full iterations on the quadcopter platform.

Coordinate transform After calculating λ^* with (2-3), a state transformation is performed using Frenet-Serret frames (FSF). This transformation maps the position $\boldsymbol{p}$ and velocity $\boldsymbol{v}$ to a set of states along the reference path (tangential) and towards the reference path (transversal). See [11] for a detailed description on FSF. Figure D-2 illustrates the transformation of the position state $\boldsymbol{p}$, onto the unit-tangent vector $\boldsymbol{t}$, unit-normal vector $\boldsymbol{n}$ and unit-binormal vector $\boldsymbol{b}$ that correspond to the FSF transformation. This results in the tangential position η and transversal positions $\boldsymbol{\xi} = (\xi_1, \xi_2)$. The time derivative of the transformed position states results in the transformed velocity states denoted as $\dot{\eta}$ and $\dot{\boldsymbol{\xi}} = (\dot{\xi}_1, \dot{\xi}_2)$. The transversal distance towards the path is denoted by scalar $\xi = \| \boldsymbol{\xi} \|$.

TFL The transversal states $\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}$ describe the motion of the quadcopter off the path. By stabilizing the origin of the transversal subsystem, the PFC forces the quadcopter to remain on the path. Simple PID controllers are used to stabilize the subsystem. In a similar fashion, a PI controller is used to make the tangential velocity $\dot{\eta}$ track the user-defined tangential reference speed $\dot{\eta}_{\text{ref}}$. Together, the separate linear controllers produce the input vectors u_η and $\boldsymbol{u}_\xi$. Using transverse feedback linearization (TFL), the input vectors can be transformed back to a desired acceleration $\boldsymbol{a}^{\text{des}}$.

$\psi^{\text{des}}, \boldsymbol{\omega}^{\text{ff}}$ computation When assuming the quadcopter to track the path accurately with the desired tangential reference velocity, a body rate feedforward $\boldsymbol{\omega}^{\text{ff}}$ can be calculated. Even when $\dot{\eta}_{\text{ref}}$ is not tracked perfectly, the feedforward term significantly improves performance. The desired yaw ψ^{des} is set to zero, meaning no further calculation are required.

Performance See Figure D-3 for the performance of the implemented PFC for four full iterations on the quadcopter platform while flying on motion-capture data. It is clear that the tangential reference speed $\dot{\eta}_{\text{ref}} = 1.5$ m/s is not properly tracked. The flight performance of the controller could be improved through tuning. However, this would likely result in a more aggressive controller that might perform worse when the controller is provided with a less accurate estimate of the agent's state. A high performing controller is not the main objective of this thesis. Rather, the aim is to have consistent flight performance in the position over iterations, so that the position-dependent characteristic of the range errors can be exploited. The transversal distance ξ does decrease to roughly zero.

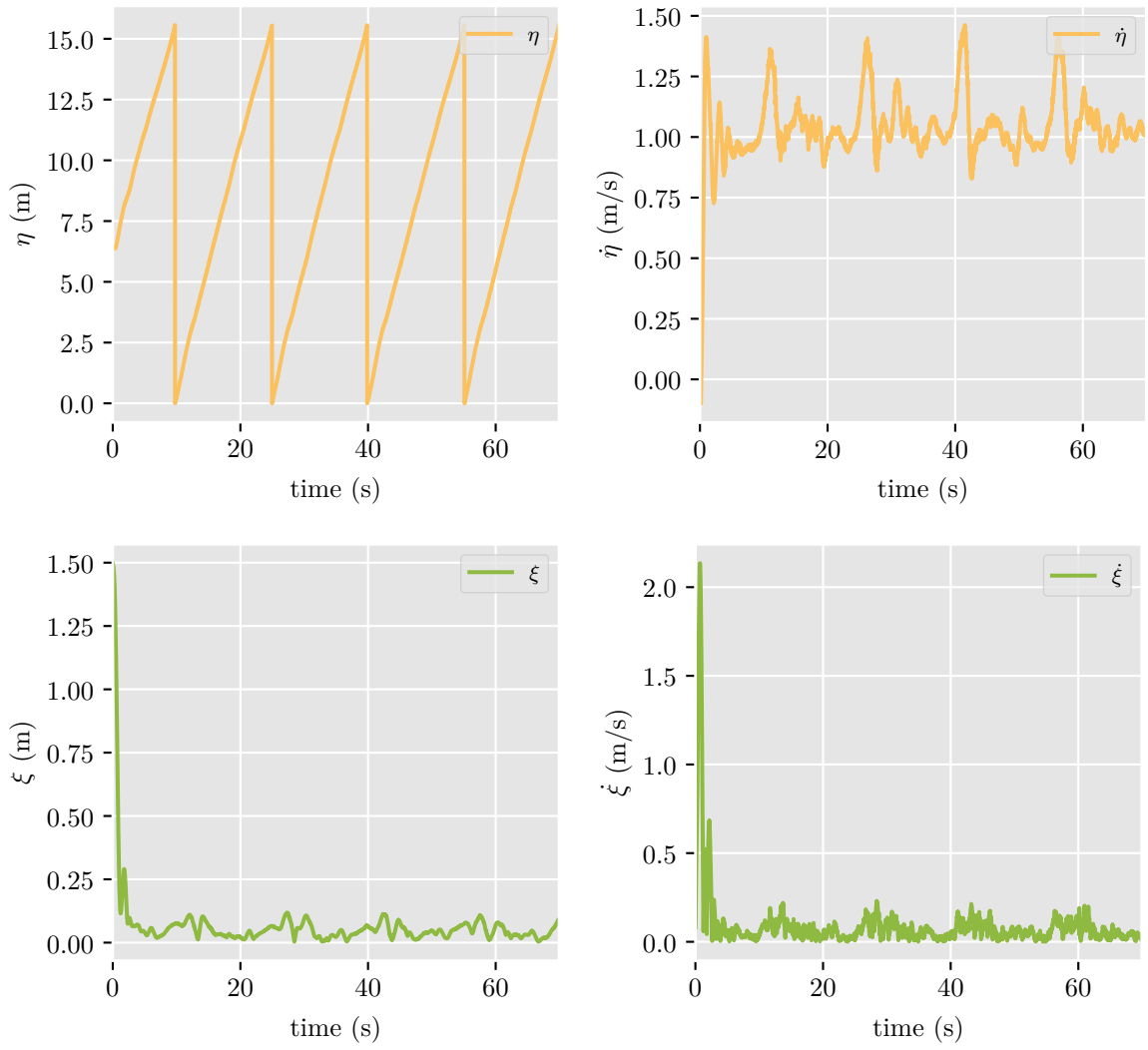

Figure D-3: The performance of the path following controller for four complete iterations, where the motion-capture system is used to provide the motion controller with the quadcopter's state. The figures on top show the tangential position and velocity, and the bottom figures show the transversal position and velocity. The quadcopter is initialized off the path, hence explaining the initial transient in the transversal distance ξ.

Appendix E

Observability Analysis

In a state augmentation framework, the agent's state $\boldsymbol{x}$ is augmented with the bias parameters such that both can be estimated simultaneously. State augmentation is a common approach for compensating biases [9]. An observability analysis can give insight into the effect of the anchor placement, reference path and bias profile on the observability of the augmented state. First, the conditions for observability are given, followed by the introduction of a simplified scenario that makes the analysis tractable. Then, the observability is analyzed for a trivial scenario where the problem formulation becomes linear to demonstrate how observability can be analyzed. The analysis for non-trivial scenarios is left for future work.

Conditions for Observability The conditions for local observability will be given for autonomous nonlinear systems, defined as

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})u \text{ with } \boldsymbol{x} \in \mathbb{R}^n \\ \boldsymbol{y} &= \boldsymbol{h}(\boldsymbol{x}) \text{ with } \boldsymbol{y} \in \mathbb{R}^p \end{aligned} \tag{E-1}$$

where $\boldsymbol{h} = (h_1, h_2, \ldots, h_p)$. Loosely speaking, the system in (E-1) is said to be locally observable at $\boldsymbol{x}_0$ if every state in the neighborhood of $\boldsymbol{x}_0$ can be distinguished from its neighbors by using system trajectories that remain close to $\boldsymbol{x}_0$. For a detailed explanation and rigorous definition of local observability, the reader is referred to [19]. To study the observability of the system in (E-1), the observation space $\mathbb{O}$ is defined as

Definition 1 (Observation Space)**.** *The observation space* $\mathbb{O}$ *of* (E-1) *is the linear space of functions on* $\mathbb{R}^n$ *containing* $h_1, h_2, \ldots, h_p$*, and all repeated Lie derivatives, i.e.,*

$$\mathcal{L}_{\boldsymbol{\xi}}^k \mathcal{L}_{\boldsymbol{\xi}}^{k-1} \ldots \mathcal{L}_{\boldsymbol{\xi}} h_j(\boldsymbol{x}) \text{ with } j \in \{1, 2, \ldots, p\},\ k = 1, 2, \ldots, \tag{E-2}$$

for $\boldsymbol{\xi} \in \{\boldsymbol{f}, \boldsymbol{g}\}$ *where the Lie derivative* $\mathcal{L}_{\boldsymbol{\xi}} h_j(\boldsymbol{x})$ *is defined as*

$$\mathcal{L}_{\boldsymbol{\xi}} h_j(\boldsymbol{x}) = \frac{\partial h_j}{\partial \boldsymbol{x}}(\boldsymbol{x})\boldsymbol{\xi}(\boldsymbol{x}) \tag{E-3}$$

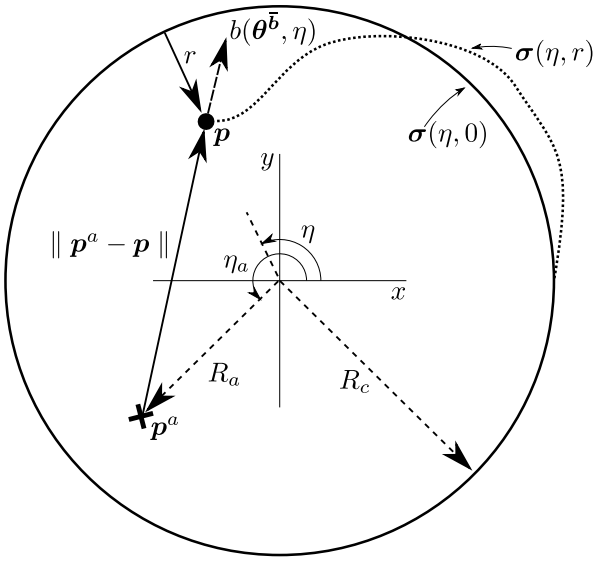

Figure E-1: The scenario of the observability analysis.

Using Definition 1, local observability for an autonomous nonlinear system can be defined as

Theorem 1 (Local Observability)**.** *Consider the system in* (E-1)*. Assume that*

$$\dim \frac{\partial \mathbb{O}}{\partial \boldsymbol{x}}(\boldsymbol{x}_0) = n, \tag{E-4}$$

then the system is locally observable at $\boldsymbol{x}_0$*. Hence, the system in* (E-1) *is locally observable if it is locally observable for every* $\boldsymbol{x}_0 \in \mathbb{R}^n$*.*

Scenario The observability of the augmented state can be analyzed in a simplified scenario which is illustrated in Figure E-1 and further specified as follows.

