State Estimation for Legged Robots

Bachelor Thesis

Author(s):
Hoffman, Benjamin

Publication date:
2021-07

Permanent link:
https://doi.org/https://doi.org/10.3929/ethz-b-000713411

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

State Estimation for Legged Robots

Bachelor Thesis

Computational Robotics Lab
Swiss Federal Institute of Technology (ETH) Zurich

Author

Benjamin Hoffman
18-928-580
bhoffman@student.ethz.ch

Supervision

Mr. Dongho Kang
Prof. Dr. Stelian Coros

July 2021

Contents

Acknowledgements

My great interest for robotics and control discovered during my studies at ETH drove me to pursue a bachelor thesis project within this field with great excitement. By the immersion into the world of legged robotics and estimation, I have gained valuable knowledge within these fields, as well as new, relevant technical skills. As much as this journey helped me discover my own strategies for dealing with arising difficulties and new challenges, I could not have completed it on my own. Throughout the process of this thesis project, I have received a great amount of guidance, advice and assistance. I want to express my profound appreciation for the support that has been provided to me.

Most of all, I would like to thank my supervisor Mr. Dongho Kang for his continuous and tireless support throughout the process of this thesis. In research, implementation and finally in the presentation of my work, Dongho provided me with the indispensable insight and technical expertise necessary for me to tackle the hurdles faced during this project. Working with Dongho and learning from him has been an amazing experience, deeply enriching my time spent at ETH and on this project. Further, I would like to thank Prof. Stelian Coros for providing me the opportunity, as well as the means necessary to do my thesis work at the CRL at ETH.

Also, I would like to thank Mr. Robin Schmid who, during his work as a Research Assistant at the CRL at ETH, implemented the Extended Kalman Filter, which I used during this project.

Abstract

In the advent of the omnipresence of robotics in our day to day lives, progressions are made in this field continuously. In particular, the recent developments achieved within the area of legged robotics substantiate their growing importance for present and future applications. Robust and accurate state estimation capabilities are essential to the feedback control that enables legged robots to function within their expanding range of deployment. This performance is achieved by employing sensor fusion frameworks that combine multiple measurements to produce an estimate of the robots state. The approaches discussed in this thesis focus on using information provided by proprioceptive sensors to maintain robustness and fast estimation times. Further, the approaches employ sensor fusion by means of stochastic Kalman Filtering methods. The challenges presented by state estimation for legged robots however lead to estimation approaches that entail ever increasing implementation complexity to guarantee the accuracy of the estimate. Further, the amount of research being done within this field steadily increases the different state estimation techniques available for legged robots. While substantial progress is being made, not every increase in complexity is justified by its increase in accuracy.

The significance of the robot's estimation capability emphasizes the need for constant review and comparison between multiple state-of-the-art estimation approaches. This is further asserted by the multitude of techniques proposed, as well as by the tendency to higher implementation complexity.

Building on the analysis of previous work, this thesis aims at implementing, combining and comparing different state-of-the-art state estimation approaches for legged robots. The implementations are tested in simulation and the results are evaluated. An emphasis is placed on the accuracy with which the approaches handle the challenges of state estimation for legged robots, as well as the level of implementation complexity they require to do so. As a result of this thesis, a Two Stage Estimation framework that separates linear and nonlinear estimation is proposed, which is based on prior work. The employment of an Unscented Kalman Filter for orientation estimation and a Classical Kalman Filter for position and velocity estimation is suggested, outlining the path to simpler and more accurate state estimation approaches for legged robots.

Chapter 1

Introduction

Throughout the rigorous process of evolution, legged locomotion has established itself as an efficient solution to navigating the rough terrain our planet consists of. The traits exhibited by this type of mobility present it as evidently preferable to other locomotion types in this context[8, 27, 10, 6, 12]. The limitations wheeled or tracked mobility faces in challenging environments are greatly reduced when the versatility of deployment enabled by legs is considered [6, 20]. These implications have led to the awareness of the disruptive possibilities legged locomotion presents for the field of robotics [6, 8]. As Bloesch [8] states, implementations of legged locomotion have greatly expanded the range of industry applications of robots within the fields of search and rescue, agriculture, mining, nuclear power, forestry, resource exploration, health care, or public services. The benefits presented by legged locomotion and its applications have recently led to significant progress in the development of ever improving legged robot platforms capable of dynamical stabilization [8, 19, 6, 20]. The mobility and versatility offered by platforms such as the quadrupedal robots ANYmal [20], Starl*ETH* [19], MIT Cheetah 3 [6] or Boston Dynamics Spot® underline the progress made by state-of-the-art legged robot implementations.

While the research in robot design and control are at the core of this development, the perceptive and estimation capacities of these platforms are essential to sustaining the increasing complexity and breadth of application [10, 8, 18]. To enable the advancement of algorithms for motion planning, as well as the robust feedback controllers necessary, accurate and fast estimates of the robot's state are required [10, 18, 9, 12]. Especially for trajectory planning and stabilization in rough or uneven terrain, a state estimator's precision is crucial to its successful deployment [10]. Further, when the robot is employed for tasks requiring a highly dynamic gait, state estimation may potentially become a bottleneck for speed and robustness within the control pipeline [10]. Spurred by the development of legged robot platforms, the importance of continuous review and improvement of state-of-the-art state estimation approaches becomes evident.

In this context, proprioceptive approaches present themselves as preferable estimation techniques [8]. Notably, the state-of-the-art state estimation approaches presented by Bledt et al. [6] and Bloesch et al. [10] succeed in employing proprioceptive sensors, namely an Inertial Measurement Unit (IMU) and joint encoders, to generate estimates of the robot's state with great precision. Apart from the required estimation accuracy, robustness and low estimation time, state estimation for legged robots presents three main challenges which must be considered when deriving a respective approach: sensors biases, sensor noise and nonlinear orientation dynamics. To address these issues, state estimators based on a stochastic Kalman Filtering approach are commonly employed [15]. While Bloesch et al. employ an Extended Kalman Filter which linearizes the dynamics to circumvent the nonlinearities [33, 28], Bledt et al. propose a Two Stage Estimation approach. In the Two Stage approach, the linear position and velocity estimation is done separately using a Classical Kalman Filter [25], while a Complementary Filter [31] is used for orientation estimation. Both further include leg kinematic measurements into their estimation to increase its accuracy and

to avoid employing computationally more expensive approaches [10].

With ever developing extension to the Kalman Filtering approach, the complexity of implementation rises simultaneously. With it, the difficulty of tuning the filter, adapting it or expanding it to new states increases. This begs the question whether the implementation complexity is justified by increase in estimation accuracy it entails or if simpler approaches exist that provide similar estimates. The Two Stage Estimator developed by Bledt et al. [6] presents an estimation approach which greatly decreases the complexity of implementation as compared to the EKF [6, 16]. As the linear states are now estimated by the Classical Kalman Filter, which is the optimal estimator in this case [25, 3], increased accuracy of position and velocity estimates are also to be expected. The orientation filter employed by Bledt et al. consists of a Complementary Filter implementation by Madgwick [31]. The Complementary Filter uses basic mechanisms of sensor fusion to efficiently combine gyroscope and accelerometer readings, resulting in a simple but accurate orientation estimator. Despite its simplicity, recent Kalman Filtering extensions such as the Unscented Kalman Filter [41, 30, 23] present viable alternatives to the Complementary Filter. The Unscented Kalman Filter provides the possibility of combining a Kalman Filtering approach for orientation estimation, while avoiding linearization and maintaining a relatively low complexity of implementation.

Considering the rapid developments within the field of legged robots and state estimation, as well as the aim to bring accuracy and complexity into accordance, the necessity of combining and comparing different estimation approaches becomes apparent. Based on prior work, the objective of this thesis is to implement and evaluate different approaches to state estimation for legged robots. A particular emphasis is placed on their ability to tackle the challenges presented by state estimation for legged robots, as well as their level of implementation complexity needed to do so. This is done by comparing and evaluating the performance of the Two Stage Estimation approach proposed by Bledt et al. [6] using both a Complementary Filter, as well as an Unscented Kalman Filter for orientation estimation, to that of the Extended Kalman Filter presented by Bloesch et al. [10]. To enable this comparison, the general and implementation specific background of the estimators is first expanded upon. Subsequently they are tested in simulation, using realistically generated measurements, across different scenarios. The resulting data is then evaluated and the differences between the estimation performances are highlighted and discussed with a distinction made between orientation estimation and position and velocity estimation. The filtering approaches best conforming with the mentioned criteria of accuracy and implementation complexity are then determined as the result of the comparison made in this thesis.

Chapter 2

Theory

In the following chapter, an introduction to State Estimation for legged robots, its machinery and the underlying theory of sensor fusion is presented. More specifically, the Kalman Filter is introduced in a general manner and its extensions to nonlinear estimation are discussed. Lastly, the Complementary Filter is proposed as an alternative to Kalman Filtering for orientation estimation.

2.1 Introduction to State Estimation

2.1.1 What is State Estimation?

State Estimation is a stochastic approach commonly employed in robotics to determine the probabilistic distribution of a robot's state. In order to control a robot successfully, accurate feedback about its current state must be available. The integration of a state estimator in the control pipeline relevant to this project is shown in Fig. 2.1.

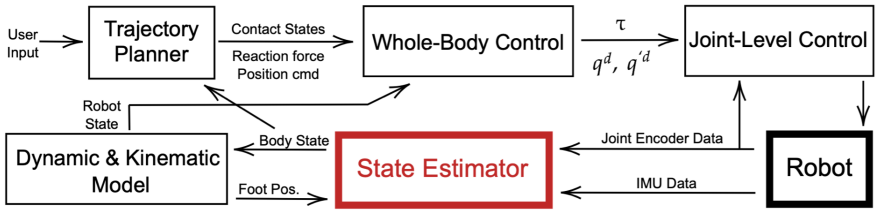

Figure 2.1: The State Estimator integrated into the Control Pipeline of the legged robot system employed in this project. The State Estimator acts as the link between the real robot and the framework used to control it. Using inertial measurements and joint encoder data from sensors employed on the robot, it generates an estimate of the robot's state using the dynamic and kinematic models of the system. This estimate is then provided to the Trajectory Planner and the Whole-Body Controller to derive the next control input.

While the system dynamics combined with the control input allow for the calculation of an expected state development, the robot's environment can strongly influence its motion by applying unknown external forces or torques. This disturbance introduces inaccuracies in the expected state which need to be accounted for. In robotics, this is done by employing sensors and forward kinematic models to supply a measurement feedback of the robot's current state. Again however, the measured state provided by the sensors is afflicted by inaccuracies due to noise and biases present in the measurements. State estimation thus condenses to combining these two imperfect state distributions to provide an estimate with increased accuracy. By using sensor fusion to combine the process model prediction with the measurements, as well as prior information about the model and sensor inaccuracies, the controller can be provided with an accurate estimate of the robot's

state from which both the trajectory and the necessary control inputs can be derived. A basic overview of the state estimation mechanism is shown in Fig. 2.2.

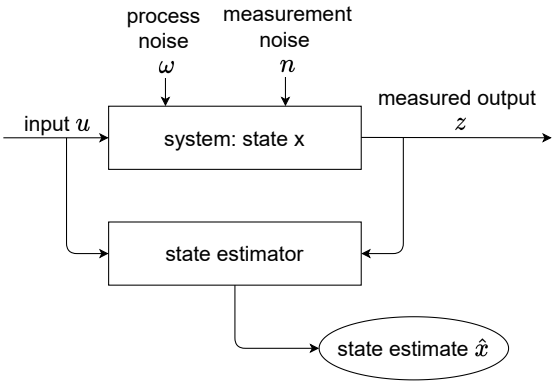

Figure 2.2: An overview of state estimation for dynamic systems showing how the state estimate $\hat{x}$ is derived from the known input u and measured output z.

2.1.2 What is the State?

In legged robotics, state estimation is a framework that allows for estimation of the full pose. This means, four of the robot's state variables are estimated: position $\boldsymbol{r}$, velocity $\boldsymbol{v}$, angular velocity $\boldsymbol{\omega}$ and orientation q. Along with acceleration measurements and prior knowledge of gravity direction and magnitude, these variables fully describe the system dynamics and make up the basic estimated state. Depending on the application and filtering approach, extensions to this basic state exist. Especially in legged robotics, state-of-the-art approaches include other variables such as the foot contact positions, as well as sensor bias terms. These are estimated in addition the robot's pose [6, 9, 10, 17]. This is done to increase the accuracy of the state estimate by efficiently including as much information as possible into the estimation process. The N dimensional state is commonly represented in vector form according to eq. 2.1, where k refers to the current discrete timestep and $\mathbf{x}_k$ is the state vector:

$$\boldsymbol{x}_k = [\boldsymbol{r}_k \ \boldsymbol{v}_k \ \boldsymbol{\omega} \ q_k \ ...] \in \mathbb{R}^N \tag{2.1}$$

As state estimation is a stochastic approach, the state itself is a random variable with a mean $\boldsymbol{\mu}_k$ and state covariance matrix P_k. What we commonly refer to as the state estimate $\hat{\boldsymbol{x}}$ is simply the mean of the state random variable.

2.1.3 State Estimation for Legged Robots

While state estimation is an essential part of the control pipeline in many different applications, each framework has its own dynamics and characteristics that must be considered when developing a suitable estimator. In the following, the dynamics involved in legged robot state estimation are introduced and subsequently, the challenges surrounding this specific application are discussed.

Legged Robot Dynamics

The differential equations governing the dynamic model of a legged robot are as follows [10]:

$$\dot{\boldsymbol{r}} = \boldsymbol{v} \tag{2.2}$$

$$\dot{\boldsymbol{v}} = \boldsymbol{a} = {}^{B}_{W}C\boldsymbol{f} + \boldsymbol{g} \tag{2.3}$$

$$\dot{q} = \frac{1}{2} q \otimes \boldsymbol{\omega} \tag{2.4}$$

Here $\boldsymbol{a}$ is the absolute acceleration of the robot's body in the world frame W, and $\boldsymbol{f}$ is the proper acceleration measured by the accelerometer in the body frame B. These two entities are related as follows:

$$\boldsymbol{f} = {}^{B}_{W}C^{\top}(\boldsymbol{a} - \boldsymbol{g}) \tag{2.5}$$

${}^{B}_{W}\boldsymbol{C}$ is the orientation matrix which rotates any vector ${}^{B}\boldsymbol{y}$ from the B to the W:

$${}^{W}\boldsymbol{y} = {}^{B}_{W}C{}^{B}\boldsymbol{y} \tag{2.6}$$

These equations form the foundation of the process model prediction of the state estimation algorithm, as described in section 2.2.

Challenges of State Estimation for Legged Robots

When tackling the challenge of state estimation for legged robots, there are, apart from the estimation itself, three main issues that must be kept in mind and dealt with when developing an estimation scheme:

1. Sensor Biases: Current state-of-the-art estimation approaches for legged robots rely on inertial sensors to provide essential measurements for the estimation process [10, 6, 29]. More precisely, Inertial Measurement Units (IMUs) are used to provide measurements of the robot's current angular velocity using a gyroscope, as well as its acceleration by means of an accelerometer [38]. Given perfect measurements and a perfect system model, these measurements alone would be sufficient to compute the robot's pose by means of dead-reckoning, as becomes evident from Fig. 2.3 [29]. However, biases are present in both sensors, which lead to accumulation of errors in the orientation estimation. Further, as the position estimation relies on double integration of accelerometer readings and is inherently linked to the orientation, it is effected even stronger by any biases present [29].

2. Sensor Noise: The second aspect that decreases the accuracy of the estimates is measurement noise. Assumed to be of Gaussian distribution [6, 10], noise is present in most any sensor reading and is further amplified by external factors such as foot-slippage or leg impact. Due to the nature of locomotion employed by legged robots, the latter has a considerable influence on the sensors measurements, which are rendered increasingly noisy due to intermittent ground impacts while walking [17].

3. Nonlinearity: A final challenge is presented in the form of the orientation estimation itself. While position estimation is a purely linear process, orientation estimation is an inherently nonlinear process [6]. By propagating a state variable through a nonlinear process or measurement model, the underlying assumption of a Gaussian distribution necessary for the conventional Kalman Filtering approach employed in this project is violated [37]. Therefore, alternative methods must be applied to deal with this nonlinearity efficiently .

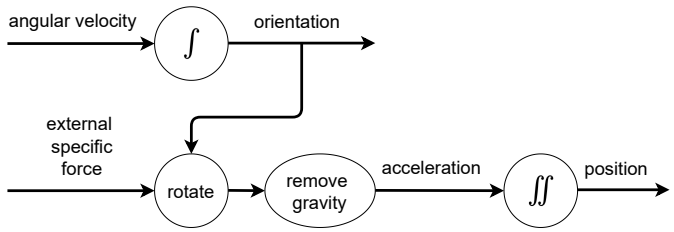

Figure 2.3: Dead-reckoning schematic for pose estimation according to [29]. The inherent link between orientation and position estimation, as well as the double integration necessary is evident.

2.2 The Kalman Filter

Different methods for state estimation exist and the choice of the estimation approach is based on the robot's structure and operating environment [17]. Due to the challenges presented in section 2.1.3, as well as the process and kinematic models available, this project focuses on using a Kalman Filter based approach to state estimation. Kalman Filtering is a stochastic recursive estimation technique that combines information about the system, measurement devices, input and measured output to provide an estimate of the state distribution of a dynamic system [42, 28, 32, 25]. While the Classical Kalman Filter (KF) is suited for the estimation of linear systems, for which it can be seen as the optimal estimator [39, 32]. However, there exist extensions to nonlinear applications. Most commonly the Extended Kalman Filter (EKF) [37, 33], Unscented Kalman Filter (UKF) [24, 41, 23] and the Invariant Extended Kalman Filter (InEKF) [11, 5, 18] are used for the estimation of nonlinear systems. In the following, the Classical Kalman Filter is presented from which the extensions are then derived. Fig. 2.4 shows the three different Kalman Filters — the KF, EKF and UKF — employed in this project.

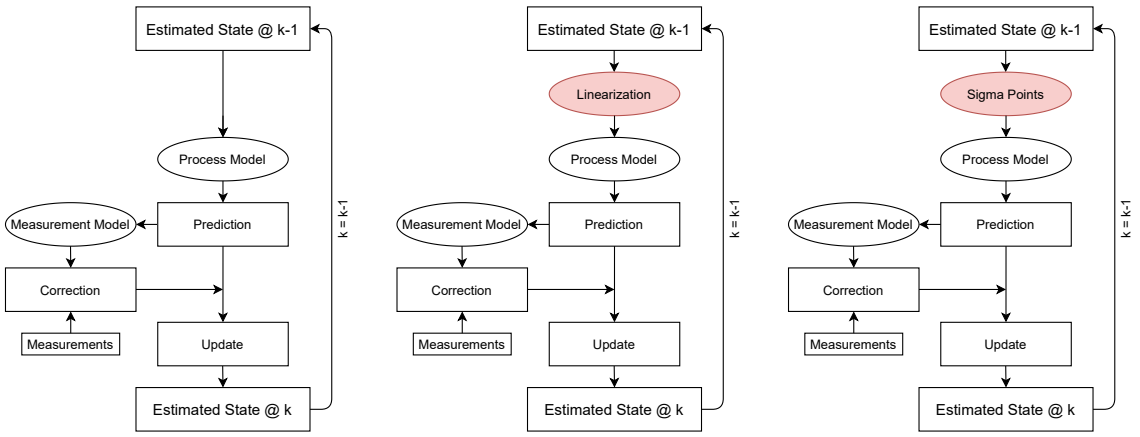

Figure 2.4: Overview of the Kalman Filter approaches: Classical KF (left), EKF (center), UKF (right). Here k is the discrete timestep.

2.2.1 The Classical Kalman Filter

The Classical Kalman Filter follows the assumption of a linear process and measurement model, as well as a Gaussian distribution of the state, noise variables and initial conditions. The linear, discrete-time and time-varying system for which the Kalman Filter is applied is shown in eq. 2.7 and eq. 2.8 [3]:

$$\boldsymbol{x}_k = A\boldsymbol{x}_{k-1} + B\boldsymbol{u}_k + \boldsymbol{\omega}_k \quad k = 1, 2, \dots \tag{2.7}$$
$$\boldsymbol{z}_k = C_k\boldsymbol{x}_k + \boldsymbol{n}_k \quad k = 0, 1, \dots \tag{2.8}$$

$$\text{system state} : \boldsymbol{x}_k \in \mathbb{R}^N \tag{2.9}$$
$$\text{initial state} : \boldsymbol{x}_0 \in \mathbb{R}^N \sim \mathcal{N}\left(\check{\boldsymbol{x}}_0, \check{P}_0\right) \tag{2.10}$$
$$\text{input} : \boldsymbol{u}_k \in \mathbb{R}^N \tag{2.11}$$
$$\text{process noise} : \boldsymbol{\omega}_k \in \mathbb{R}^N \sim \mathcal{N}\left(\mathbf{0}, Q_k\right) \tag{2.12}$$
$$\text{measurement} : \boldsymbol{z}_k \in \mathbb{R}^M \tag{2.13}$$
$$\text{measurement noise} : \boldsymbol{n}_k \in \mathbb{R}^M \sim \mathcal{N}\left(\mathbf{0}, R_k\right) \tag{2.14}$$

A is the process model, B the control model and C the measurement model of our system. These models are determined by the system dynamics and the filter design. $\check{\boldsymbol{x}}_0$ and $\check{P}_0$ are the initial conditions of the state, Q_k is the process noise covariance and R_k is the measurement noise covariance. As before, these parameters are determined by the specific application.

The Kalman Filter framework is divided into two steps: the Prediction and the Update [28], which are elucidated in the following sections.

The Prediction

In the prediction step, an *a priori* estimate $\hat{\boldsymbol{x}}_k^-$ of the state is computed by propagating the previous *a posteriori* estimate $\hat{\boldsymbol{x}}_{k-1}^+$ through the process model A and adding the control input $B\boldsymbol{u}_k$. Further, an *a priori* estimate of the covariance $\hat{P}_k^-$ is calculated in a similar fashion, combining prior knowledge of the previous state covariance $\hat{P}_{k-1}^+$ and the process noise covariance Q_k:

$$\hat{\boldsymbol{x}}_k^- = A\hat{\boldsymbol{x}}_{k-1}^+ + B\boldsymbol{u}_k \tag{2.15}$$
$$\hat{P}_k^- = A\hat{P}_{k-1}^+A^\top + Q_k \tag{2.16}$$

In essence, the prediction step constitutes of combining prior knowledge of the previous state with knowledge of the system dynamics, input and covariance to produce a first estimate of the state.

The Update

The update is initiated once the next measurement of the current state becomes available. In this step, the measured outcome is used to compute a weighted mean between the *a priori* estimate $\hat{\boldsymbol{x}}_k^-$ and the measured state $\boldsymbol{z}_k$. First, the innovation $\tilde{\boldsymbol{y}}_k$ is computed which is weighted by the Kalman Gain K_k and combined with the *a priori* estimate $\hat{\boldsymbol{x}}_k^-$ to calculate the *a posteriori* estimate $\hat{\boldsymbol{x}}_k^+$ of the state. In a final step, the state covariance is updated such that we receive the *a posteriori* estimate $\hat{P}_k^+$.

$$\tilde{\boldsymbol{y}}_k = \boldsymbol{z}_k - C\hat{\boldsymbol{x}}_k^- \tag{2.17}$$
$$K_k = P_k^-C^\top(R + CP_k^-C^\top)^{-1} \tag{2.18}$$
$$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + K_k\tilde{\boldsymbol{y}}_k \tag{2.19}$$
$$\hat{P}_k^+ = (I - K_kC)\hat{P}_k^- \tag{2.20}$$

This step can be seen as combining two different state distributions and weighting them by considering the uncertainties (covariances) involved in their derivation, such that the best estimate of

the real state can be computed.

The main drawback of the Classical Kalman Filter is its inability to deal with nonlinear process and measurement models. Due to its dependence on Gaussian distributions, any nonlinearity would corrupt this assumption and thus invalidate any further updates [24, 37]. To overcome this challenge, extensions have been developed which deal with the nonlinearity in different ways.