1. A deterministic scenario is considered.
2. The motion control input is determined independently from the state estimator, i.e. there is no interaction between the state estimator and controller.
3. The agent is a point mass.
4. All sensors provide continuous measurement trajectories, i.e. a sensor measurement $z(t)$ is a continuous function of time.
5. The agent's global z-coordinate p_z is directly observable due to the altitude measurements $z^{\text{bar}}(t)$. Therefore, the analysis is performed in 2D.
6. Acceleration measurements $\boldsymbol{z}^{\text{acc}}(t)$ are collected in the global frame. Therefore, the angular rate measurements $\boldsymbol{z}^{\text{gyr}}(t)$ are not not considered in the analysis as the agent is a point mass, i.e. no attitude, and the acceleration measurements are already in the global frame.

7. Polar coordinates are used to describe the agent's position $\boldsymbol{p}$ in the global frame. Herein, η is the angular coordinate and the radial distance r is measured with respect to a certain known radius R_c. Then, the augmented state can be defined as $\boldsymbol{x} = (\eta, \dot{\eta}, r, \dot{r}, \boldsymbol{\theta}^{\bar{b}})$.

8. The flown trajectory $\boldsymbol{\sigma}(\eta, r)$ in the global euclidean frame is defined as

$$\boldsymbol{\sigma}(\eta, r) = \Big((R_c - r)\cos(\eta), (R_c - r)\sin(\eta) \Big) \tag{E-5}$$

9. The reference path is a circle with radius R_c and can be written as $\boldsymbol{\sigma}(\cdot, 0)$ with the definition in E-5. The agent tracks the circular path with tangential velocity $\dot{\eta}_{\text{ref}}$.

10. The range measurements $z^{\text{uwb}}(t)$ are corrupted by a deterministic tangential position-dependent bias fully described by the mean bias parameters $\boldsymbol{\theta}^{\bar{b}}$ and η for a given bias parametrization. One anchor is considered with its location $\boldsymbol{p}^a$ specified in polar coordinates (η_a, R_a) where η_a is the angular coordinate and R_a the radial distance. This results in the measurement model

$$z^{\text{uwb}} = \underbrace{\overbrace{\sqrt{R_a^2 + (R_c - r)^2 - 2R_a(R_c - r)\cos(\eta - \eta_a)}}^{\|\boldsymbol{p}^a - \boldsymbol{p}\|} + b(\boldsymbol{\theta}^{\bar{b}}, \eta)}_{h(\eta, r, \boldsymbol{\theta}^{\bar{b}})} \tag{E-6}$$

11. The accelerations $\ddot{\eta}$ and $\ddot{r}$ can be inferred from the acceleration trajectory $\boldsymbol{z}^{\text{acc}}(t)$. These accelerations are taken as inputs to the system, such that the dynamical system can be defined to evolve as

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}_1\ddot{\eta} + \boldsymbol{g}_2\ddot{r} \\ y &= h(\boldsymbol{x}) \end{aligned} \tag{E-7}$$

where $h(\boldsymbol{x}) = h(\eta, r, \boldsymbol{\theta}^{\bar{b}})$ in (E-6) and

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{x}) &= (\dot{\eta}, 0, \dot{r}, 0, \mathbf{0}) & \boldsymbol{g}_1 &= (0, 1, 0, 0, \mathbf{0}) \\ & & \boldsymbol{g}_2 &= (0, 0, 0, 1, \mathbf{0}) \end{aligned} \tag{E-8}$$

The agent is assumed to track the reference path with the tangential reference speed, i.e. $\ddot{\eta} = 0$ and $x = (\eta, \dot{\eta}_{\text{ref}}, 0, 0, \boldsymbol{\theta}^{\bar{b}})$. Given the anchor location (η_a, R_a) and the type of bias parametrization, the problem is to find out whether the system is locally observable at $x_0 = (\eta, \dot{\eta}_{\text{ref}}, 0, 0, \boldsymbol{\theta}^{\bar{b}})$ for $\forall\eta, \forall\boldsymbol{\theta}^{\bar{b}}$. According to Definition 1, the observation space $\mathbb{O}$ of (E-7) is the linear space of functions on $\mathbb{R}^{4+N^b}$ containing $h(\boldsymbol{x})$, and all repeated Lie derivatives, i.e.,

$$\mathcal{L}_{\boldsymbol{\xi}}^k \mathcal{L}_{\boldsymbol{\xi}}^{k-1} \dots \mathcal{L}_{\boldsymbol{\xi}} h(\boldsymbol{x}) \text{ with } k = 1, 2, \dots, \text{ for } \boldsymbol{\xi} \in \{\boldsymbol{f}, \boldsymbol{g}_2\} \tag{E-9}$$

Constant bias and anchor placed at center If the anchor is placed at the center of the circular reference path, i.e. $R^a = 0$, and the bias is assumed to be constant, i.e. no dependence on η, the range measurement model in (E-6) reduces to

$$z^{\text{uwb}} = \underbrace{R_c - r + \boldsymbol{\theta}^{\bar{b}}}_{h(r, \boldsymbol{\theta}^{\bar{b}})}, \tag{E-10}$$

which makes the output model linear, thus making the dynamical system in (E-7) a linear time-invariant (LTI) system. This allows us to easily construct the observability matrix and check its rank. The system matrices A and C can be defined as

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad C = \begin{bmatrix} 0 & 0 & -1 & 0 & 1 \end{bmatrix}, \tag{E-11}$$

such that the observation space is

$$\frac{\partial \mathbb{O}}{\partial \boldsymbol{x}} = \begin{bmatrix} C \\ CA \\ CA^2 \\ CA^3 \\ CA^4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{E-12}$$

which is not full rank. Therefore, a constant bias is not observable when flying a circular reference path with the anchor placed at its center. Moreover, even if the range measurements were unbiased, the agent's state would be unobservable. This analysis demonstrates that the observability of the state is indeed influenced by the anchor placement. The analysis for non-trivial scenarios (arbitrary anchor placement, path-dependent bias) will be more involved, as the measurement model will be non-linear. The analysis for such scenarios is left for future work.

Appendix F

Performance

The agent traverses the reference path repeatedly, where every repetition is called an iteration. A motion-capture system tracks the global position and orientation of the quadcopter with an accuracy of the order of millimeters and milliradians at 200 Hz [13]. This system is used to asses the performance of the online state estimator and motion controller over iterations.

Online State Estimator Performance The performance of the online state estimator will be evaluated based on the mean absolute error (MAE) in the position $\boldsymbol{p}$ and velocity $\boldsymbol{v}$ over iterations.

Path Tracking Performance Minimizing the distance towards the path is considered to be the most important control objective, because it ensures a safe flight. This distance is referred to as the tracking error and is defined as

$$\xi = \| \boldsymbol{p} - \boldsymbol{\sigma}(\lambda^*) \| . \tag{F-1}$$

The first time-derivative of the tracking error is denoted by $\dot{\xi}$. A large $\dot{\xi}$ means the distance error varies rapidly, which expresses itself in an oscillatory flight around the reference path. The flight performance will be evaluated based on the MAE of ξ and $\dot{\xi}$ over iterations.

Comparison Experiments The performance of the frameworks presented in this thesis are compared with that of two different experiments. In a base-case experiment, no bias estimation is performed. In a motion-capture experiment, the true range error set $\mathcal{B}$ in (2-5) is obtained with the motion-capture system, and subsequently combined over iterations in a recursive least squares (RLS) filter such that the best-possible bias fit is obtained. The performance of the motion-capture experiment is interpreted as the best case. Each experiment is executed multiple times. Only the mean performance μ with one standard deviation σ will be plotted for the base-case and motion-capture experiments in the performance plots of the proposed frameworks. See Appendix F-1 and F-2 for performance figures of the aforementioned experiments, and a short explanation on how the mean performances with one standard deviation was obtained.

F-1 Motion-Capture

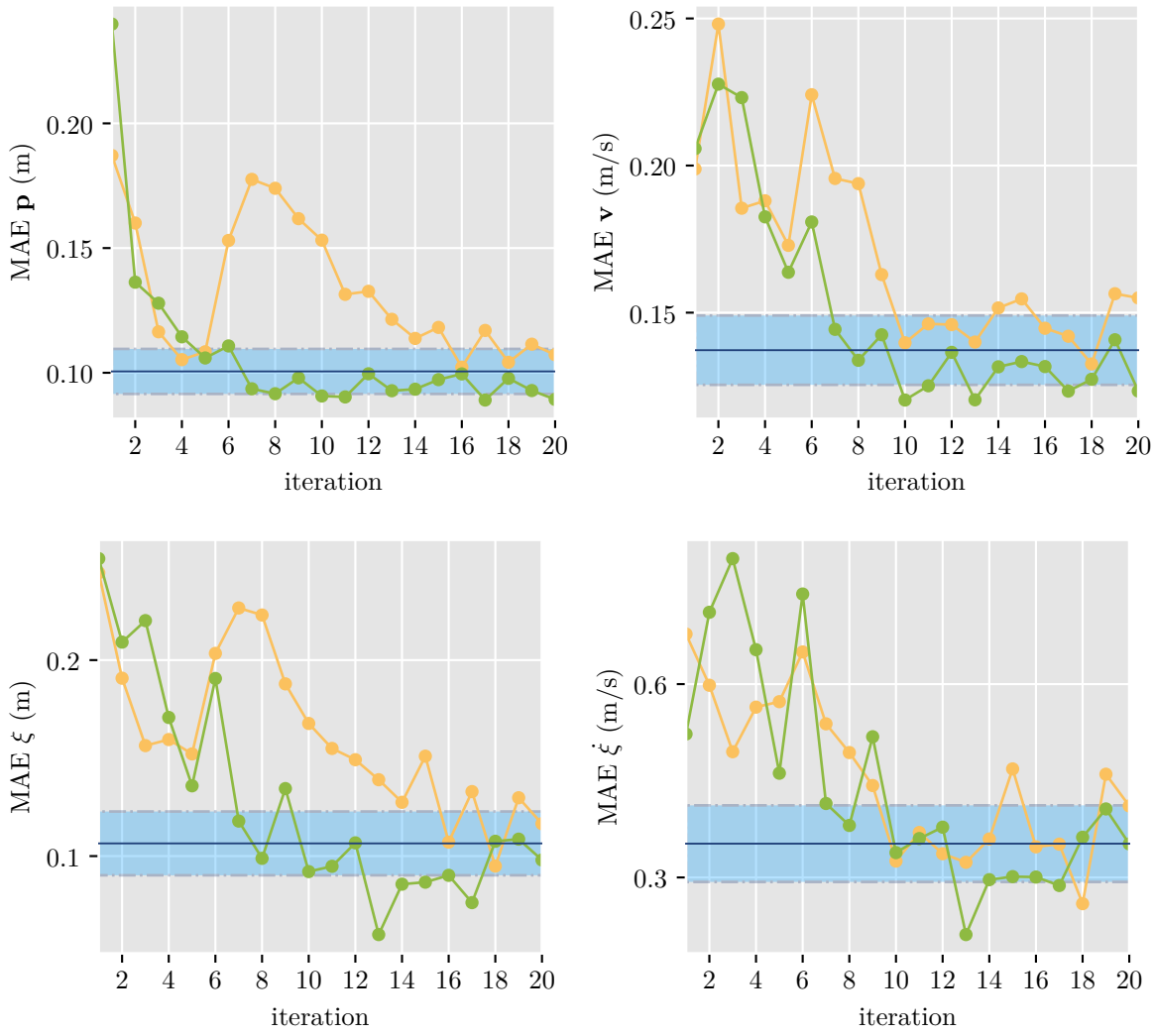

Figure F-1: The estimator performance (top) and path tracking performance (bottom) of the motion-capture experiments are plotted for two runs with the experimental set-up in Appendix A. The architecture in Figure 5-1 was used, but differs in that the range errors $\mathcal{B}_{(j)}$ are inferred with motion-capture data. Only the last 5 iterations of the two runs are used to calculate the mean performance (line) and one standard deviation (colored area) of the motion-capture experiments shown in blue. In one run (yellow-dotted), an erroneous update was performed at iteration 6. After roughly 10 iterations the effect of the erroneous update diminished, hence nicely demonstrating the effect of the forgetting factor in the RLS filter.

F-2 Base-Case

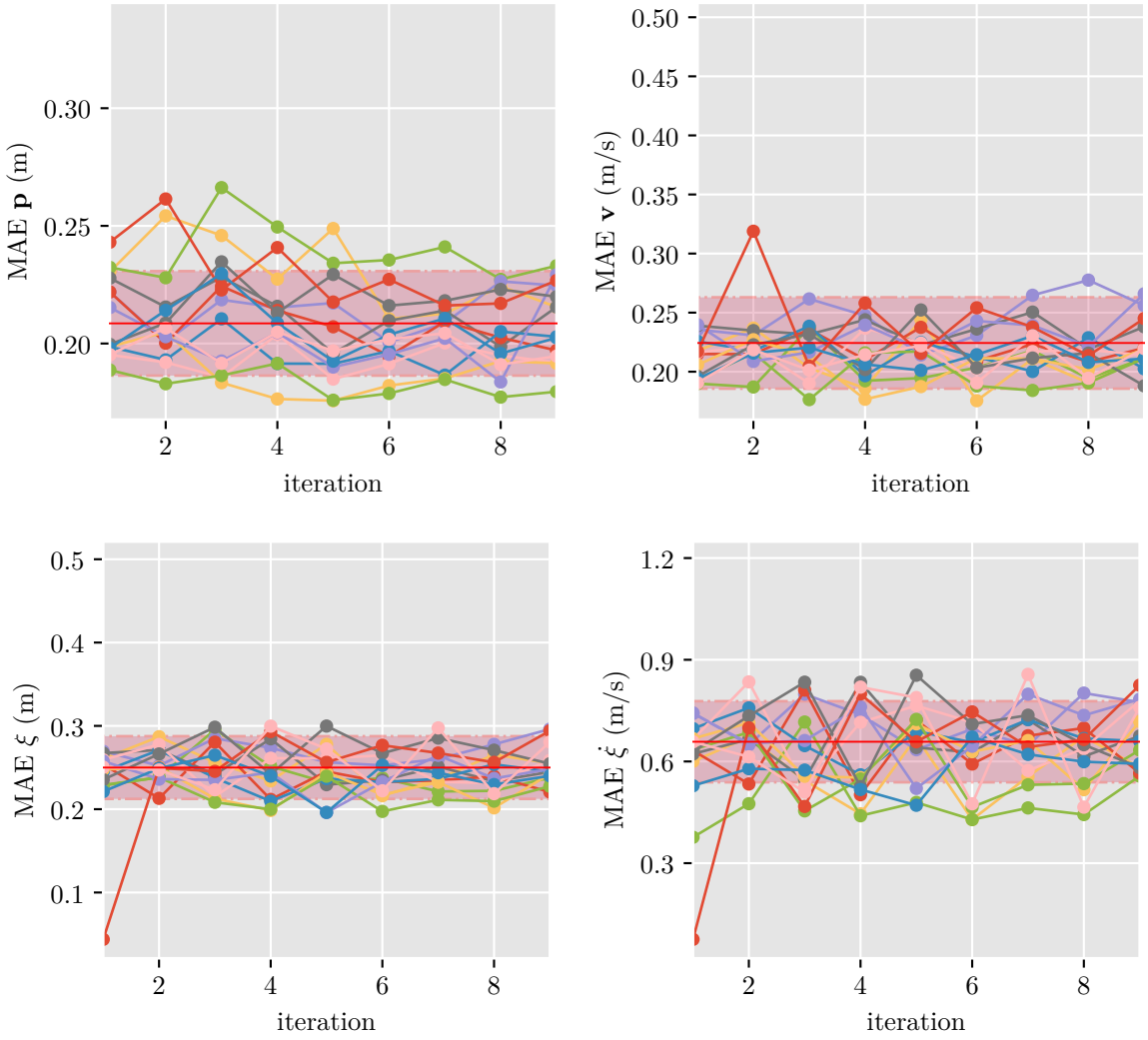

Figure F-2: The estimator performance (top) and path tracking performance (bottom) of the base-case experiments are plotted for fourteen runs with the experimental set-up in Appendix A. The architecture in Figure 2-3 was used, where the Bias Estimator block was omitted. All iterations of the fourteen runs are used to calculate the mean performance (line) and one standard deviation (colored area) of the base-case experiments, shown in red.

F-3 Variance Only

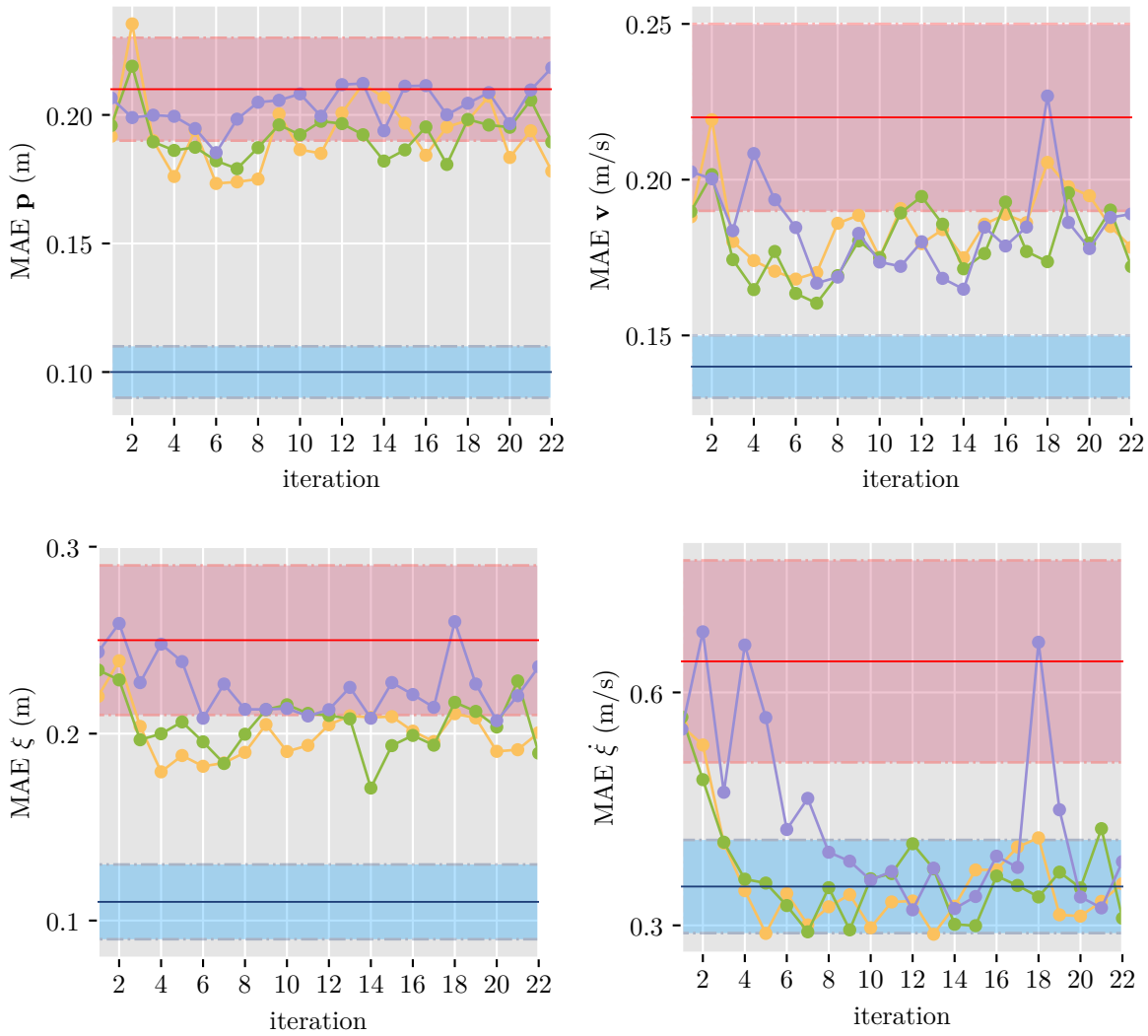

Figure F-3: The estimator performance (top) and path tracking performance (bottom) of the RIBE are plotted for three runs with the experimental set-up in Appendix A, where only the variance is estimated. The architecture in Figure 6-1 was used, where the Online State Estimator block only included the bias' variance σ^2 into the range measurement update, i.e. the mean bias $\bar{b}$ was not used. The mean performance (line) with one standard deviation (colored area) of the base-case (red) and motion-capture (blue) experiments are plotted for comparison.