2.2.2 The Extended Kalman Filter

Many applications for which one might want to employ a Kalman Filter are not linear systems. On the contrary, they often consist of nonlinear dynamics, which render the Classical Kalman Filter unsuitable for the task of estimation. However, as is common when dealing with nonlinear models, the dynamics can be linearized around a point of interest and the estimation can be carried out as before [24]. This is the essence of an Extended Kalman Filter. The EKF follows from the KF by making some slight adjustments to the filter framework [28]. Below the discrete-time and time-varying system presented in section 2.2.1 is recalled, newly adapted to be a nonlinear system:

$$\boldsymbol{x}_k = a(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k) + \boldsymbol{\omega}_k \quad k = 1, 2, \dots \tag{2.21}$$
$$\boldsymbol{z}_k = c(\boldsymbol{x}_k) + \boldsymbol{n}_k \quad k = 0, 1, \dots \tag{2.22}$$

Here, a is the nonlinear process and c the nonlinear measurement model. The nonlinearity becomes evident when considering that the output $\boldsymbol{x}_k$ and measurement $\boldsymbol{z}_k$ can no longer be represented as a linear combination of the previous state $\boldsymbol{x}_{k-1}$, as well as the input $\boldsymbol{u}_k$. To be able to employ the Classical Kalman Filtering framework derived in section 2.2.1, the nonlinear process and measurement models are now linearized around the previous estimate $\hat{\boldsymbol{x}}_{k-1}^{+}$ and input $\boldsymbol{u}_{k-1}$ and the current estimate $\hat{\boldsymbol{x}}_k^{-}$ respectively:

$$A_{k-1} = \left.\frac{\partial a}{\partial x}\right|_{\hat{\boldsymbol{x}}_{k-1}^{+}, \boldsymbol{u}_{k-1}} \tag{2.23}$$
$$C_k = \left.\frac{\partial c}{\partial x}\right|_{\hat{\boldsymbol{x}}_k^{-}} \tag{2.24}$$

Here the subscripts $k-1$ and k now refer to the estimate about which the models were linearized. The prediction step then follows naturally as:

$$\hat{\boldsymbol{x}}_k^{-} = f(\hat{\boldsymbol{x}}_{k-1}^{+}, \boldsymbol{u}_k) \tag{2.25}$$
$$\hat{P}_k^{-} = A_{k-1}\hat{P}_{k-1}^{+}A_{k-1}^{\top} + Q_k \tag{2.26}$$

Lastly, the update is computed:

$$\tilde{\boldsymbol{y}}_k = \boldsymbol{z}_k - C\hat{\boldsymbol{x}}_k^{-} \tag{2.27}$$
$$K_k = P_k^{-}C_k^{\top}(R + C_k P_k^{-} C_k^{\top})^{-1} \tag{2.28}$$
$$\hat{\boldsymbol{x}}_k^{+} = \hat{\boldsymbol{x}}_k^{-} + K_k\tilde{\boldsymbol{y}}_k \tag{2.29}$$
$$\hat{P}_k^{+} = (I - K_k C_k)\hat{P}_k^{-} \tag{2.30}$$

The brilliance of the EKF lies in allowing the use of familiar methods with slight modifications. Further, it is computationally cheaper than other methods such as the particle or point-mass filter, which are also used for nonlinear problems [28]. On the downside however, we are left to deal with the limitations of linearization. These include calculating jacobians, linearization errors and difficulty to tune, all of which add to the complexity and inaccuracy of the filter [24].

2.2.3 The Unscented Kalman Filter

To address the shortcomings present in the EKF, the UKF employs a different approach. While the UKF maintains the same two-step prediction and update procedure, the mechanisms involved are slightly different. Instead of using linearization, the UKF uses sampling to bypass any non-linearities while still guaranteeing a Gaussian distribution of the state. This is done by means of applying the Unscented Transform to propagate carefully chosen sigma vectors (samples) through the nonlinear process and measurement models [24, 30, 41]. The adapted estimation procedure is described in the following.

Sigma Vector Sampling

As before, the challenge consists of estimating the nonlinear system presented in section 2.2.2. The first step of the UKF consists of sampling a set of sigma vectors $\mathcal{X}_i$ that fully describe the estimated state at $k-1$, ie. the *a priori* estimate of the state. There are different approaches on how to choose these sigma vectors, a possible set is presented here from [30].

The first step consist of calculating the matrix S which is defined in eq. 2.31, where P_{k-1}^{+} is the previous state covariance and Q_k denotes the process noise covariance matrix:

$$S = \sqrt{P_{k-1}^{+} + Q_k} \tag{2.31}$$

In practice, a Cholesky Decomposition is commonly used for calculating S. Next, the column vectors of S are used to calculate the set of $2n$ vectors $\{\mathcal{W}_i\}$ as follows, where n is the dimension of the state:

$$\mathcal{W}_{i,i+n} = \text{columns}\left(\pm\sqrt{2n \cdot \left(P_{k-1}^{+} + Q_k\right)}\right) \tag{2.32}$$

The sigma vectors are then computed, where $\hat{\boldsymbol{x}}_{k-1}^{+}$ denotes the previous state estimate:

$$\mathcal{X}_i = \hat{\boldsymbol{x}}_{k-1}^{+} + \mathcal{W}_i \tag{2.33}$$

It should be noted that this calculation is only valid for vector space state variables. When employing quaternion representation for the orientation state however, $\hat{\boldsymbol{x}}_{k-1}^{+} + \mathcal{W}$ corresponds to $\hat{q}_{k-1}^{+} q_{\mathcal{W}_i}$, where $\hat{q}_{k-1}$ is the orientation estimate from the prior timestep and $q_{\mathcal{W}_i}$ is a quaternion from the set $\{\mathcal{W}_i\}$.

The Prediction

The prediction follows a similar schematic as in the KF and the EKF. However, since we are dealing with the propagation of several state vectors, ie. the sigma vectors, some additional steps are required. First, the sigma vectors $\mathcal{X}_i$ are passed through the process model A, resulting in the propagated set of state vectors $\{\mathcal{Y}_i\}$:

$$\mathcal{Y}_i = A(\mathcal{X}_i) \tag{2.34}$$

The computed set $\{\mathcal{Y}_i\}$ is now used to calculate the a priori estimate $\hat{\boldsymbol{x}}_k^{-}$ and covariance P_k^{-}:

$$\hat{\boldsymbol{x}}_k^{-} = \text{mean}\left(\{\mathcal{Y}_i\}\right) \tag{2.35}$$

$$P_k^{-} = \text{covariance}\left(\{\mathcal{Y}_i\}\right) \tag{2.36}$$

The procedure involved to calculate the mean and covariance are dependent on the representations used in the state vector. The definitions specific to the implementation used in this project are given in subsections following the general update step description.

The Update

In the update step, the priorly computed set $\{\mathcal{Y}_i\}$ is transformed by the measurement model C, resulting in the set of measurement vectors $\{\mathcal{Z}_i\}$:

$$\mathcal{Z}_i = C(\mathcal{X}_i) \tag{2.37}$$

As before, the expected measurement $\hat{\boldsymbol{z}}_k^-$ is the mean of the set $\{\mathcal{Z}_i\}$ and the expected covariance matrix P_{zz} is its covariance:

$$\hat{\boldsymbol{z}}_k^- = \text{mean}\left(\{\mathcal{Z}_i\}\right) \tag{2.38}$$
$$P_{zz} = \text{covariance}\left(\{\mathcal{Z}_i\}\right) \tag{2.39}$$

Finally, the innovation $\tilde{\boldsymbol{y}}_k$ and Kalman gain K_k can be calculated and subsequently the a posteriori estimate and state covariance are computed, where R is the measurement noise covariance matrix and P_{xz} is the cross correlation matrix:

$$\tilde{\boldsymbol{y}}_k = \boldsymbol{z}_k - \hat{\boldsymbol{z}}_k^- \tag{2.40}$$
$$K_k = P_{xz}(P_{zz} + R)^{-1} \tag{2.41}$$
$$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + K_k\tilde{\boldsymbol{y}}_k \tag{2.42}$$
$$\hat{P}_k^+ = (I - K_kC_k)\hat{P}_k^- \tag{2.43}$$

Mean Calculation

When our state variables are simple elements of a vector space, for example angular velocity, the calculation of the set mean is straightforward. It follows as the *barycentric mean*, where $2n$ is the number of vectors within the set $\{\mathcal{X}_i\}$:

$$\text{mean}\left(\{\mathcal{X}_i\}\right) = \frac{1}{2n}\sum_{i=1}^{2n}\mathcal{X}_i \tag{2.44}$$

However, for state variables which are members of a homogenous Riemannian manifold, this computation does not yield valid results [30]. Since this thesis aims at employing the UKF for orientation estimation purposes using a quaternion representation, a new method must be found. Kraft [30] suggests using a gradient descent algorithm for averaging a set of quaternions, an identical approach is adopted in this project.

Covariance Calculation

For a general set of $2n$ vectors $\{\mathcal{X}_i\}$ and mean $\bar{\boldsymbol{x}}$, the set covariance follows as:

$$P_{xx} = \frac{1}{2n}\sum_{i=1}^{2n}[\mathcal{X}_i - \bar{\boldsymbol{x}}][\mathcal{X}_i - \bar{\boldsymbol{x}}]^\top \tag{2.45}$$

Again however, dealing with orientation quaternions as state variables entails its difficulties. The simple subtraction of the mean from the set element is no longer viable. Instead, $[\mathcal{X}_i - \bar{\boldsymbol{x}}]$ corresponds to $[q_{\mathcal{X}_i}\bar{q}^{-1}]$, where $\bar{q}$ is the set mean and $q_{\mathcal{X}_i}$ a quaternion of the set $\{\mathcal{X}_i\}$ [30].

The cross correlation matrix calculation follows similarly. In this application, it is solely used to calculate the cross correlation between the sets $\mathcal{W}_i^{'}$ and $\mathcal{Z}_i$:

$$P_{xz} = \frac{1}{2n}\sum_{i=1}^{2n}[\mathcal{W}_i^{'}][\mathcal{Z}_i - \boldsymbol{z}_k^-]^\top \tag{2.46}$$

Here $\mathcal{W}_i^{'}$ corresponds to $[\mathcal{Y}_i - \hat{\boldsymbol{x}}_k^-]$ in the case of vector space state variables or $[q_{\mathcal{Y}_i}\hat{q}_k^{-}{}^{-1}]$ for quaternion state variables.

The main benefits proposed by the UKF are its ability to eliminate linearization from the estimation process, which in turn also alleviates the need for jacobian calculation. Not only does this offer increased robustness against nonlinearities, it renders the implementation less complex and easier to adapt to new state extensions. However, as we are propagating numerous sampling points, the computational costs are slightly higher as compared to an EKF [9].

2.2.4 The Invariant Extended Kalman Filter

Recent developments in state estimation for robotics have brought forward a novel way of dealing with the challenges of pose estimation. As an extension to the EKF, the InEKF is presented as a new estimator that is based on invariant observer design theory [18, 11, 4, 5]. The inclusion of knowledge on the geometric structure of both the involved dynamics and state space structure into the approach, yield an estimator that improves on the trusted performance of the EKF [5, 18]. This mention of the InEKF is included here solely for completeness, the InEKF is not expanded upon or employed in this thesis. However, as it shows great promise for pose estimation applications [18], the reader should be aware of its existence.

2.3 The Complementary Filter

While Kalman Filtering approaches offer many important benefits, making them very attractive for pose estimation, their stochastic nature render them increasingly complex and computationally heavy. When orientation estimation is considered independently, a simpler and remarkably computationally inexpensive alternative can be found in the form of the Complementary Filter [31, 16]. The underlying principle of the Complementary Filter is to combine gyroscope measurements with accelerometer readings. The gyroscope provides an increasingly accurate high-frequency response, while the accelerometer readings are able to de-drift the estimate at lower frequencies [6, 34]. Pairing these two complementary measurements through a sensor fusion framework then provides an orientation estimate with increased accuracy, even at low sampling rates [31]. A schematic of this process is shown in Fig. 2.5.

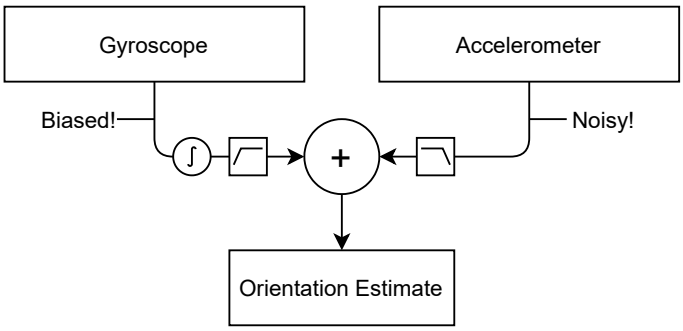

Figure 2.5: A basic Complementary Filter implementation using high- and low-pass filtering for gyroscope and accelerometer readings respectively.

The implementation used in this project follows the Complementary Filter proposed by Madgwick [31], which uses a quaternion representation for orientation and relies on optimization for sensor fusion. While an extension for the additional usage of a MARG sensor array exists, this project focuses on the IMU centered implementation. The filter algorithm can be divided into three steps: orientation estimation from angular rate, orientation estimation from acceleration and filter fusion. In the following, these three steps are explained.

Orientation from Angular Rate

This first step consists of calculating the estimated orientation quaternion ${}^{B}_{W}\hat{q}_{\omega,k}$ at the timestep k from the estimate at the prior timestep ${}^{B}_{W}\hat{q}_{est,k-1}$ and the current gyroscope readings ${}^{B}\boldsymbol{\omega}_k$. Here the sub- and superscripts ${}^{B}_{W}$ refer to the body frame and world frame respectively, ie. ${}^{B}\boldsymbol{\omega}$ is measured in the body frame B and ${}^{B}_{W}q$ rotates a vector from body frame B to world frame W. Further, the subscript ${}_{\omega}$ indicates that the orientation estimation is the result of the angular rate step only, while ${}_{est}$ refers to it being the result from the combined steps.

First, the quaternion derrivative ${}^{B}_{W}\dot{q}_{\omega,k}$, which quantifies the rate of change of the world frame relative to the sensor frame, is calculated. The orientation estimation then follows from numerical integration, where Δt denotes the sampling period:

$$ {}^{B}_{W}\dot{q}_{\omega,k} = \frac{1}{2} {}^{B}_{W}\hat{q}_{est,k-1} \otimes {}^{B}\boldsymbol{\omega}_k \tag{2.47} $$

$$ {}^{B}_{W}\hat{q}_{\omega,k} = {}^{B}_{W}\hat{q}_{est,k-1} + {}^{B}_{W}\dot{q}_{\omega,k}\Delta t \tag{2.48} $$

Orientation from Acceleration

This step aims at calculating the orientation ${}^{B}_{W}\hat{q}_{\Delta,k}$ by comparing the known gravity direction ${}^{W}\hat{\mathbf{g}}$ with the direction measured by the accelerometer ${}^{B}\hat{\mathbf{s}}_k$. Here the subscript ${}_{\Delta}$ indicates that the orientation estimation is calculated solely in this step and not the combined steps. However, this simple idea would result in an infinite amount of solutions, since the true orientation rotated around an axis parallel to the known direction is again a solution. As we are employing a quaternion representation of the orientation, which requires a complete solution, this redudancy is problematic. Here Madgwick proposes the solution in form of an optimization problem, seeking to minimize the error between the gravity vector calculated from the accelerometer reading ${}^{B}\hat{\mathbf{s}}_k$ using the preliminary orientation estimate ${}^{B}_{W}\hat{q}_k$ and the known gravity direction ${}^{W}\hat{\mathbf{g}}$. The problem can be formulated as follows:

$$ \min_{{}^{B}_{W}\hat{q}_k \in \mathbb{R}^4} \boldsymbol{f}({}^{B}_{W}\hat{q}_k, {}^{W}\hat{\mathbf{g}}, {}^{B}\hat{\mathbf{s}}_k) \tag{2.49} $$

$$ \boldsymbol{f}({}^{B}_{W}\hat{q}_k, {}^{W}\hat{\mathbf{g}}, {}^{B}\hat{\mathbf{s}}_k) = {}^{B}_{W}\hat{q}^{*}_k \otimes {}^{W}\hat{\mathbf{g}} \otimes {}^{W}_{W}{}^{B}\hat{q}_k - {}^{B}\hat{\mathbf{s}} \tag{2.50} $$

$$ {}^{B}_{W}\hat{q}_k = [q_w \; q_x \; q_z \; q_y] \tag{2.51} $$

$$ {}^{W}\hat{\mathbf{g}} = [0 \; g_x \; g_y \; g_z] \tag{2.52} $$

$$ {}^{B}\hat{\mathbf{s}} = [0 \; s_x \; s_y \; s_z] \tag{2.53} $$

$$ \tag{2.54} $$

Madgwick employs a gradient descent algorithm for optimization and assumes convergence in one step in order to keep the computational costs low. While ${}^{B}_{W}\hat{q}_{\Delta,k}$ can be calculated explicitly, this is neglected here as it is not needed for the further steps of the algorithm. The result of this step is the objective function gradient $\nabla \boldsymbol{f}({}^{B}_{W}\hat{q}_k, {}^{W}\hat{\mathbf{g}}, {}^{B}\hat{\mathbf{s}}_k)$ which is calculated according to eq. 2.55, where $\boldsymbol{J}({}^{B}_{W}\hat{q}_k, {}^{W}\hat{\boldsymbol{g}})$ is the Jacobian of the objective function:

$$ \nabla \boldsymbol{f}({}^{B}_{W}\hat{q}_k, {}^{W}\hat{\mathbf{g}}, {}^{B}\hat{\mathbf{s}}_k) = \boldsymbol{J}^{\top}({}^{B}_{W}\hat{q}_k, {}^{W}\hat{\mathbf{g}}) \boldsymbol{f}({}^{B}_{W}\hat{q}_k, {}^{W}\hat{\mathbf{g}}, {}^{B}\hat{\mathbf{s}}_k) \tag{2.55} $$

Filter Fusion

The last step of the algorithm is concerned with fusing the two prior orientation estimates ${}^{B}_{W}\hat{q}_{\omega,k}$ and ${}^{B}_{W}\hat{q}_{\Delta,k}$ to calculate the improved estimate ${}^{B}_{W}\hat{q}_{est,k}$. The algorithm proposed by Madgwick greatly simplifies this step in favor of efficiency. First the direction of the estimated orientation derivative error ${}^{B}_{W}\dot{\hat{q}}_{\epsilon,k}$ is calculated using the computed objective function gradient $\nabla \boldsymbol{f}$. The combined estimated orientation derivative ${}^{B}_{W}\dot{q}_{est,k}$ is then the derivative calculated in the first step ${}^{B}_{W}\dot{q}_{\omega,k}$ with the error compensated by subtracting the scaled derivative error ${}^{B}_{W}\dot{\hat{q}}_{\epsilon,k}$. Here β is the gain parameter used for scaling the error, which takes the magnitude of the gyroscope measurement error $\tilde{\omega}_\beta$ into consideration. The estimated orientation ${}^{B}_{W}\hat{q}_{est,k}$ then follows from integration.

$$
\begin{aligned}
{}^{B}_{W}\dot{\hat{q}}_{\epsilon,k} &= \frac{\nabla \boldsymbol{f}}{\|\nabla \boldsymbol{f}\|} && (2.56)\\
{}^{B}_{W}\dot{q}_{est,k} &= {}^{B}_{W}\dot{q}_{\omega,k} - \beta {}^{B}_{W}\dot{\hat{q}}_{\epsilon,k} && (2.57)\\
{}^{B}_{W}\hat{q}_{est,k} &= {}^{B}_{W}\hat{q}_{est,k-1} + {}^{B}_{W}\dot{q}_{est,k}\Delta t && (2.58)\\
\beta &= \|\frac{1}{2}\hat{q} \otimes [0\ \tilde{\omega}_\beta\ \tilde{\omega}_\beta\ \tilde{\omega}_\beta\| = \sqrt{\frac{3}{4}}\tilde{\omega}_\beta && (2.59)
\end{aligned}
$$

The Complementary Filter succeeds at offering a simple, lightweight and efficient algorithm for orientation estimation. It can easily be integrated into the estimation framework and, supplied with IMU measurements, provides an accurate orientation estimate. Its simplicity however also prevents any extensions to the variables being estimated. While variables such as sensors biases can be readily estimated by Kalman Filtering approaches, this is not possible when employing a Complementary Filter. Further, this implementation offers only one parameter to tune to the specific application, significantly less than when using a Kalman Filter.

Chapter 3

Methods

In the following chapter, the implemented state estimation approaches are discussed. First, the choice of sensors is explained with a specific focus on simplicity and robustness. Further, the concept of Two Stage Estimation which this project relies on is introduced and its mechanism and benefits are elucidated. Expanding on this, the chosen implementations of orientation filters, as well as position and velocity filters are discussed. Lastly, it is explained how the performance of the different state estimation approaches is tested and by which metrics it is delineated.

3.1 Sensory Devices

While many different sensors have been shown to valuable to state estimation problems, this thesis focuses on the use of proprioceptive sensors only. More specifically, the following two sensors are used:

1. IMU: The IMU provides measurements of the robot's acceleration and angular velocity to the estimator. This is done by means of an accelerometer and gyroscope respectively. This information is essential to estimating the robots pose [29].

2. Joint Encoders: The joint encoders provide the estimator with angular position measurements of all joints. By means of kinematics, the position of any rigid link can be calculated. This is especially useful to estimate the position of the robot's feet, yielding a valuable measurement for pose estimation purposes [17].

The reasoning for this approach is twofold. Firstly, the use of a minimalistic, proprioceptive sensor setup offers a high level of simplicity and versatility of the estimation approach. The IMU is self-contained and its measurements provide readily available data that can be easily employed for estimation purposes. In contrast, visual sensor or laser based approaches (such as lidar) often entail expensive computational costs and are thus not applicable with the same breadth as inertial sensor approaches [1]. Unfortunately, inertial sensors do suffer more significantly from both sensor noise and position drift over time, mainly due to the second-order integration required for position estimation, than their visual or laser based counterparts [1]. However, as has been shown in prior works, fusing information from kinematics and inertial sensors by means of a Kalman Filter based approach offers very promising results in terms of accuracy of the estimation and simplicity of its deployment [6, 10, 9, 16, 18].

Secondly, proprioceptive sensors offer a robust set of measurements, largely irrespective of their environment. Exteroceptive sensors, as the name implies, rely heavily on external information [17] and are thus very sensitive to their environment. Visual approaches are strongly influenced by lighting and imaging conditions, laser and sonar sensors are conditional to the surrounding surface material and orientation and GPS/GNSS approaches are dependent on the ability to receive uncorrupted satellite signal [1]. As proprioceptive sensors make measurements of internal values [17],

they do not suffer from the same limitations. However it should be recalled that, as is discussed in section 2.1, legged robots in particular are susceptible to noisy internal sensor measurements due to inevitable intermittent ground contact brought on by the nature of their locomotion.

It should be noted that the robot to which the estimation approaches in this thesis are tailored also employs foot pressure sensors. These sensors provide foot contact measurements which are also used in the estimation approaches. However, as the evaluation is done in simulation and future work aims at replacing these sensors by means of contact estimation techniques [21, 7, 12], they are not further considered.

3.2 Two Stage Estimation

While the Classical Kalman Filter cannot be applied to systems with a nonlinear process or observation model due to the underlying assumption of a Gaussian distribution being violated, it can still be used to estimate any linear part of the state with great accuracy. With this in mind, an estimator that combines the accuracy and simplicity of the Classical Kalman Filter with a nonlinear filter equipped to handle the nonlinear estimation would provide a very simple and adaptable method of state estimation. Legged robotics presents an ideal situation to apply this logic. While orientation estimation is a highly non-linear process, position and velocity is well suited to be estimated using linear models, as the underlying dynamics are linear. Such an approach is employed with great success in prior work by Bledt et al. [6], where a Two Stage Estimator (TS) is developed for the MIT Cheetah 3 quadruped robot and by Flayols et al. [16], where the same is done for the HRP-2 humanoid robot [26]. Fig. 3.1 shows the structural differences between pose estimation using a EKF framework versus using a Two Stage Estimation framework.

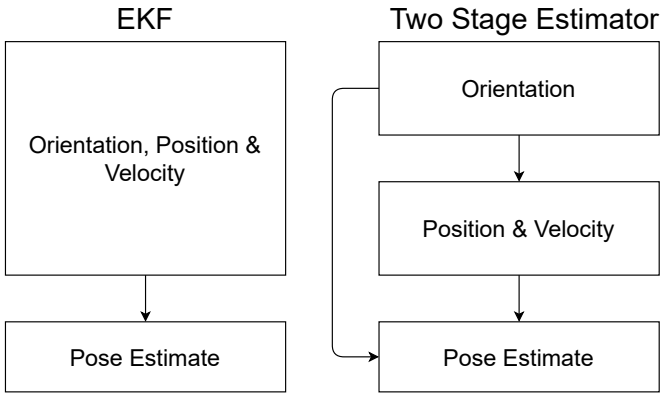

Figure 3.1: The EKF (left) estimates the entire pose within the same algorithm. The TS (right) on the other hand estimates the orientation separately from the position and velocity.

By decoupling the orientation estimation from the linear estimation of the velocity and position, two primary benefits become evident.

Firstly, as the position and velocity follow linear dynamics, the accuracy and simplicity the Classical Kalman Filter provides for such scenarios can be readily exploited. Not only is the Classical Kalman Filter the optimal estimator for linear problems [37], the difficulty of implementation is also reduced. There is no longer the need for including these linear states in the linearization framework, greatly simplifying the task at hand.

Secondly, the two stage estimation approach provides an excellent platform for comparing and employing different non-linear and linear filtering approaches. This benefit is illustrated by Floyals et al. [16], where two different approaches to pose estimation are compared. Building on the basis

of the linear estimation filter, much any non-linear filter can be implemented to provide the needed orientation estimates. As this thesis shows, especially applications in research can benefit from this modularity, making the benchmarking of different approaches very simple.

Building on this approach of Two Stage Estimation, section 3.3 elaborates on the specific orientation filter implementations employed in this thesis, while section 3.4 focuses on the position and velocity filters. To avoid any redundancy, only the specific implementation is discussed, as the theory of the respective filters is explained priorly in chapter 2.