F-4 Constant Bias

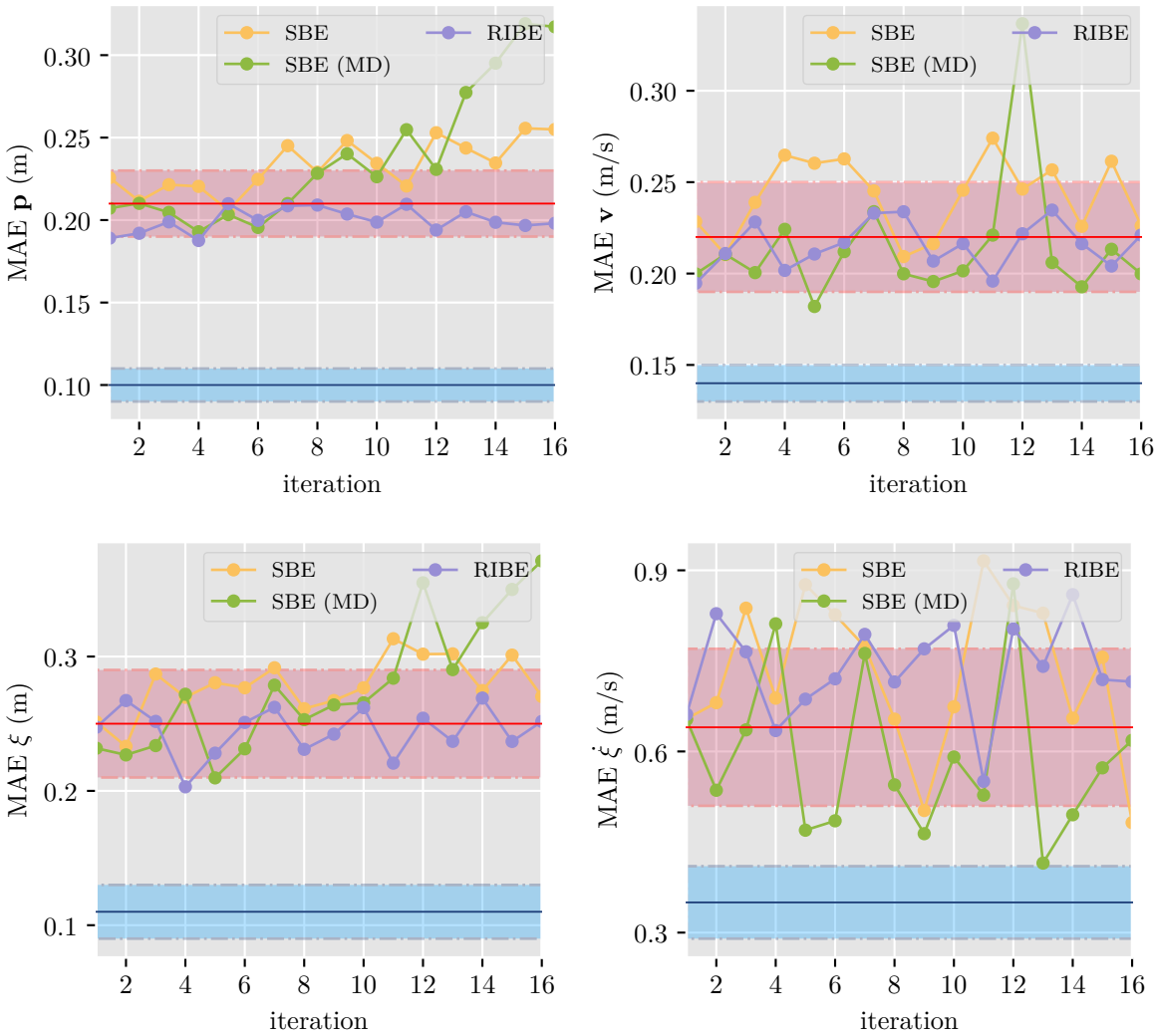

Figure F-4: The estimator performance (top) and path tracking performance (bottom) of three different experiments are plotted with the experimental set-up in Appendix A. The SBE experiments used the architecture in Figure 4-1 and $N^b = 1$. In one SBE experiment the MD was used to reject relatively biased range measurements as described in [18]. The RIBE experiment used the architecture in Figure 6-1, where the calculated range errors $\mathcal{B}_{(j)}$ are replaced with the mean range error per anchor, such that the RLS filter estimates a constant bias. Also, the bias' variance σ^2 was omitted in the online state estimator during the RIBE experiment. The mean performance (line) with one standard deviation (colored area) of the base-case (red) and motion-capture (blue) experiments are plotted for comparison.

Appendix G

Paper Submission (21st IFAC World Congress, 2020)

Iterative Bias Estimation for an Ultra-Wideband Localization System

Bas van der Heijden * **Anton Ledergerber** ** **Rajan Gill** **
Raffaello D'Andrea **

* *Delft Center for Systems and Control, TU Delft, 2628 CN Delft, The Netherlands (e-mail: b.heijden@hotmail.com).*
** *Institute for Dynamic Systems and Control, ETH Zurich, 8092 Zurich, Switzerland (email: {antonl, rgill, rdandrea}@ethz.ch)*

Abstract: An iterative bias estimation framework is presented that mitigates position-dependent ranging errors often present in ultra-wideband localization systems. State estimation and control are integrated, such that the positioning accuracy improves over iterations. The framework is experimentally evaluated on a quadcopter platform, resulting in improvements in the tracking performance with respect to ground truth, and also smoothing the overall flight by significantly reducing unwanted oscillations; see `https://youtu.be/J-htfbzf40U` for a video.

Keywords: Ultra-wideband technology, adaptive observer design, iterative and repetitive control, Bayesian methods, sensor fusion, recursive least squares, classification.

1. INTRODUCTION

Ultra-wideband (UWB) localization systems are one of the enabling technologies for indoor robotics (Alarifi et al. (2016)). Often, the time-of-flight of transmitted UWB radio signals is measured to acquire range measurements for positioning. Under non line-of-sight (NLOS) conditions, these time-of-flight measurements are usually biased. NLOS conditions frequently occur in indoor robotics and the resulting systematic errors in the range measurements limit the positioning accuracy of UWB localization systems, as described in Denis et al. (2003).

Previous works addressed NLOS conditions by building channel classifiers that include additional information, such as floor plans of the environment in Meissner et al. (2010), channel impulse response data of the received UWB radio signal in Schroeder et al. (2007) and references therein, or models trained with labeled data in Maranó et al. (2010). Other works mitigated range errors directly, by using deep learning on channel impulse response data in Tiemann et al. (2017), or using tracking algorithms, while assuming a temporal evolution model for the range error in Denis et al. (2005) and Jourdan et al. (2005).

The scenario considered in this paper is that of an autonomous robotic agent that is tasked to traverse a given reference path repeatedly. The reference path is a parametrized curve, defined as

$$\boldsymbol{\sigma}(\cdot) : [0, \lambda_{\max}] \to \mathbb{R}^3 \text{ with } \lambda_{\max} \in \mathbb{R}^+. \tag{1}$$

The agent moves autonomously, therefore requiring an estimate of its state $\boldsymbol{x}$ to determine the motion control input $\boldsymbol{u}$. The scenario is situated in a densely cluttered indoor environment, causing GPS localization and conventional UWB localization to be inaccurate. In addition to a UWB transceiver, the agent carries an inertial measurement unit (IMU) and barometer, that provide angular rates, accelerations, and altitude measurements. Though the agent is able to roughly traverse the path, the agent is unable to track the reference path accurately due to the aforementioned systematic measurement errors.

Unlike previous works, this paper exploits the repetitive nature of indoor robotic applications by improving the positioning accuracy over iterations. An iterative framework is proposed that combines estimation with control, and can deal with non-static environments because the systematic range error is estimated adaptively. Apart from an IMU and barometer, the proposed framework does not require any additional information such as labeled data (e.g. from a motion-capture system) or floor plans.

The paper is organized as follows. In Section 2, we present experimentally obtained range error data from which a biased range measurement model is derived. In Section 3, the iterative bias estimator framework is presented. Subsequently, the framework is experimentally evaluated on a quadcopter platform in Section 4. We conclude the paper with an outlook in Section 5.

2. RANGE MEASUREMENT ANALYSIS

The UWB localization system considered consists of multiple UWB transceivers placed at known locations $\boldsymbol{p}^a$, which are hereafter called anchors. Each anchor a provides the agent with the time-of-flight of transmitted radio signals, such that the distance towards the anchor can be inferred. The set of range measurements is denoted by $\mathcal{Z}^{\text{uwb}}$. The standard measurement model for such a range measurement $k \in \mathcal{Z}^{\text{uwb}}$ is

$$z_k = \| \boldsymbol{p}^{a_k} - \boldsymbol{p}(t_k) \| + \nu_k^{\text{uwb}}, \tag{2}$$

where z_k, t_k, and a_k denote the measured range, the timestamp and the anchor of the range measurement k, respectively. Furthermore, $\boldsymbol{p}(t_k)$ denotes the agent's position in the global frame at the time the range measurement k

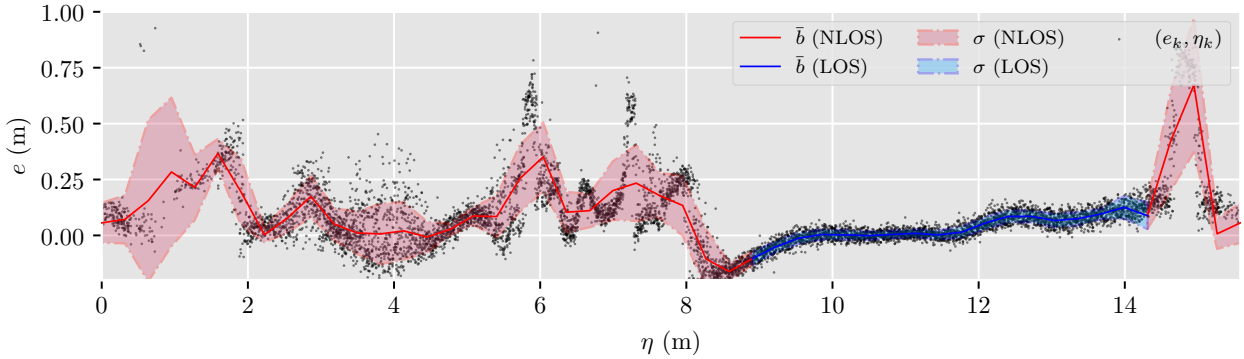

Fig. 1. The range error set of the dashed anchor in Fig. 2, with respect to the path variable η. Two piece-wise linear parametrizations (see (23), (24)) are fitted on the range error set (see (6)), using a least-squares approach with $N^b = 50$. The anchor has both LOS (blue) and NLOS (red) with parts of the path.

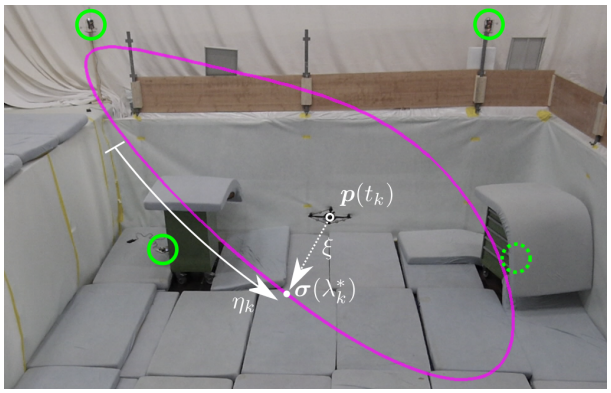

Fig. 2. The state transformation is illustrated for an experimental set-up with a closed reference path $\boldsymbol{\sigma}$ (magenta). The locations of three visible anchors are marked with solid circles. One anchor is behind a metal trolley, and marked with a dashed circle. There are three other anchors not visible in the photo.

was taken. Often, the measurement noise ν_k^{uwb} is assumed to be zero mean, normally distributed white noise.