3.3 The Orientation Filter

3.3.1 The Complementary Filter

Both Bledt et al. [6] and Flayols et al. [16] employ adaptations of the Complementary Filter based approach in their estimation techniques. Primary arguments for using the CF for orientation estimation in this project is its undeniable simplicity in implementation and the amount of relevant information readily available from prior works.

The CF implemented for this thesis is based on the IMU filter originally proposed by Madgwick [31] and explained in section 2.3. The provided algorithm is adapted to function within the y-up axis convention used for this project. This is done by accordingly setting the predefined reference direction of the gravity field in the earth's frame ${}^{E}\hat{\boldsymbol{d}} = [0\,0\,1\,0]$. This further entails changes in the objective function $f({}^{S}_{E}\hat{q}, {}^{E}\hat{\boldsymbol{d}}, {}^{S}\hat{\boldsymbol{s}})$ and its Jacobian.

Further, the beta parameter is tuned to correspond with the inertial sensor parameters assumed in the testing simulation. More specifically, it is set to 1.09254×10^{-3} which is calculated based on the eq. 2.59 proposed by Madgwick. Here $\tilde{\omega}_\beta$ is set to the assumed gyroscope white noise strength of $8.920572 \times 10^{-5}\,{}^{\text{rad}}/_{\text{s}^2}\sqrt{\text{Hz}}$ discretized at an IMU sampling frequency of 200 Hz (a more in depth discussion of the noise generation is done in section 4.1.1).

Due to its prior employment in very similar applications, ie. orientation estimation for legged robots [6, 16], the CF is assumed in this project as a baseline performance benchmark which the following approaches seek to improve upon.

3.3.2 The Unscented Kalman Filter

The UKF employed in this project is derived from the implementation introduced by Kraft [30], which is explained in section 2.2.3 and uses a quaternion representation of orientation. This filtering approach is chosen to explore simpler alternatives to the EKF approach employed in state-of-the-art state estimation implementations [10]. Especially the sampling and propagating of sigma vectors offers an interesting alternative to linearization commonly applied in non-linear estimation scenarios.

For this thesis, two different versions of the UKF proposed by Kraft are implemented. This distinction is done to determine if the angular velocity must necessarily be estimated alongside the orientation or if the measurements provided by the IMU suffice.

The first version, labelled UKF, focuses solely on orientation estimation. While Kraft includes the angular velocity $\boldsymbol{\omega} = [\omega_x\ \omega_y\ \omega_z]$ into the filter state, this is fully neglected in the first implementation, producing the following state vector $\boldsymbol{x}$, where q is the orientation quaternion $q = [q_w\, q_x\, q_y\, q_z]$:

$$\boldsymbol{x} = [q] \tag{3.1}$$

This reduces the DOF of the state to three, meaning less sigma vectors have to be calculated and propagated through the algorithm. It further eliminates the need for process and measurement

models that deal with two distinct physical entities, ie. the angular velocity $\boldsymbol{\omega}$ and the orientation quaternion q. This yields a simpler filtering algorithm which is only concerned with estimating the robots orientation.

The second version, labelled UKF AV, implements the estimation algorithm for the state including the angular velocity, as is proposed by Kraft. The state vector $\mathbf{x}$ now becomes:

$$\mathbf{x} = [q\ \boldsymbol{\omega}] \tag{3.2}$$

As mentioned before, a primary challenge now is the necessity of dealing with both orientation quaternions and angular velocity vectors. However, the extensions explained in section 2.2.3 deal with this issue efficiently.

3.4 The Position and Velocity Filter

3.4.1 The Classical Kalman Filter

The Two Stage Estimation approach allows for employing a Classical Kalman Filter for position and velocity estimation. The KF implemented in this project is based on the filter presented by Bledt et al. [6], which in turn functions similarly as the linear section of the filter developed by Bloesch et al. [10]. Both approaches suggest augmentations to the basic KF framework by including kinematic calculations of the robot's relative foot positions. As is emphasized by Bloesch et al., this efficiently increases the estimation accuracy. Bledt et al. go one step further and also include relative foot velocity calculations aimed at further increasing estimation performance. A schematic of the Kalman Filter approach including these augmentations is shown in Fig. 3.2.

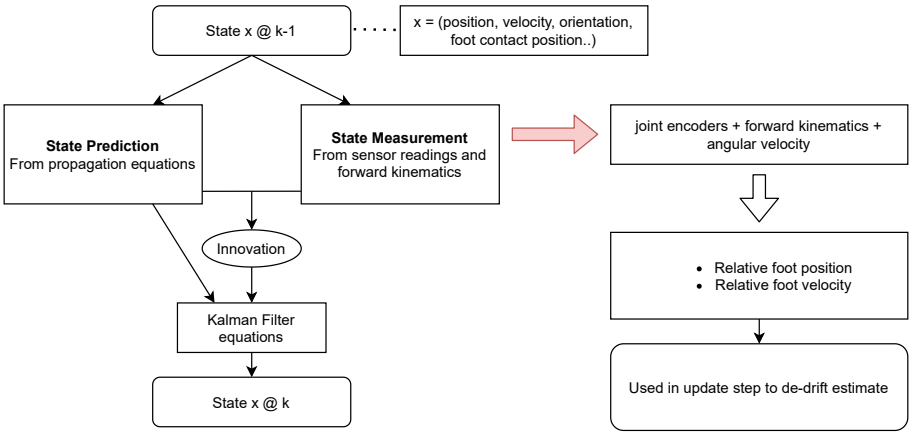

Figure 3.2: The basic KF can be extended by including relative foot position and relative foot velocity into the measurement model as proposed by Bloesch et al. [10] and Bledt et al. [6]. Here k is the discrete timestep.

The implementations developed in this project adopt these extensions, the details of which are provided in the following sections.

Including Relative Foot Position

Using a kinematic model of the robot, its relative foot positions can be calculated from the joint encoder readings. This enables for an additional measurement to be used in the update step, increasing the accuracy of the estimation [10]. An essential point of this extension is the inclusion of the foot contact positions into the filter state, such that they can be estimated alongside the pose

and are readily available to calculate the additional residual. The last element of this approach then follows naturally: while the foot contact positions are being estimated under the assumption of them staying constant, the relative foot position measurements are clearly not constant in swing phase. Bloesch et al. solve this by simply increasing the associated covariance parameter of the foot to a large value when it is in swing phase, which leads to the update from forward kinematics being skipped by the KF. When the foot regains contact with the ground, the covariance parameter is reset to a small value and the forward kinematics update is then used.

The state vector $\boldsymbol{x}$ then follows as, where $\boldsymbol{p}_i$ is the contact position of foot i and M is the number of legs:

$$\boldsymbol{x}_k = [\boldsymbol{r}_k \ \boldsymbol{v}_k \ \boldsymbol{p}_1 \ ... \ \boldsymbol{p}_M] \tag{3.3}$$

The approach leads to the following prediction equations for the *a priori* contact positions estimate $\hat{\boldsymbol{p}}_{i,k}^-$ of foot $i \in (1, ..., M)$ at timestep k:

$$\hat{\boldsymbol{p}}_{i,k}^- = \hat{\boldsymbol{p}}_{i,k-1}^+ \tag{3.4}$$

The associated innovation vector $\tilde{\mathbf{y}}_{i,k}$ is then calculated, where the subscript $_{p,i}$ denotes its calculation from the position of foot i:

$$\tilde{\boldsymbol{y}}_{i,p,k} = \tilde{\boldsymbol{s}}_{i,k} - \hat{C}_k(\hat{\boldsymbol{p}}_{i,k}^- - \hat{\boldsymbol{r}}_k^-) \tag{3.5}$$

Here $\tilde{\boldsymbol{s}}_{i,k}$ is the relative foot position measurement of foot i in body frame as calculated from kinematics. $\hat{\boldsymbol{r}}_k^-$ is the *a priori* body position estimate and thus $(\hat{\boldsymbol{p}}_{i,k}^- - \hat{\boldsymbol{r}}_k^-)$ is the estimated relative foot position in the world frame. $\hat{C}_k$ is the estimated orientation which converts the estimated relative foot position from world to body frame.

Including Relative Foot Velocity

Bledt et al. further develop this extension by adding a relative foot velocity measurement residual based on the same logic used for the relative foot position. As the foot contact position is assumed to stay constant, the foot velocity must be zero accordingly. This eliminates the need for including the foot velocity into the filter state, as the assumed relative foot velocity is then equal to the negative velocity of the body. The relative foot velocity measurement is then calculated using joint encoder readings and angular velocity measurements. The corresponding innovation vector $\tilde{\boldsymbol{y}}_{i,\hat{p},k}$ is then computed as follows, where the subscript $_{i,\dot{p}}$ denotes its calculation from the velocity of foot i:

$$\tilde{\boldsymbol{y}}_{i,\dot{p},k} = \tilde{\boldsymbol{u}}_{i,k} - \hat{C}_k(-\hat{\boldsymbol{v}}_k^-) \tag{3.6}$$

Similarly as before, $\tilde{\boldsymbol{u}}_{i,k}$ is the relative foot velocity measurement of foot i in body frame. $\hat{\boldsymbol{v}}_k^-$ is the *a priori* body velocity estimate. As the foot velocity is assumed to be constant, the estimated relative foot velocity in world frame follows as $(-\hat{\boldsymbol{v}}_k^-)$, which is again converted to body frame.

Following these extension, two different position and velocity filters are implemented in the Two Stage Estimator framework of this project: Firstly, the TS which employs only the extension using the relative foot position. Secondly, the TS FV which uses both extensions. This is done to determine any performance differences between these two approaches.

3.5 The Extended Kalman Filter

As the Two Stage Estimation approach is presented as a simpler alternative to the EKF framework commonly used for pose estimation in robotics, their respective estimation performance is naturally compared. The EKF employed for orientation, position and velocity estimation is based on the state-of-the-art implementation by Bloesch et al. [10]. The general theory of the EKF is presented in section 2.2.2. While it is used for generating results to compare to the filters developed here, the implementation of the EKF was provided for this project and was done prior to its commencement. The development or implementation of the EKF is not the subject of this thesis and is not further expanded upon.

3.6 Performance Evaluation

Due to the modularity of the Two Stage Estimation approach, performance evaluation of the different filters is split into two separate parts, namely:

1. Orientation Estimation
2. Position and Velocity Estimation

3.6.1 Orientation Estimation

In order to evaluate the orientation estimation performance, the estimates of the three different orientation estimators are compared. Namely, the Complementary Filter (CF), Unscented Kalman Filter (UKF), Unscented Kalman Filter with angular velocity estimation (UKF AV) and the Extended Kalman Filter (EKF) are tested. In this section of the evaluation, primarily the orientation estimates are of interest, position and velocity estimates are not evaluated. Further, the ground truth angular velocity as well as its computed estimate is evaluated for the case of the UKF AV in order to determine the respective estimation performance and to determine any benefits of this approach.

Testing Scenarios

To compare the orientation estimation performance of the different filters, the following two scenarios are tested. An overview is provided in Tab. 3.1.

1. Stance for 30 s
2. Counterclockwise (CCW) rotation around the y-axis for 60 s

3.6.2 Position and Velocity Estimation

The position and velocity estimation evaluation compares three different filters: The Extended Kalman Filter (EKF), the Two Stage position and velocity estimator (TS), as well as the Two Stage position and velocity estimator using relative foot velocity (TS FV). This section of the evaluation is concerned with position and velocity estimation only, thus, as the TS and TS FV can be combined with any suitable orientation filter, ground truth values are used as orientation estimates provided to the estimator. The EKF on the other hand estimates the complete pose of the robot in one combined step, making the use of ground truth orientation values unreasonable, however they are provided as measurements in the update step.

Testing Scenarios

For evaluating the position and velocity estimation performance of the different filters, the following five scenarios are tested. An overview is provided in Tab. 3.2.

1. Stance for 30 s
2. Forward motion in z for 60 s with a velocity of 0.2 $^{m}/_{s}$
3. Forward motion in x for 60 s with a velocity of 0.2 $^{m}/_{s}$
4. Sideways motion in x for 60 s with a velocity of 0.2 $^{m}/_{s}$
5. Variation in y for 30 s: $y = 0.35\,\text{m}$ at $t = 0\,\text{s}$, $y = 0.30\,\text{m}$ at $t = 10\,\text{s}$, $y = 0.37\,\text{m}$ at $t = 20\,\text{s}$

3.6.3 Testing Overview

An overview of both testing sections is given in Tab. 3.1 and Tab. 3.2. Here "✓" signifies the variables most significant for the evaluation of the respective testing scenario. Further, Fig 3.3 gives a visual overview of the different scenarios employed in simulation for estimation performance testing.