2.1 Position Dependent Bias

The model in (2) describes the ideal situation. However, in practice range measurements are often inaccurate. Main causes for range errors are propagation delays due to small manufacturing differences in the UWB modules, and NLOS conditions. The former results in a mostly constant error. In contrast, the latter is expected to vary with respect to the geometry of the environment, the anchor placement, and the agent's position. The error in the range measurement $k \in \mathcal{Z}^{\text{uwb}}$ is defined as

$$e_k = z_k - \| \boldsymbol{p}^{a_k} - \boldsymbol{p}(t_k) \|, \tag{3}$$

and is visualized for an example path and anchor in Fig 1. Consider the set-up in Fig. 2. A quadcopter is commanded to traverse the path accurately, for 21 times, using motion-capture data and using a control approach as described in Kumar and Gill (2017). The quadcopter gathers a set of range measurements from the dashed anchor, which has has both line-of-sight (LOS) and NLOS with parts of the path. The motion-capture system provides the true position $\boldsymbol{p}(t_k)$ for all range measurements $k \in \mathcal{Z}^{\text{uwb}}$. The true position is used to calculate the range error e_k in (3), and a path variable η_k. The path variable η_k denotes the arc-length along the path, as visualized in Fig. 2. The path variable is calculated with a state transformation, explained in more detail in Kumar and Gill (2017). First, the closest point to the reference path $\boldsymbol{\sigma}(\lambda_k^*)$ must be calculated with

$$\lambda_k^* = \underset{\lambda \in [0, \lambda_{\max}]}{\arg\min} \ \| \boldsymbol{p}(t_k) - \boldsymbol{\sigma}(\lambda) \|, \tag{4}$$

such that the path variable can be calculated with

$$\eta_k = \int_0^{\lambda_k^*} \left\| \frac{d\boldsymbol{\sigma}(r)}{dr} \right\| dr. \tag{5}$$

Hence, a set of range error e_k and path variable η_k pairs,

$$\mathcal{B} = \{(e_k, \eta_k) \text{ s.t. } k \in \mathcal{Z}^{\text{uwb}}\}, \tag{6}$$

is collected. Fig. 1 shows the subset of $\mathcal{B}$ collected with the dashed anchor in Fig. 2, and illustrates the spatial range error evolution along the path. It is evident that the range error is systematic, caused by NLOS conditions, and dependent on the position along the path. When comparing the variance of the range error for positions along the path, the figure further reveals that there is a position-dependent noise component, in addition to the white noise ν_k^{uwb}.

2.2 Biased Range Measurement Model

The model in (2) is extended with a path-dependent bias $b^{a_k}(\eta_k)$ to capture the position dependency of the range error,

$$z_k = \| \boldsymbol{p}^{a_k} - \boldsymbol{p}(t_k) \| + b^{a_k}(\eta_k) + \nu_k^{\text{uwb}}. \tag{7}$$

As the bias evolution is assumed to be a correlated process with respect to the position along the path, the mean and variance of $b^{a_k}(\eta_k)$ are parametrized as functions of the path variable η_k and a set of bias parameters $\boldsymbol{\theta}^{a,\bar{b}}, \boldsymbol{\theta}^{a,\sigma^2} \in \mathbb{R}^{N^b}$. For ease of notation, the anchor superscript a is dropped in $\boldsymbol{\theta}^{a,\bar{b}}, \boldsymbol{\theta}^{a,\sigma^2}$, but do note that each anchor has its own bias profile. We denote the piece-wise linear parametrizations of its mean and variance by

$$\text{E}[b(\eta_k)] = \bar{b}(\eta_k, \boldsymbol{\theta}^{\bar{b}}) \qquad \text{Var}[b(\eta_k)] = \sigma^2(\eta_k, \boldsymbol{\theta}^{\sigma^2}), \tag{8}$$

and further specify this parametrization in Section 3.3. The number of parameters N^b per anchor is chosen such

that a satisfactory fit on the underlying bias profile is obtained. See Fig. 1 for an example of a fit on experimental data for $N^b = 50$.

3. ITERATIVE BIAS ESTIMATOR

The iterative bias estimator estimates the bias parameters $\boldsymbol{\theta}^{\bar{b}}, \boldsymbol{\theta}^{\sigma^2}$. An attempt was made to assume constant variance, augment the agent's state $\boldsymbol{x}$ with the mean bias parameters $\boldsymbol{\theta}^{\bar{b}}$, and estimate the augmented state in a Kalman filter framework. Though stable configurations were found in simulation, the estimator was prone to diverge in practice. The divergent behavior was attributed to the augmented state being weakly observable, as observability is generally a local property for non-linear systems (Isidori (1995)), and to the model mismatch between the bias parametrization and the actual bias profile.

Instead, the control and estimation architecture in Fig. 3 is proposed. Every timestep, the motion controller receives an estimate of the state $\boldsymbol{x}$ from the online state estimator, which it uses to determine a control input $\boldsymbol{u}$ that forces the agent's position on the path. The agent's sensor measurements are fused in the online state estimator and logged over one iteration j, after which they are sent to the iterative bias estimator, with its components encircled by the dashed line in Fig. 3. The iterative bias estimator relies on a classifier to distinguish reliable (i.e. unbiased) from unreliable range measurements. The iterative framework enables the reliable measurements to be fused in a non-causal state estimator whose state trajectory estimate is used to infer the range error in both the reliable, and unreliable range measurements. The range errors are combined over iterations in a recursive least squares (RLS) filter, allowing the estimation of the bias' variance, in addition to its mean. A forgetting factor in the RLS filter allows the framework to deal with non-static environments. The updated bias parameters $\boldsymbol{\theta}^{\bar{b}}_{(j)}, \boldsymbol{\theta}^{\sigma^2}_{(j)}$ are subsequently used by the online state estimator to evaluate the bias of an anchor for a given position $\boldsymbol{p}$. The set of logged measurements in iteration j is defined as the union

$$\mathcal{Z}_{(j)} = \mathcal{Z}^{\text{uwb}}_{(j)} \cup \mathcal{Z}^{\text{acc}}_{(j)} \cup \mathcal{Z}^{\text{gyr}}_{(j)} \cup \mathcal{Z}^{\text{bar}}_{(j)}, \tag{9}$$

of the range, acceleration, angular rate, and altitude measurement sets, respectively. The components of the iterative bias estimator are specified in the following subsections.

3.1 Classifier

The logged measurements are sent to a classifier. The classifier must partition range measurements $\mathcal{Z}^{\text{uwb}}_{(j)}$ into a reliable partition $\mathcal{Z}^{\text{uwb,rel}}_{(j)}$ and an unreliable partition $\mathcal{Z}^{\text{uwb,unrel}}_{(j)}$. The classification algorithm is an adapted version of the Residual Weighting algorithm presented in Chen (1999), which is based on a least squares (LS) algorithm. The algorithm herein differs in that its task is to classify reliable measurements, instead of estimating the position. Furthermore, the LS algorithm uses the altitude measurements in addition to the range measurements.

When the measurements $\mathcal{Z}_{(j)}$ are received, the range measurements are first partitioned into N^{sub} subsequences, i.e.

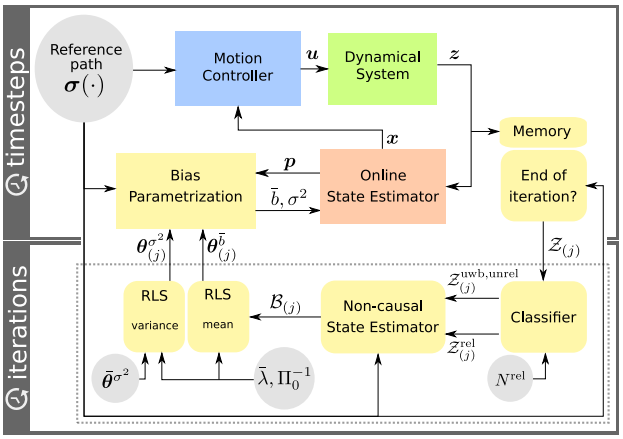

Fig. 3. The iterative bias estimation framework, integrated into a common architecture for controlling the motion of a dynamical system.

$$\mathcal{Z}^{\text{uwb}}_{(j)} = \{\mathcal{Z}^{\text{uwb}}_{(j),1}, \mathcal{Z}^{\text{uwb}}_{(j),2}, \ldots, \mathcal{Z}^{\text{uwb}}_{(j),N^{\text{sub}}}\}. \tag{10}$$

Each subsequence $\mathcal{Z}^{\text{uwb}}_{(j),n} \in \mathcal{Z}^{\text{uwb}}_{(j)}$ contains consecutive range measurements obtained with different anchors. Hence, if N^a is the number of anchors from which sequential range measurements are obtained, such a subsequence is of size $|\mathcal{Z}^{\text{uwb}}_{(j),n}| = N^a$. The classifier labels N^{rel} range measurements per subsequence $\mathcal{Z}^{\text{uwb}}_{(j),n}$ as reliable, where N^{rel} is set to the expected number of anchors with LOS at any given time. Reliable range measurements are consistent with each other, which is reflected in a low residual of a LS position estimate given reliable range measurements. Therefore, a LS position estimate is calculated for different range measurement combinations $\mathcal{C}^i_{(j),n} \subset \mathcal{Z}^{\text{uwb}}_{(j),n}$, $|\mathcal{C}^i_{(j),n}| = N^{\text{rel}}$. These combination are obtained by taking all $|\mathcal{Z}^{\text{uwb}}_{(j),n}|$ choose N^{rel} different combinations, resulting in N^{c} combinations per subsequence. The combination $\mathcal{C}^i_{(j),n}$ that results in the lowest LS position estimate residual $r^i_{(j),n}$ is classified as reliable, i.e.

$$\mathcal{Z}^{\text{uwb,rel}}_{(j),n} = \mathcal{C}^i_{(j),n} \text{ s.t. } i = \underset{i \in 1,2,\ldots,N^{\text{c}}}{\arg\min}\ r^i_{(j),n}. \tag{11}$$

The altitude measurements $l \in \mathcal{Z}^{\text{bar}}_{(j),n}$, measured within the time spanned by the range measurement timestamps of $\mathcal{Z}^{\text{uwb}}_{(j),n}$, are additionally fused into the LS estimate. The residual of such a LS estimate is given by

$$r^i_{(j),n} = \min_{\boldsymbol{p} \in \mathbb{R}^3} \left(r^{i,\text{uwb}}_{(j),n}(\boldsymbol{p}) + r^{\text{bar}}_{(j),n}(p_{\text{z}}) \right), \tag{12}$$

where p_{z} is the z-coordinate of the position $\boldsymbol{p}$, and

$$r^{i,\text{uwb}}_{(j),n}(\boldsymbol{p}) = \frac{1}{\Sigma^{\text{uwb}}} \sum_{k \in \mathcal{C}^i_{(j),n}} (z_k - \| \boldsymbol{p}^{a_k} - \boldsymbol{p} \|)^2, \tag{13}$$

$$r^{\text{bar}}_{(j),n}(p_z) = \frac{1}{\Sigma^{\text{bar}}} \sum_{l \in \mathcal{Z}^{\text{bar}}_{(j),n}} (z_l - p_z)^2. \tag{14}$$

where Σ^{uwb} and Σ^{bar} are the corresponding measurement noise variances. Then, the set of reliable range measurements over a complete iteration is given as the union of the combinations with the lowest residuals, i.e.

$$\mathcal{Z}^{\text{uwb,rel}}_{(j)} = \{\mathcal{Z}^{\text{uwb,rel}}_{(j),1}, \mathcal{Z}^{\text{uwb,rel}}_{(j),2}, \ldots, \mathcal{Z}^{\text{uwb,rel}}_{(j),N^{\text{sub}}}\}, \tag{15}$$

and the unreliable set as the complement to $\mathcal{Z}_{(j)}^{\text{uwb,rel}}$, i.e.

$$\mathcal{Z}_{(j)}^{\text{uwb,unrel}} = \mathcal{Z}_{(j)}^{\text{uwb}} \setminus \mathcal{Z}_{(j)}^{\text{uwb,rel}}. \tag{16}$$

3.2 Non-causal Estimator

For each iteration j, the non-causal estimator estimates the range error set $\mathcal{B}_{(j)}$ defined in (6). For this, it requires estimates of the position at every time $t_k, k \in \mathcal{Z}_{(j)}^{\text{uwb}}$. An adapted version of the continuous-time batch optimization framework presented in Furgale et al. (2015) is used, as follows.