Table 3.1: Orientation Testing Scenarios and Significant Variables

Scenario	Description	Duration	Speed	Orientation			Angular Velocity (for UKF AV)		
				$Roll(R)$	$Pitch(P)$	$Yaw(Y)$	ω_x	ω_y	ω_z
1.	Stance	30s	-			✓	✓	✓	✓
2.	CCW Rotation	60s	0.1 $^{rad}/_{s}$			✓	✓	✓	✓

Table 3.2: Position and Velocity Testing Scenarios and Significant Variables

Scenario	Description	Duration	Speed	Position			Velocity		
				p_x	p_y	p_z	v_x	v_y	v_z
1.	Stance	30s	-		✓			✓	
2.	Forward in z	60s	0.2 $^{m}/_{s}$			✓			✓
3.	Forward in x	60s	0.2 $^{m}/_{s}$	✓			✓		
4.	Sideways in x	60s	0.2 $^{m}/_{s}$	✓			✓		
5.	Variation in y	30s	-		✓			✓	

3.6.4 Evaluation Metrics

Plots

A primary evaluation of the performance is done by means of plotting the estimated data for position and velocity or orientation over time, alongside the ground truth data. This is done to generate an overview of all estimator performances, as well as enabling comparisons between the different approaches.

The locomotion scenarios 2., 3. and 4. within the position and velocity testing section are further evaluated by means of plotting a trajectory map using the x and z position estimates and ground truth values. This is done to generate a visual representation of the position estimation deviations during motion.

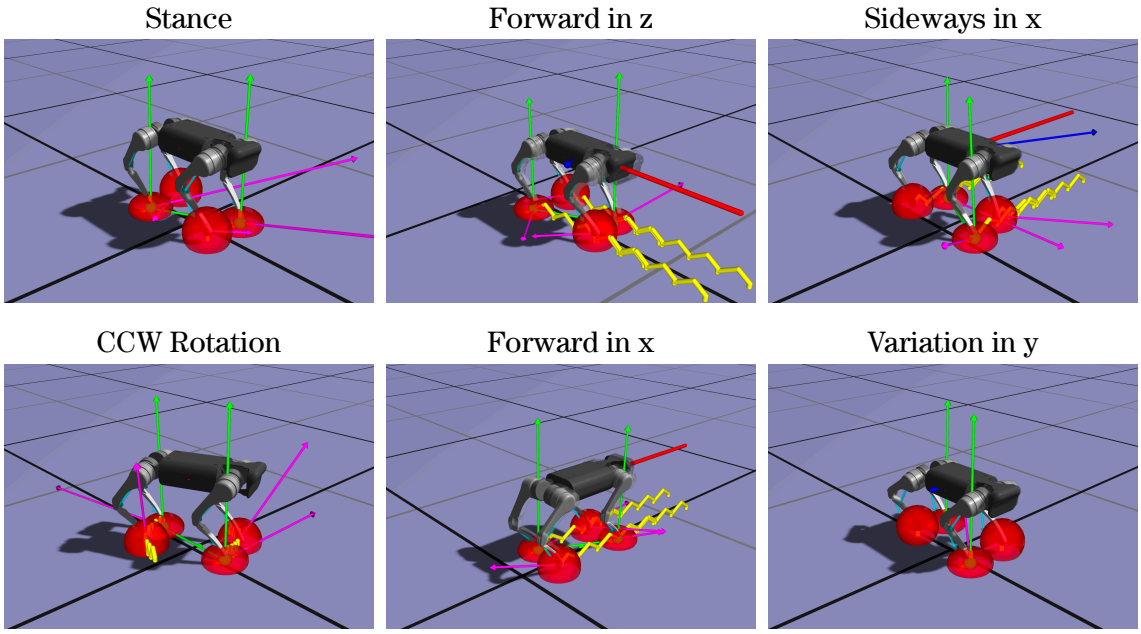

Figure 3.3: An overview of the scenarios used for testing orientation, as well as position and velocity estimation of the different filters.

Root-Mean-Square Error (RMSE)

To quantify the accuracy of the estimation, the root-mean-square error (RMSE) is calculated for all estimated time series. The RMSE is computed as follows, where N is the total number of data points collected, $\hat{x}_i$ is the estimated value and x_i is the ground truth value [35]:

$$RMSE(x, \hat{x}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{x}_i - x_i)^2} \tag{3.7}$$

The RMSE is chosen as it is an excellent error metric for numerical-predictions and suitable for performance comparisons between different prediciton or estimation models [13, 35]. It should be noted however, that comparisons of estimation performance between different variables, for example velocity versus position estimation accuracy, can not be done by means of RMSE evaluation. This is due to the scale dependency of the RMSE [35]. A perfect estimation would result in an RMSE of zero, meaning no deviation of the estimation from the ground truth is present.

Maximum Absolute Error (MaAR)

The maximum absolute error (MaAR) computes the largest estimation error present in the data set, representing a metric for quantifying the worst-case performance of the estimator. The MaAR is calculated as follows, where again $\hat{x}_i$ is the estimated value and x_i is the ground truth value:

$$MaAR(x, \hat{x}) = \max(|\hat{x}_i - x_i|) \tag{3.8}$$

The MaAR is chosen to allow evaluation not only of the average performance of the estimator, but to specifically quantify its most unfavorable estimations and thus its stability.

Final Position Error Versus Final Distance Travelled

The final position error versus final distance travelled (FEFD) computes the final position estimation error in percent of the final distance the ground truth robot travels in the desired direction specified by the scenario. The FEFD is calculated as follows, where $\hat{x}_N$ and x_N are the final values of the data set $(x, \hat{x})$ with data points $(x_i, \hat{x}_i)$ ranging from $i = 1...N$:

$$FEFD(x, \hat{x}) = \frac{|\hat{x}_N - x_N|}{|x_N|} * 100 \tag{3.9}$$

The FEFD quantifies the final estimation error in terms of the robot's objective of the corresponding scenario, ie. the distance it travels in the desired direction. The FEFD thus provides a graspable metric for the final position drift of the estimator. It should be noted that, due to the nature of the FEFD, it is only used for evaluation of position estimation in forward and sideways motion scenarios.

Chapter 4

Results

4.1 Simulation

In this section, the simulation environment in which the different estimators were tested is described. Especially the models used for generating sensor noise and biases are elucidated. A brief explanation of the environment and robot setup is also given.

4.1.1 Noise and Biases

Gyroscope and Accelerometer

To simulate and test the performance of the filters in hardware deployment, the challenges presented by legged robot state estimation must be considered. This means that the inevitable innaccuracies of the gyroscope and accelerometer must be included in the simulation by means of noise and bias generation. These disturbances must then be added to the accelerometer and gyroscope values provided to the estimation algorithm. The values used to generate these disturbances are based on hardware parameters, rendering a physically relevant model.

The following equations describe the disturbance process to which the gyroscope and accelerometer values adhere to [10]:

$$\tilde{\boldsymbol{f}} = \boldsymbol{f} + \boldsymbol{b}_f + \boldsymbol{w}_f \tag{4.1}$$

$$\dot{\boldsymbol{b}}_f = \boldsymbol{w}_{bf} \tag{4.2}$$

$$\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} + \boldsymbol{b}_\omega + \boldsymbol{w}_\omega \tag{4.3}$$

$$\dot{\boldsymbol{b}}_\omega = \boldsymbol{w}_{b\omega} \tag{4.4}$$

Here $\boldsymbol{f}$ and $\boldsymbol{\omega}$ are the actual accelerometer and gyroscope values, $\boldsymbol{w}_f$ and $\boldsymbol{w}_\omega$ are additive white Gaussian noise processes and $\boldsymbol{b}_f$ and $\boldsymbol{b}_\omega$ are bias terms. The bias terms, as becomes evident from the schematic, are modeled as Brownian motions and their corresponding derivatives are again described by white Gaussian noise processes $\boldsymbol{w}_{bf}$ and $\boldsymbol{w}_{bw}$ [10]. $\tilde{\boldsymbol{f}}$ and $\tilde{\boldsymbol{\omega}}$ are the resulting disturbed accelerometer and gyroscope values, which are provided to the state estimator in the simulation.

The noise values are defined by their respective standard deviation terms which are specific to the IMU employed. This simulation assumes values from an InvenSense MPU9150 IMU [22], for which the white noise strength σ_f (accelerometer) and σ_ω (gyroscope), as well as the bias diffusion values σ_{bf} (accelerometer) and $\sigma_{b\omega}$ (gyroscope) are provided readily in [36] and displayed in Tab. 4.1.
In order to obtain the standard deviation needed to generate the respective noise and bias values, these white noise strengths and bias diffusion values must first be discretized. This can be done by taking the IMU sampling rate $f_{IMU} = \frac{1}{\Delta t}$ into consideration. A value of $f_{IMU} = 200\,\text{Hz}$ is assumed

for this simulation [27]. The discretization follows a simple procedure [40], where σ_c denotes the continuous-time value (from Tab. 4.1) and σ_d the corresponding disctetized value:

$$\sigma_d = \sigma_c \sqrt{f_{IMU}} = \frac{\sigma_c}{\sqrt{\Delta t}} \tag{4.5}$$

The generation of the discrete noise terms follows accordingly [2], where k is the discrete time step, $\boldsymbol{w}_d[k]$ is the discrete noise and σ_d is the discrete standard deviation:

$$\boldsymbol{w}_d[k] = \boldsymbol{g}[k]\sigma_d \tag{4.6}$$

$$\boldsymbol{g}[k] = [g_x \; g_y \; g_z] \quad \text{with } g_i \sim \mathcal{N}(0,1) \quad \text{for } i \in (x,y,z) \tag{4.7}$$

In the case of bias propagation, a subsequent additional integration step is required which, in discrete time, follows accordingly [14]. Here $\boldsymbol{b}_d[k]$ is the discrete bias, $\sigma_{b,d}$ the discrete bias noise, $\boldsymbol{b}_d[k-1]$ the bias of the previous time step and $\Delta t = \frac{1}{f_{IMU}}$ the sampling time step:

$$\boldsymbol{b}_d[k] = \boldsymbol{b}_d[k-1] + \Delta t \boldsymbol{w}_d[k] \tag{4.8}$$

As becomes evident from eq. (4.5) and eq. (4.8), after discretization and after subsequent integration in the case of the biases, the units of the noise and bias terms are again conform with the corresponding measurement values. The discretized parameters can be found in Tab. 4.2.

Joint Encoders

Several of the proposed estimation frameworks, namely the EKF, TS and TS FV, make use of the relative foot position (and relative foot velocity in the case of the TS FV). These values are computed by means of forward kinematics, using the measurements provided from joint encoders employed in the robot [6, 10]. To again emulate relevant measurements in simulation, noise values must generated and added to the computed joint encoder readings. The following equations describe the process from which the joint position and velocity values are obtained [10]:

$$\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha} + \boldsymbol{n}_\alpha \tag{4.9}$$

$$\tilde{\dot{\boldsymbol{\alpha}}} = \dot{\boldsymbol{\alpha}} + \boldsymbol{n}_{\dot{\alpha}} \tag{4.10}$$

Here $\boldsymbol{\alpha}$ and $\dot{\boldsymbol{\alpha}}$ are the actual joint position and velocity values respectively. The disturbances to the actual values are modeled as discrete Gaussian noise $\boldsymbol{n}_\alpha$ and $\boldsymbol{n}_{\dot{\alpha}}$ [10]. Again, these noise values are specified by the corresponding standard deviations σ_α and $\sigma_{\dot{\alpha}}$, which in turn depend on the specific joint encoder used. The values assumed in the simulation can be found in Tab. 4.2 and are adopted from the implementation of Bledt et al. [6].