The measurement set $\mathcal{Z}_{(j)}^{\text{rel}} = \{\mathcal{Z}_{(j)}^{\text{uwb,rel}}, \mathcal{Z}_{(j)}^{\text{acc}}, \mathcal{Z}_{(j)}^{\text{gyr}}, \mathcal{Z}_{(j)}^{\text{bar}}\}$ is used to find the maximum a-posteriori estimate of the state trajectory $\boldsymbol{x}(t, \boldsymbol{c})$, expressed as the weighted sum of a finite set of temporal basis functions with parameters $\boldsymbol{c}$. It is assumed that the position trajectory $\boldsymbol{p}(t)$ can be inferred from the state trajectory $\boldsymbol{x}(t, \boldsymbol{c})$. Using Bayes' law, the posterior estimate can be written as

$$p(\boldsymbol{x}(t, \boldsymbol{c})|\mathcal{Z}_{(j)}^{\text{rel}}) = \frac{p(\boldsymbol{x}(t, \boldsymbol{c}))}{p(\mathcal{Z}_{(j)}^{\text{rel}})} p(\mathcal{Z}_{(j)}^{\text{rel}}|\boldsymbol{x}(t, \boldsymbol{c})). \tag{17}$$

Assuming that the sensor measurements are conditionally independent given $\boldsymbol{x}(t, \boldsymbol{c})$, the posterior is rewritten as

$$p(\boldsymbol{x}(t, \boldsymbol{c})|\mathcal{Z}_{(j)}^{\text{rel}}) = \frac{p(\boldsymbol{x}(t, \boldsymbol{c}))}{p(\mathcal{Z}_{(j)}^{\text{rel}})} \prod_{m \in \mathcal{Z}_{(j)}^{\text{rel}}} p(z_m|\boldsymbol{x}(t_m, \boldsymbol{c})). \tag{18}$$

The measurement likelihood for an arbitrary sensor measurement m with a Gaussian distribution is given by

$$p(z_m|\boldsymbol{x}(t_m, \boldsymbol{c})) = \mathcal{N}(h_m(\boldsymbol{x}(t_m, \boldsymbol{c})), \Sigma_m), \tag{19}$$

where the measurement model $h_m(\boldsymbol{x}(t_m, \boldsymbol{c}))$ denotes the mean of z_m, and Σ_m denotes the noise covariance matrix. Only reliable range measurements are used in the non-causal estimator. Nevertheless, small differences in the antennas cause a constant offset o_a^{uwb}, slightly different for each anchor. To compensate for o_a^{uwb}, the model

$$h_k(\boldsymbol{x}(t_k, \boldsymbol{c}), o_{a_k}^{\text{uwb}}) = \parallel \boldsymbol{p}^{a_k} - \boldsymbol{p}(\boldsymbol{x}(t_k, \boldsymbol{c})) \parallel + o_{a_k}^{\text{uwb}} \tag{20}$$

is used for the reliable range measurements $k \in \mathcal{Z}_{(j)}^{\text{uwb,rel}}$. Similarly, slow drifting behavior in the barometer can occur, and causes an offset o^{bar} in the altitude measurements that is approximately constant over one iteration. Therefore, a constant offset o^{bar} is estimated each iteration by assuming the barometer measurement model

$$h_l(\boldsymbol{x}(t_l, \boldsymbol{c}), o^{\text{bar}}) = p_z(\boldsymbol{x}(t_l, \boldsymbol{c})) + o^{\text{bar}} \tag{21}$$

for the altitude measurements $l \in \mathcal{Z}_{(j)}^{\text{bar}}$. See Ledergerber and D'Andrea (2018) for the measurement models of a biased IMU, with the state modeled as a rigid body. In case the IMU is unbiased, the bias terms in the models can be omitted. Then, the maximum a-posteriori estimates $({}^*\boldsymbol{c}, {}^*o_a^{\text{uwb}}, {}^*o^{\text{bar}})$ are found by minimizing the negative logarithm of the posterior (18),

$$\min_{\boldsymbol{c}, o_a^{\text{uwb}}, o^{\text{bar}}} \sum_{m \in \mathcal{Z}_{(j)}^{\text{rel}}} -\log(p(z_m|\boldsymbol{x}(t_m, \boldsymbol{c}), o_a^{\text{uwb}}, o^{\text{bar}})), \tag{22}$$

where the term $p(\mathcal{Z}_{(j)}^{\text{rel}})$ is omitted, as it does not influence the optimized solution, and the prior $p(\boldsymbol{x}(t, \boldsymbol{c}))$ is omitted because the ratio of measurements to optimization variables is large. This reduces (22) to a maximum-likelihood problem (Furgale et al. (2015)). A local optimization is performed, where the estimates of the online state estimator serve as the initial guess for the optimization. With the resulting position trajectory $\boldsymbol{p}(t, {}^*\boldsymbol{c})$, an estimate of the range error set $\mathcal{B}_{(j)}$ is calculated with (4), (5), and (6). It is important to note here, that the range error is calculated for all range measurements, i.e. for measurements classified as reliable and for measurements classified as unreliable. The barometer offset ${}^*o^{\text{bar}}$ is provided to the classifier and online state estimator, such that the drift is compensated for in the altitude measurements $z_l, l \in \mathcal{Z}^{\text{bar}}$ before being incorporated in (14) and before being fused in the online state estimator.

3.3 Recursive Least Squares Filter

The bias parameters $\boldsymbol{\theta}_{(j)}^{\bar{b}}, \boldsymbol{\theta}_{(j)}^{\sigma^2}$ are estimated recursively by combining the range error estimates $\mathcal{B}_{(j)}$ with the sets of range error estimates from all previous iterations, i.e. $\mathcal{B}_{(1..j)} = \cup_{i=1}^{j} \mathcal{B}_{(i)}$, yielding the two piece-wise linear bias parametrizations introduced in (8) and defined as

$$\bar{b}(\eta_k, \boldsymbol{\theta}_{(j)}^{\bar{b}}) = \boldsymbol{u}^{\mathsf{T}}(\eta_k)\boldsymbol{\theta}_{(j)}^{\bar{b}} \tag{23}$$

$$\sigma^2(\eta_k, \boldsymbol{\theta}_{(j)}^{\sigma^2}) = \boldsymbol{u}^{\mathsf{T}}(\eta_k)\boldsymbol{\theta}_{(j)}^{\sigma^2}. \tag{24}$$

The elements of vector

$$\boldsymbol{u}^{\mathsf{T}}(\eta_k) = [u_1(\eta_k), u_2(\eta_k), \ldots, u_{N^b}(\eta_k)] \tag{25}$$

are zero, except for

$$u_i(\eta_k) = 1 - \frac{\eta_k - \tau_i}{\tau_{i+1} - \tau_i} \qquad u_{i+1}(\eta_k) = \frac{\eta_k - \tau_i}{\tau_{i+1} - \tau_i}, \tag{26}$$

where i is s.t. $\tau_i \leq \eta_k < \tau_{i+1}$, and where the knots τ_i of the piece-wise linear parameterization are given as

$$\tau_i = \frac{\eta_{\max}}{N^b - 1}(i - 1) \text{ for } i = \{1, \ldots, N^b\}, \tag{27}$$

where $\eta_{\max}$ is the arc-length of the reference path. To solve for the mean bias parameters $\boldsymbol{\theta}_{(j)}^{\bar{b}}$, a regressor matrix $U \in \mathbb{R}^{N^b \times |\mathcal{B}_{(1..j)}|}$ and range error vector $\boldsymbol{e} \in \mathbb{R}^{|\mathcal{B}_{(1..j)}|}$ are constructed with the set $\mathcal{B}_{(1..j)}$ of all range error, path variable pairs up to iteration j, i.e.,

$$U(\mathcal{B}_{(1..j)}) = [\boldsymbol{u}(\eta_1), \boldsymbol{u}(\eta_2), ..., \boldsymbol{u}(\eta_{|\mathcal{B}_{(1..j)}|})] \tag{28}$$

$$\boldsymbol{e}(\mathcal{B}_{(1..j)}) = [e_1, e_2, ..., e_{|\mathcal{B}_{(1..j)}|}]^{\mathsf{T}}. \tag{29}$$