The noise generation follows analogously as in eq. (4.6), only here the values are already assumed to be discrete, thus the discretization becomes superfluous:

$$\boldsymbol{n}_d[k] = \boldsymbol{g}[k]\sigma_d \tag{4.11}$$

$$\boldsymbol{g}[k] = [g_x \; g_y \; g_z] \quad \text{with } g_i \sim \mathcal{N}(0,1) \quad \text{for } i \in (x,y,z) \tag{4.12}$$

Table 4.1: MPU9150 IMU Parameters [36]

Metric	Symbol	Value	Unit
Accelerometer			
White noise strength	σ_f	2.24×10^{-3}	$\mathrm{m/s^2\sqrt{Hz}}$
Bias Diffusion	σ_{bf}	7.53×10^{-5}	$\mathrm{m/s^3\sqrt{Hz}}$
Gyroscope			
White noise strength	σ_ω	1.84×10^{1}	$^\circ/\mathrm{h}\sqrt{\mathrm{Hz}}$
Bias Diffusion	$\sigma_{b\omega}$	1.08×10^{-5}	$\mathrm{rad/s^2\sqrt{Hz}}$

Table 4.2: Simulation Noise and Bias Parameters (Discrete Values) [36, 6]

Metric	Symbol	Value	Unit
Accelerometer			
Noise Std.	$\sigma_{f,d}$	3.17×10^{-2}	$\mathrm{m/s^2}$
Bias Noise Std.	$\sigma_{bf,d}$	1.06×10^{-3}	$\mathrm{m/s^3}$
Gyroscope			
Noise Std.	$\sigma_{\omega,d}$	1.26×10^{-3}	$\mathrm{rad/s}$
Bias Noise Std.	$\sigma_{b\omega,d}$	1.53×10^{-4}	$\mathrm{rad/s^2}$
Joint Encoders			
Position Noise Std.	σ_α	1×10^{-3}	rad
Velocity Noise Std.	$\sigma_{\dot{\alpha}}$	1×10^{-1}	$\mathrm{rad/s}$

4.1.2 Robot and Simulation Setup

To evaluate the performance of the estimation approaches, they are employed in simulation and the scenarios detailed in section 3.6 are tested. The simulation environment is provided by the Computational Robotics Lab (CRL) at ETHZ. The environment enables simulation of different robots, of which the Unitree A1 quadruped is used for this evaluation. Further, different gaits can be selected within the environment. For the employed experiments, a trotting gait is selected. The desired control inputs can be set in the simulation environment and the data required for evaluation is also exported directly by an extension created in the process of this project.

4.2 Experiment Results

In this section, the results from the test scenarios discussed in section 3.6 are presented. First, the results obtained from the orientation experiments are given and subsequently the results from position and velocity experiments are presented. The results consist of the evaluation metrics presented in section 3.6.4.

4.2.1 Orientation Experiment Results

Tab. 4.3 shows the values of the RMSE and MaAR metrics as defined in section 3.6.4 for all filters and both orientation experiment scenarios. Also, the RMSE of the angular velocity estimation for the UKF AV approach is shown. The lowest error values per column, ie. the most accurate estimation performance for the respective variable, are displayed in bold, allowing for a vertical comparison between the different filters.

1. Stance

Fig. 4.1 compares the yaw estimate (blue) to the ground truth data (red) for all filters for the stance scenario. The accuracy achieved in simulation becomes apparent, with all estimators achieving a rounded RMSE value of 0.000 rad. Taking all orientation angles into consideration as displayed in Tab. 4.3, similarly low RMSE and MaAR values can be determined. The EKF offers the best performance throughout all metrics, however the values differ only minimally between the filters. Further notable is the lack of performance difference between the UKF and the UKF AV, the only deviation being the minimally higher MaAR (0.001 rad) in the roll angle estimation for the UKF AV. In terms of angular velocity estimation, the UKF AV offers estimates with a RMSE below $0.004\,{}^{\mathrm{rad}}/_{\mathrm{s}}$ for all axis.

2. CCW Rotation

Fig. 4.2 compares the yaw estimate (blue) to the ground truth data (red) for all filters for the CCW rotation scenario. Differences in the estimation performance between the filters becomes evident when the RMSE is considered. The RMSE for the CF (0.098 rad) lies slightly below the RMSE for both the UKF and the UKF AV (0.105 rad), with a difference of only 0.007 rad. The RMSE for the EKF (0.427 rad) is larger by a factor of ~ 4. When regarding the other orientation angles, only minimal differences between the performance of the different filters can be determined, with the maximum RMSE being 0.004 rad for the CF. Notably, the MaAR for the yaw angle (6.276 - 6.283 rad) in this scenario is considerably larger than the one in the stance scenario (0.001 rad). Again, no improvement in performance of the UKF AV over the UKF can be determined. Lastly, the angular velocity estimation performance by the UKF AV is on par with the stance scenario with RMSE values below $0.004\,{}^{\mathrm{rad}}/_{\mathrm{s}}$ for all axis.

		Orientation R, P, Y						**Angular Velocity** ω_x, ω_y, ω_z		
	Filter / Metric	RMSE [rad]			MaAR [rad]			RMSE [rad/s]		
1. Stance	CF	0.001	0.004	**0.000**	0.002	0.005	**0.001**			
	UKF	0.001	0.001	**0.000**	0.002	0.002	**0.001**			
	UKF AV	0.001	0.001	**0.000**	0.003	0.002	**0.001**	0.002	0.001	0.004
	EKF	**0.000**	**0.000**	**0.000**	**0.001**	**0.001**	**0.001**			
2. Rotation	CF	0.004	0.004	**0.098**	0.008	0.009	6.283			
	UKF	0.001	0.001	0.105	**0.001**	0.002	6.282			
	UKF AV	0.001	0.001	0.105	**0.001**	0.002	6.282	0.002	0.001	0.004
	EKF	**0.000**	**0.000**	0.427	**0.001**	**0.001**	**6.276**			

Table 4.3: Orientation Results: Error Results for all filters in both orientation experiment scenarios. The RMSE and MaAR are displayed for the estimated roll, pitch and yaw angles. The RMSE is displayed for the angular velocity estimated by the UKF AV. The lowest values per column are displayed in bold.

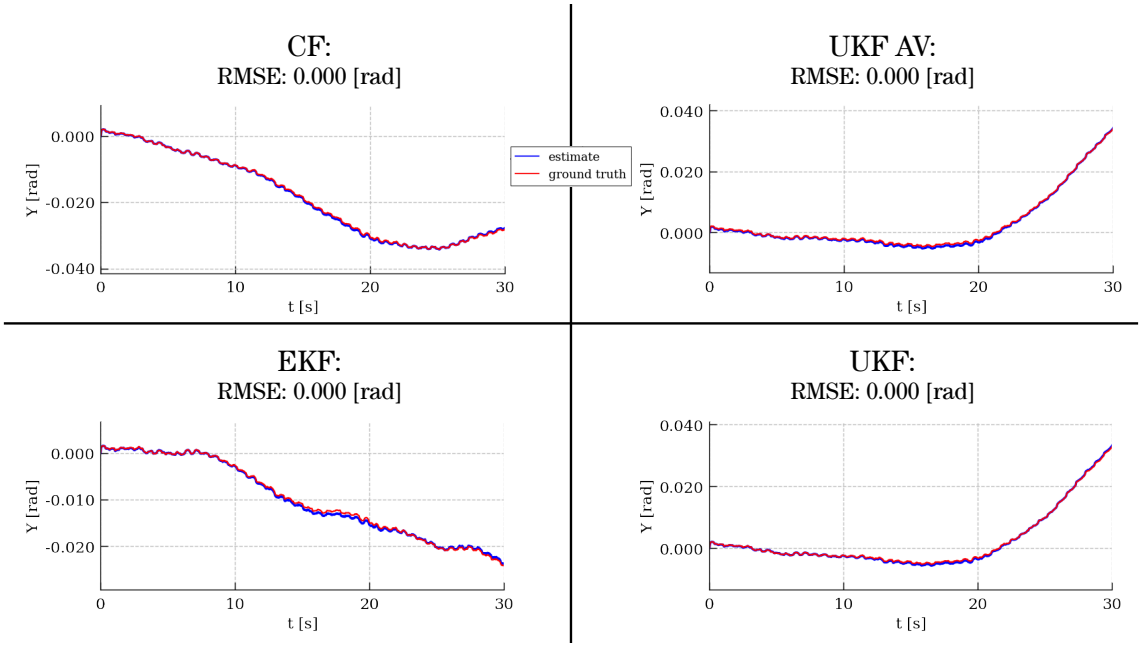

Figure 4.1: 1. Stance: Comparison between estimated yaw angles (blue) and ground truth data (red) in [rad] which are plotted over time in [s]. RMSE values are given for each filter, with all showing results at an RMSE of 0.000 rad.

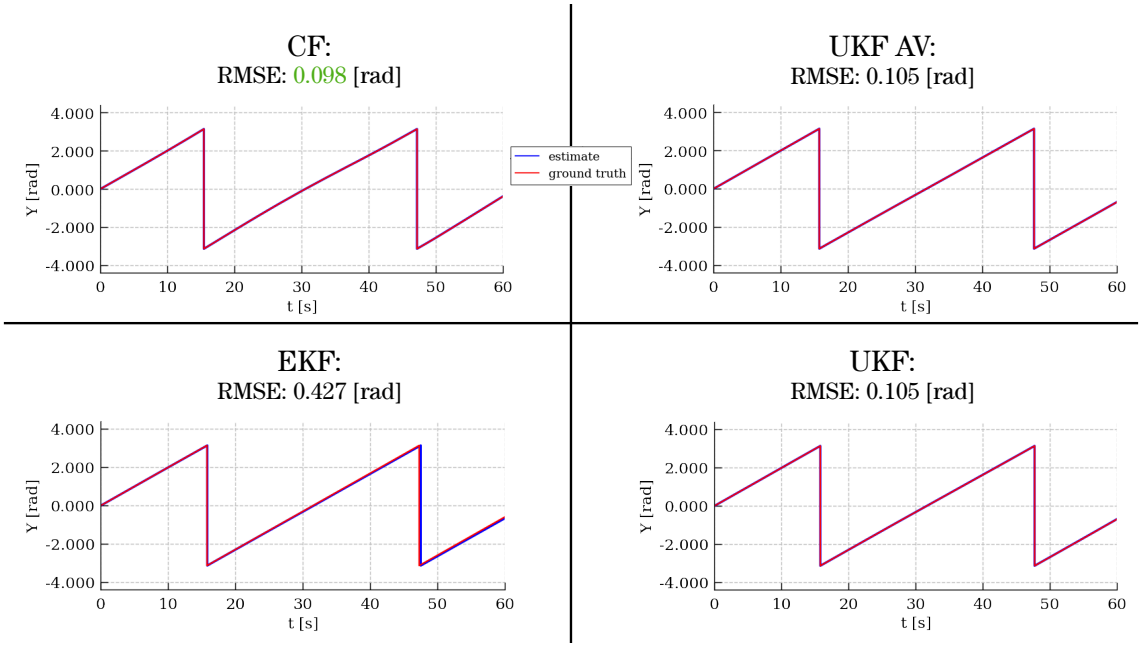

Figure 4.2: 2. CCW Rotation: Comparison between estimated yaw angles (blue) and ground truth data (red) in [rad] which are plotted over time in [s]. RMSE values are given for each filter, with the lowest (green) being 0.098 rad for the CF.

4.2.2 Position and Velocity Experiment Results

Tab. 4.4 and Tab. 4.5 show the error metrics for position and velocity estimation respectively, resulting from the position and velocity experiment scenarios defined in section 3.6.2. For the x, y and z position estimates, the RMSE and MaAR are provided. Further, for the scenarios 2., 3. and 4., when the robot performs locomotion, the FEFD is provided for the direction of travel. In the case of the velocity estimates v_x, v_y and v_z, both the RMSE and MaAR are shown. Again, the lowest value per column is displayed in bold, highlighting the best filter performance for the respective variable.

1. Stance

From Fig. 4.3 the y position and velocity estimation performance in stance can be determined. The low RMSE values show the accuracy of all filters at this task. This sentiment is resonated when considering the error metrics provided in Tab. 4.4 and Tab. 4.5. Further, the velocity estimation RMSE is similarly low for all filters across all variables. Comparing the x and z position estimation performance however, some distinction between the filters becomes apparent. For x position estimation, the TS FV outperforms the EKF and the TS in terms of both RMSE (0.003 m) and MaAR (0.006 m). The inverse situation can be determined for the z position estimation performance, where both the TS and the EKF provide similarly accurate estimates and outperform the TS FV in terms of RMSE and MaAR.

2. Forward in z

In locomotion, the estimation performance of the filters show stronger distinctions as in stance mode. From Fig. 4.4 this becomes apparent for the estimation of the z position and velocity. While the EKF and the TS provide similarly accurate results in terms of the RMSE, the TS FV performs decidedly worse in both tasks. Especially in the position estimation, the RMSE for the TS FV (0.714 m) is 0.122 m higher than that of the TS (0.592 m). The additional metrics provided in Tab. 4.4 follow the same narrative with both MaAR and FEFD being clearly higher in the case of the TS FV. Notably, the FEFD for the TS FV is 9.87 % while that of the TS is only 8.4 %. This distinction is further made visible in the trajectory comparison offered in Fig. 4.8 where a clear difference in performance between the TS FV and the TS or EKF is apparent.

3. Forward in x

Under motion in x direction, the estimation performance of the filters is almost identical to the previous scenario, only with the z and x variables switched. This comes as no surprise, as the two scenarios performed are indistinguishable in nature with the sole difference being the change in trajectory direction. This replicability is underlined by both the Fig. 4.5 and Fig. 4.8, as well as by the data provided in Tab. 4.4 and Tab. 4.5.