Subsequently, the problem of finding the mean bias parameters $\boldsymbol{\theta}_{(j)}^{\bar{b}}$ can be formulated as a least squares problem,

$$\min_{\boldsymbol{\theta}_{(j)}^{\bar{b}}} \left[(\boldsymbol{\theta}_{(j)}^{\bar{b}} - \bar{\boldsymbol{\theta}}^{\bar{b}})^{\mathsf{T}} \Pi_0^{-1} (\boldsymbol{\theta}_{(j)}^{\bar{b}} - \bar{\boldsymbol{\theta}}^{\bar{b}}) + \parallel \boldsymbol{e} - U^{\mathsf{T}} \boldsymbol{\theta}_{(j)}^{\bar{b}} \parallel^2 \right], \tag{30}$$

where the initial guess $\bar{\boldsymbol{\theta}}^{\bar{b}}$ and the weighting matrix Π_0^{-1} represent the prior knowledge. Each iteration, $\boldsymbol{\theta}_{(j)}^{\bar{b}}$ is estimated by solving (30) recursively with the regularized RLS algorithm described in Section 21.4 of Sayed and Kailath (2000). A forgetting factor $\bar{\lambda}$ is included in the algorithm, so that the framework is adaptive to non-static environments. Similarly, the variance bias parameters $\boldsymbol{\theta}_{(j)}^{\sigma^2}$ are estimated, by recursively solving

$$\min_{\boldsymbol{\theta}_{(j)}^{\sigma^2}} \left[(\boldsymbol{\theta}_{(j)}^{\sigma^2} - \bar{\boldsymbol{\theta}}^{\sigma^2})^{\mathsf{T}} \Pi_0^{-1} (\boldsymbol{\theta}_{(j)}^{\sigma^2} - \bar{\boldsymbol{\theta}}^{\sigma^2}) + \parallel \boldsymbol{r} - U^{\mathsf{T}} \boldsymbol{\theta}_{(j)}^{\sigma^2} \parallel^2 \right], \tag{31}$$

where $\bar{\boldsymbol{\theta}}^{\sigma^2}$ is the initial guess for the variance. In practice, $\bar{\boldsymbol{\theta}}^{\sigma^2}$ is set to a large value, while $\bar{\boldsymbol{\theta}}^{\bar{b}}$ is initialized to zero, as

the shape of the bias profile is unknown. Vector $\boldsymbol{r}$ denotes the squared residual between the range error data and mean bias fit. The squared residual is a common metric for estimating variances, as described in Buckley et al. (1988), and is defined as

$$\boldsymbol{r} = (\boldsymbol{e} - U^{\mathsf{T}}\boldsymbol{\theta}^{\bar{b}}_{(j)}) \odot (\boldsymbol{e} - U^{\mathsf{T}}\boldsymbol{\theta}^{\bar{b}}_{(j)}), \tag{32}$$

where $\odot$ is the element-wise multiplication operator. The estimated parameters $\boldsymbol{\theta}^{\bar{b}}_{(j)}, \boldsymbol{\theta}^{\sigma^2}_{(j)}$ are provided to the bias parametrization block in Fig. 3, such that the online state estimator can evaluate the mean and variance bias parametrizations in (23), and (24) for a given position $\boldsymbol{p}$, using (4), and (5).

4. EXPERIMENTAL EVALUATION ON A QUADCOPTER PLATFORM

The proposed iterative bias estimation framework was experimentally evaluated on a quadcopter platform. The quadcopter is the autonomous robotic agent, tasked to fly a closed reference path repeatedly. First, the implementation details are provided, showing how the framework was able to run in real-time. Subsequently, the experimental results are presented.

4.1 Implementation

The motion controller and online state estimator are run on a Snapdragon flight board, located onboard a quadcopter. The iterative bias estimator is run in a separate process, offboard on a laptop with an i7-Intel processor. The onboard and offboard processes communicate with a TCP client/server set-up. Both the anchors and quadcopter are equipped with DWM 1000 modules, and communicate using the two-way ranging algorithm with repeated reply described in Mueller et al. (2015). The quadcopter communicates with every anchor in a sequential order at a frequency of 200 Hz. The MPU9250 IMU and BMP280 barometer modules are integrated on the Snapdragon flight board, and provide measurements with a frequency of 1 kHz and 50 Hz, respectively.

The online state estimator estimates the position, velocity, orientation representation, and angular velocity of the quadcopter in a Kalman filter framework, specified in Ledergerber and D'Andrea (2018). The range measurements are fused using model (7) via an unscented Kalman filter. For every position sigma point $\boldsymbol{p}^{(i)}$, the resulting $(\bar{b}, \sigma^2)^{(i)}$ can be incorporated into the range measurement update.

A path following controller is implemented as the motion controller of the quadcopter. The controller minimizes the agent's distance towards the path and makes the agent traverse along the path with a constant reference speed of 1.5 m/s. See Kumar and Gill (2017) for more details.

It is assumed that the quadcopter has LOS with 5 out of 7 anchors at any given time. Therefore, N^{rel} is set to 5, and $N^c = 21$ for $|\mathcal{Z}^{\text{uwb}}_{(j),n}| = 7$. For an iteration j with a duration of 10 seconds, roughly $|\mathcal{Z}^{\text{uwb}}_{(j)}| \approx 2000$ range measurements are logged, resulting in $N^{\text{sub}} \approx |\mathcal{Z}^{\text{uwb}}_{(j)}|/N^a \approx 286$ partitions, as defined in (10). For every partition, (12) must be evaluated for all $N^c = 21$ combinations, resulting in a total of $286 * 21 = 6006$ computations of (12). All computations are independent from one another, so that the computations can be parallelized.

The non-causal state estimator is implemented similarly to Ledergerber and D'Andrea (2018). The state trajectory $\boldsymbol{x}(t)$ is assumed to consist of a position trajectory $\boldsymbol{p}(t)$ and quaternion trajectory $\boldsymbol{q}(t)$. Both trajectories are parametrized by a set of uniform cubic B-splines. The knot spacing for the splines is set to roughly 500 ms. The IMU is assumed to be unbiased, because it showed no drifting behavior after calibrating prior to flying.

The RLS filter has a forgetting factor $\bar{\lambda} = 0.9$, which means that the estimated parameters $\boldsymbol{\theta}^{\bar{b}}_{(j)}, \boldsymbol{\theta}^{\sigma^2}_{(j)}$ depend on roughly the last 10 iterations. A large uncertainty in the bias is encoded by setting each element in the initial variance $\bar{\boldsymbol{\theta}}^{\sigma^2}$ to a large value of 0.15^2. The weight matrix is set to be $\Pi_0^{-1} = UU^{\mathsf{T}}$, where U is constructed with (28) using N^{sub} uniformly sampled path variables η along the path. The reference path is a closed path, so the parametrization in (25), (26), and (27) is adjusted accordingly.

4.2 Experimental Results

The experiments were performed in the Flying Machine Arena of ETH Zurich (Lupashin et al. (2014)). See Fig. 2 for the anchor placement and reference path the quadcopter is tasked to fly. Objects (e.g. metal trolleys) placed in the room cause some anchors to have NLOS with parts of the path, inducing a position-dependent bias in the range measurements. See Fig. 1 for an example.

Minimizing the distance towards the path is considered to be the most important control objective, because it ensures a safe flight. This distance is referred to as the tracking error, and is defined as

$$\xi = \| \boldsymbol{p} - \boldsymbol{\sigma}(\lambda^*) \| . \tag{33}$$

The first time-derivative of the tracking error is denoted by $\dot{\xi}$. A large $\dot{\xi}$ means the distance error varies rapidly, which expresses itself in an oscillatory flight around the reference path. The mean absolute error (MAE) in ξ and $\dot{\xi}$ is shown in Fig. 4 for four runs. The performance of the proposed framework is compared with that of two different experiments. In a base-case (base) experiment, no bias estimation is performed. In a motion-capture (mc) experiment, the true range error set $\mathcal{B}$ is obtained with a motion-capture system, and subsequently fed to the RLS filter such that the best-possible bias fit is obtained. The performance of the motion-capture experiment is interpreted as the best case. Each experiment is executed multiple times. Only the mean performance μ with one standard deviation σ is plotted for the base-case and motion-capture experiments.

Fig. 4 shows that the proposed framework consistently improves performance over iterations in both ξ, and $\dot{\xi}$, with respect to the base-case experiments. Concerning $\dot{\xi}$, a similar performance level with respect to the motion-capture experiment is obtained. The improvements in $\dot{\xi}$ are ascribed to the inclusion of a position-dependent noise component in addition to the mean bias, and results in reduced oscillations during flight. See `https://youtu.be/J-htfbzf40U` for a video, showcasing the improved flight performance. The performance of the iterative bias

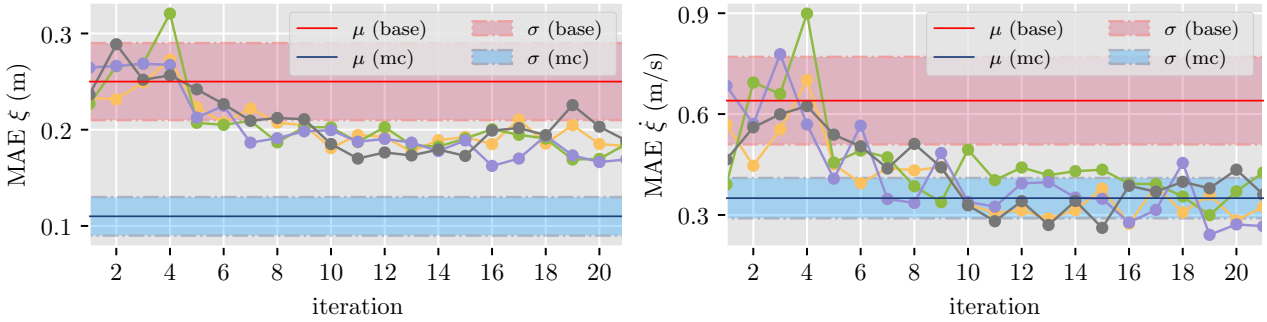

Fig. 4. The path tracking performance of the proposed framework is plotted for four runs (solid-dotted), alongside the mean performance μ with one standard deviation σ of the base-case (red) and motion-capture (blue) experiments.

estimation framework largely depends on the ability of the classifier to correctly distinguish reliable from unreliable measurements. Therefore, the difference in path tracking performance improvement between the framework (26%), and best case (56%), is primarily attributed to the sub-optimal performance of the classifier.

5. OUTLOOK

Improvements in the classifier are expected to be most effective in improving the overall performance of the framework. For example, the additional inclusion of IMU measurements into the presented classification scheme is worth investigating. Also, completely different classification schemes based on channel impulse response data should be tested and could lead to a reduction of the classifier's computational complexity. Finally, it would be interesting to apply iterative learning control on top of the presented framework for high-performance maneuvering under UWB.

ACKNOWLEDGEMENTS

The authors would like to thank Marc-Andrè Corzillius for the design of the electrical hardware and Michael Egli for construction of the platform. Many people have contributed to the Flying Machine Arena in which this research was conducted. A list of these people can be found at: `http://flyingmachinearena.org/people`.

REFERENCES

Alarifi, A., Al-Salman, A., Alsaleh, M., Alnafessah, A., Al-Hadhrami, S., Al-Ammar, M., and Al-Khalifa, H. (2016). Ultra wideband indoor positioning technologies: Analysis and recent advances. *Sensors*, 16(707).

Buckley, M., Eagleson, G., and Silverman, B. (1988). The estimation of residual variance in nonparametric regression. *Biometrika*, 75, 189–199.

Chen, P. (1999). A non-line-of-sight error mitigation algorithm in location estimation. *IEEE Wireless Communications and Networking Conference*, 1, 316–320.

Denis, B., Keignart, J., and Daniele, N. (2003). Impact of nlos propagation upon ranging precision in uwb systems. In *IEEE Conference on Ultra Wideband Systems and Technologies, 2003*, 379–383.

Denis, B., Ouvry, L., Uguen, B., and Tchoffo-Talom, F. (2005). Advanced bayesian filtering techniques for uwb tracking systems in indoor environments. In *2005 IEEE International Conference on Ultra-Wideband*, 638–643.

Furgale, P., Tong, C., Barfoot, T., and Shin, S. (2015). Continuous-time batch trajectory estimation using temporal basis functions. *International Journal of Robotics Research*, 34(14), 1688–17101.

Isidori, A. (1995). *Nonlinear control systems*. Springer-Verlag.

Jourdan, D.B., Deyst, J.J., Win, M.Z., and Roy, N. (2005). Monte carlo localization in dense multipath environments using uwb ranging. In *2005 IEEE International Conference on Ultra-Wideband*, 314–319.

Kumar, S. and Gill, R. (2017). Path following for quadrotors. *IEEE Conference on Control Technology and Applications*, 2075–2081.

Ledergerber, A. and D'Andrea, R. (2018). Calibrating away inaccuracies in uwb range measurements: A maximum likelihood approach. *IEEE Access*, 6, 78719–78730.

Lupashin, S., Hehn, M., Mueller, M., Schoellig, A., Sherback, M., and D'Andrea, R. (2014). A platform for aerial robotics research and demonstration: The flying machine arena. *Mechatronics*, 24(1), 41–54.

Maranó, S., Gifford, W.M., Wymeersch, H., and Win, M.Z. (2010). Nlos identification and mitigation for localization based on uwb experimental data. *IEEE Journal on Selected Areas in Communications*, 28(7), 1026–1035.

Meissner, P., Steiner, C., and Witrisal, K. (2010). Uwb positioning with virtual anchors and floor plan information. In *2010 7th Workshop on Positioning, Navigation and Communication*, 150–156.

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

Sayed, A. and Kailath, T. (2000). Recursive least-squares adaptive filters. In *Digital Signal Processing Handbook*, chapter 21. CRC press LLC.

Schroeder, J., Galler, S., Kyamakya, K., and Jobmann, K. (2007). Nlos detection algorithms for ultra-wideband localization. In *2007 4th Workshop on Positioning, Navigation and Communication*, 159–166.

Tiemann, J., Pillmann, J., and Wietfeld, C. (2017). Ultra-wideband antenna-induced error prediction using deep learning on channel response data. In *2017 IEEE 85th Vehicular Technology Conference*, 1–5.

Bibliography

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] A. Alarifi, A. Al-Salman, M. Alsaleh, A. Alnafessah, S. Al-Hadhrami, M.A. Al-Ammar, and H.S. Al-Khalifa. Ultra wideband indoor positioning technologies: analysis and recent advances. *MDPI Sensors*, 2016.

[3] M.N. Berberan-Santos, E.N. Bodunov, and L. Pogliani. On the barometric formula. *American Journal of Physics*, 65(5):404–412, 1997.

[4] M.J. Buckley, G.K. Eagleson, and B.W. Silverman. The estimation of residual variance in nonparametric regression. *Biometrika*, 75:189–199, 1988.

[5] P.C. Chen. A non-line-of-sight error mitigation algorithm in location estimation. *IEEE Wireless Communications and Networking Conference (WCNC)*, 1:316–320, 1999.

[6] B. Denis, J. Keignart, and N. Daniele. Impact of nlos propagation upon ranging precision in uwb systems. In *IEEE Conference on Ultra Wideband Systems and Technologies, 2003*, pages 379–383, 2003.

[7] B. Denis, L. Ouvry, B. Uguen, and F. Tchoffo-Talom. Advanced bayesian filtering techniques for uwb tracking systems in indoor environments. In *2005 IEEE International Conference on Ultra-Wideband*, pages 638–643, 2005.

[8] P. Furgale, C.H. Tong, T.D. Barfoot, and S.Y. Shin. Continuous-time batch trajectory estimation using temporal basis functions. *International Journal of Robotics Research*, 34(14):1688–17101, 2015.

[9] E. Höckerdal, E.F. Frisk, and L. Eriksson. Observer design and model augmentation for bias compensation with a truck engine application. *Control Engineering Practice*, 17(3):408 – 417, 2009.

[10] D. B. Jourdan, J. J. Deyst, M. Z. Win, and N. Roy. Monte carlo localization in dense multipath environments using uwb ranging. In *2005 IEEE International Conference on Ultra-Wideband*, pages 314–319, 2005.

[11] S. Kumar and R. Gill. Path following for quadrotors. *IEEE Conference on Control Technology and Applications (CCTA)*, pages 2075–2081, 2017.

[12] A. Ledergerber and R. D'Andrea. Calibrating away inaccuracies in ultra wideband range measurements: A maximum likelihood approach. *IEEE Access*, 6:78719–78730, 2018.

[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*, 24(1):41–54, 2014.

[14] R. De Maesschalck, D. Jouan-Rimbaud, and D.L. Massart. The mahalanobis distance. *Chemometrics and intelligent laboratory systems*, 50(1):1–18, 2000.

[15] S. Maranó, W. M. Gifford, H. Wymeersch, and M. Z. Win. Nlos identification and mitigation for localization based on uwb experimental data. *IEEE Journal on Selected Areas in Communications*, 28(7):1026–1035, 2010.

[16] P. Meissner, C. Steiner, and K. Witrisal. Uwb positioning with virtual anchors and floor plan information. In *2010 7th Workshop on Positioning, Navigation and Communication*, pages 150–156, 2010.

[17] N. Metni, J.M. Pflimlin, T. Hamel, and P. Souères. Attitude and gyro bias estimation for a vtol uav. *Control Engineering Practice*, 14(12):1511 –1520, 2006.

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

[19] H. Nijmeijer and A. van der Schaft. In *Nonlinear Dynamical Control Systems.*, chapter 3. Springer-Verlag, 1990.

[20] A.H. Sayed and T. Kailath. Recursive least-squares adaptive filters. In *Digital Signal Processing Handbook*, chapter 21. CRC press LLC, 2000.

[21] J. Schroeder, S. Galler, K. Kyamakya, and K. Jobmann. Nlos detection algorithms for ultra-wideband localization. In *2007 4th Workshop on Positioning, Navigation and Communication*, pages 159–166, 2007.

[22] D. Simon. *Optimal state estimation: Kalman, H∞, and nonlinear approaches.* John Wiley & Sons, 2006.

[23] J. Tiemann, J. Pillmann, and C. Wietfeld. Ultra-wideband antenna-induced error prediction using deep learning on channel response data. In *2017 IEEE 85th Vehicular Technology Conference*, pages 1–5, 2017.

Glossary

List of Acronyms

IDSC	Institute for Dynamic Systems and Control
UWB	ultra-wideband
LOS	line-of-sight
NLOS	non line-of-sight
IMU	inertial measurement unit
EKF	extended Kalman filter
UKF	unscented Kalman filter
MAE	mean absolute error
RLS	recursive least squares
LS	least squares
MD	Mahalanobis distance
SBE	sequential bias estimator
IBE	iterative bias estimator
RIBE	reliable iterative bias estimator
PFC	path following controller
TFL	transverse feedback linearization
FSF	Frenet-Serret frames
LTI	linear time-invariant
IFAC	International Federation for Automatic Control

List of Symbols

$\bar{\boldsymbol{\theta}}^{\bar{b}}, \boldsymbol{\theta}_0^{\bar{b}}$	Initial mean bias parameters
$\bar{\boldsymbol{\theta}}^{\sigma^2}$	Initial variance bias parameters
$\bar{\lambda}$	Forgetting factor
$\bar{b}$	Bias' mean
$\bar{y}$	Expected measurement
$\boldsymbol{\omega}$	Angular velocity
$\boldsymbol{a}$	Agent's acceleration in body frame
$\boldsymbol{c}_b$	Temporal basis function bias parameters
$\boldsymbol{c}_x$	Temporal basis function state parameters
$\boldsymbol{g}$	Gravitational acceleration
$\boldsymbol{h}, H$	Measurement model/regressor vector, regressor matrix
$\boldsymbol{q}$	Quaternion/Euler angle attitude representation
$\boldsymbol{u}$	Control input
$\boldsymbol{z}$	Sensor measurement
$\boldsymbol{\theta}^{\bar{b}}$	Mean bias parameters
$\boldsymbol{p}, p_x, p_y, p_z$	Agent's position
$\boldsymbol{p}^a$	Anchor position
$\boldsymbol{v}$	Velocity
$\boldsymbol{\theta}^{\sigma^2}$	Variance bias parameters
$\boldsymbol{x}, \tilde{\boldsymbol{x}}$	Agent's state, augmented state
$\boldsymbol{x}^+, P^+$	A-posteriori state and covariance matrix
$\boldsymbol{x}^-, \tilde{\boldsymbol{x}}^-$	Prior (augmented) state
$\boldsymbol{x}_0, \tilde{\boldsymbol{x}}_0$	Initial (augmented) state
$\mathcal{B}$	Range error set
$\mathcal{Z}$	Index set of measurements
$\mathcal{Z}^{\text{rel}}, \mathcal{Z}^{\text{unrel}}$	Reliable and unreliable set of measurements
$\dot{\eta}_{\text{ref}}$	Tangential reference speed
$\eta, \eta_{\max}$	Path variable, arc-length of path
(j)	Iteration index
$\lambda, \lambda^*, \lambda_{\max}$	Curve variable, closest point to path, max curve variable
$\mathcal{C}$	Range measurement combination
ν	Normally distributed white noise
$\boldsymbol{\sigma}$	Reference path
$\Phi, \boldsymbol{S}$	Memory variables of the recursive least squares (RLS) filter
Π_0^{-1}	Positive weighting matrix
Σ	Sensor noise covariance matrix
σ^2	Bias' variance
τ	Piece-wise linear knot location

$\xi, \dot{\xi}$	Transversal distance (or path tracking error) and transversal velocity
b	Path-dependent bias
C	Number of harmonics
$e, \boldsymbol{e}$	Range error, range error vector
L	Covariance matrix square root
m, k, n, w, l	General, range, acceleration, angular rate, altitude index
n	State dimension
N^{rel}	Number of measurements with line-of-sight (LOS) at any given time
N^{sub}	Number of partitions/expected number of measurements per anchor, collected during one iteration
N^a	Number of anchors
N^b	Number of bias parameters per anchor
N^c	Number of parameters to express the bias trajectory
N^x	Number of parameters to express the state trajectory
$o^{\text{bar}}, o_a^{\text{uwb}}$	Altitude offset, range offset
$P, \tilde{P}$	Covariance matrix, augmented covariance matrix
$P^-, \tilde{P}^-$	Prior (augmented) covariance matrix
$P_0, \tilde{P}_0$	Initial (augmented) covariance matrix
$P_{\text{xy}}, \tilde{P}_{\text{xy}}$	Cross covariance matrix
$P_{\text{y}}, \tilde{P}_{\text{y}}$	Measurement covariance matrix
$Q, \tilde{Q}$	Process noise covariance matrix
$R(\boldsymbol{\delta}), R(\boldsymbol{q})$	Attitude representation
$r, \boldsymbol{r}$	Squared residual, squared residual vector
${}^*\boldsymbol{c}_x, {}^*\boldsymbol{c}_b$	Maximum a-posteriori parameter estimate
${}^*o^{\text{bar}}, {}^*o_a^{\text{uwb}}$	Maximum a-posteriori offset estimate