4. Sideways in x

The ability of the estimators to function in sideways motion is shown in Fig. 4.6, where the x position and velocity estimation performance of the filters is compared. A similar pattern to scenario 2. becomes evident, with both the TS and EKF outperforming the TS FV for the x and z variables. The values provided for the position estimation performance in Tab. 4.4 strengthen this sentiment, with the TS FV exhibiting distinctly higher RMSE and MaAR values for the x and z variables. Again, the FEFD comparison reflects this observation, resulting in 9.02 % for the TS FV versus 6.27 % for the EKF. This difference is also apparent when evaluating the trajectory plotted in Fig. 4.8 where the EKF and TS show almost identical performance while drift in the TS FV estimation is evident.

5. Variation in y

The results of the final scenario, which aims at comparing the filters y position and velocity estimation performance when varying the body height of the robot, can be viewed in Fig. 4.7. The plots showcase great accuracy across all estimators, which is further underlined by the low RMSE values. This stability is resonated by the y variable position and velocity error metrics shown in Tab. 4.4 and Tab. 4.5. The RMSE for y estimation peaks at 0.001 m for position (TS) and 0.006 $^{m}/_{s}$ for velocity (TS and TS FV).

		Position Error Metrics for x, y, z							
	Filter / Metric	RMSE [m]			MaAR [m]			FEFD [%]	
1. Stance	TS	0.021	0.001	0.008	0.036	0.003	0.014		
	TS FV	**0.003**	**0.000**	0.019	**0.006**	0.002	0.038		
	EKF	0.020	0.001	**0.007**	0.035	**0.001**	**0.013**		
2. Forward in z	TS	**0.032**	**0.000**	**0.592**	**0.052**	0.003	**1.025**		**8.40**
	TS FV	0.062	**0.000**	0.714	0.097	0.002	1.242		9.87
	EKF	0.034	**0.000**	0.593	0.054	**0.001**	1.032		8.41
3. Forward in x	TS	**0.591**	**0.000**	**0.032**	**1.025**	0.003	**0.052**	**8.39**	
	TS FV	0.707	**0.000**	0.063	1.229	0.002	0.099	9.78	
	EKF	0.593	**0.000**	0.034	1.032	**0.001**	0.054	8.41	
4. Sideways in x	TS	0.101	0.001	**0.017**	0.172	0.003	**0.024**	6.61	
	TS FV	0.172	**0.000**	0.104	0.303	0.002	0.148	9.02	
	EKF	**0.092**	0.001	0.022	**0.159**	**0.001**	0.033	**6.27**	
5. Variation in y	TS	0.033	0.001	0.009	0.054	0.003	0.017		
	TS FV	**0.009**	**0.000**	0.020	**0.016**	**0.002**	0.044		
	EKF	0.034	**0.000**	**0.008**	0.057	**0.002**	**0.015**		

Table 4.4: Position Results: Position error results for all filters in all position and velocity experiment scenarios. The RMSE and MaAR are displayed for the estimated x, y and z positions. The FEFD is given for all locomotion scenarios in the direction of travel. The lowest values per column are displayed in bold.

		Velocity Error metrics for v_x, v_y, v_z					
	Filter / Metric	RMSE [m/s]			MaAR [m/s]		
1. Stance	TS	**0.002**	0.006	0.002	0.015	0.209	0.050
	TS FV	**0.002**	0.005	0.003	0.009	**0.013**	**0.007**
	EKF	**0.002**	**0.001**	**0.001**	**0.004**	0.022	0.008
2. Forward in z	TS	**0.001**	0.004	**0.017**	0.016	0.208	0.048
	TS FV	0.003	0.004	0.024	0.010	0.013	0.032
	EKF	**0.001**	**0.001**	0.018	**0.008**	**0.021**	**0.024**
3. Forward in x	TS	**0.017**	0.004	**0.001**	0.048	0.208	0.018
	TS FV	0.024	0.004	0.003	0.032	**0.013**	0.010
	EKF	0.018	**0.001**	**0.001**	**0.024**	0.021	**0.008**
4. Sideways in x	TS	**0.003**	0.004	**0.002**	0.014	0.209	0.048
	TS FV	0.007	0.005	0.006	**0.013**	**0.013**	0.015
	EKF	**0.003**	**0.001**	0.003	**0.013**	0.023	**0.013**
5. Variation in y	TS	**0.002**	0.006	0.002	0.015	0.209	0.050
	TS FV	**0.002**	0.006	0.003	0.008	**0.013**	**0.008**
	EKF	0.003	**0.003**	**0.001**	**0.007**	0.093	0.012

Table 4.5: Velocity Results: Velocity error results for all filters and all position and velocity experiment scenarios. The RMSE and MaAR are displayed for the estimated v_x, v_y and v_z positions. The lowest values per column are displayed in bold.

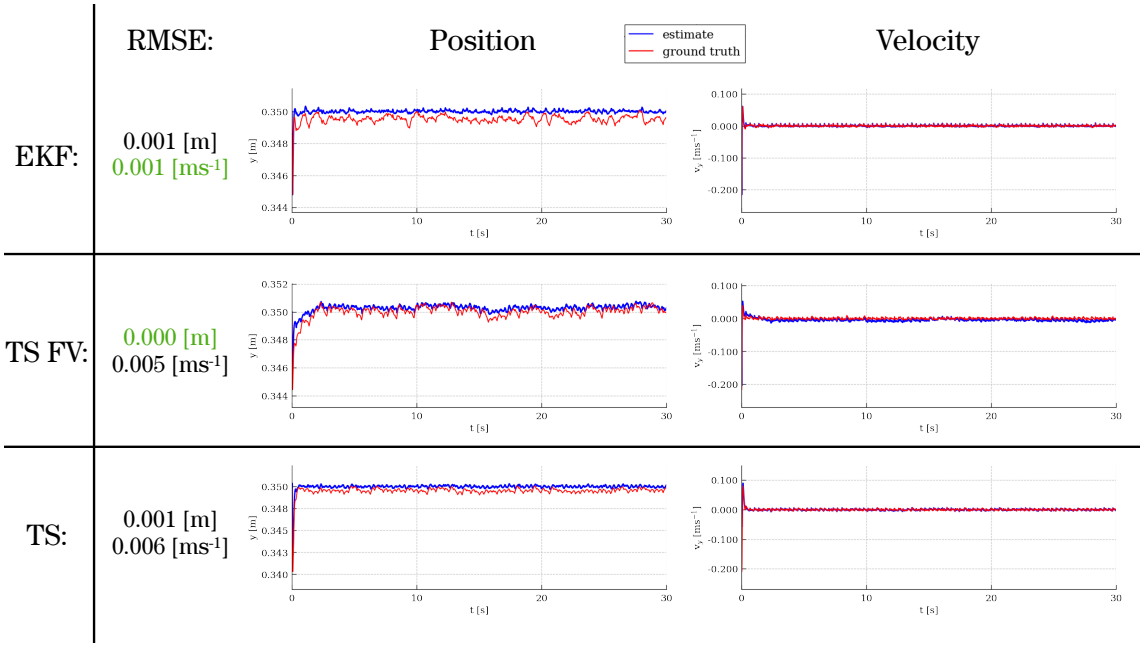

Figure 4.3: 1. Stance: Comparison between estimated y position and velocity (blue) and ground truth data (red) in [m] and [m/s] which are plotted over time [s]. RMSE values are given for each filter, with the lowest highlighted in green.

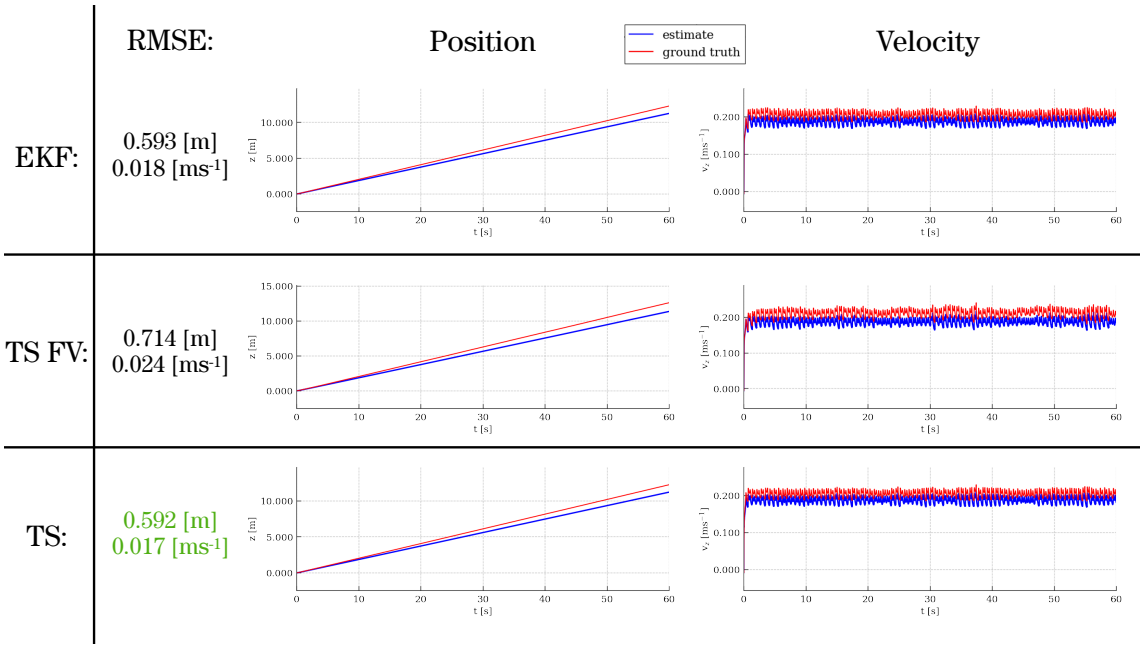

Figure 4.4: 2. Forward in z: Comparison between estimated z position and velocity (blue) and ground truth data (red) in [m] and [m/s] which are plotted over time in [s]. RMSE values are given for each filter, with the lowest highlighted in green.

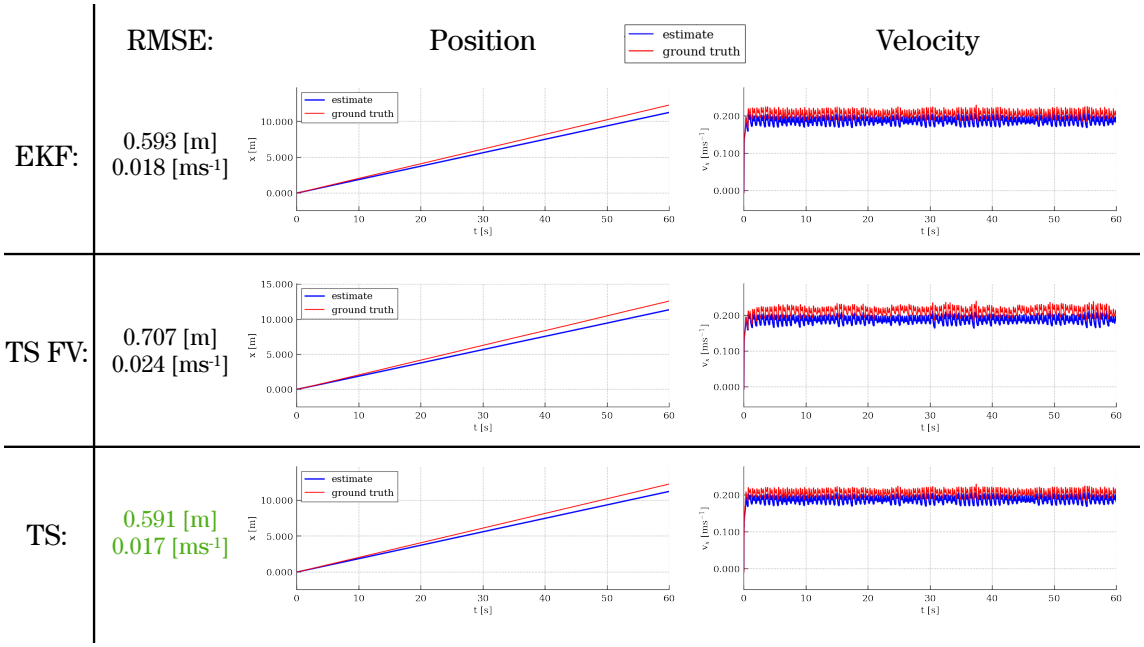

Figure 4.5: 3. Forward in x: Comparison between estimated x position and velocity (blue) and ground truth data (red) in [m] and [m/s] which are plotted over time in [s]. RMSE values are given for each filter, with the lowest highlighted in green.

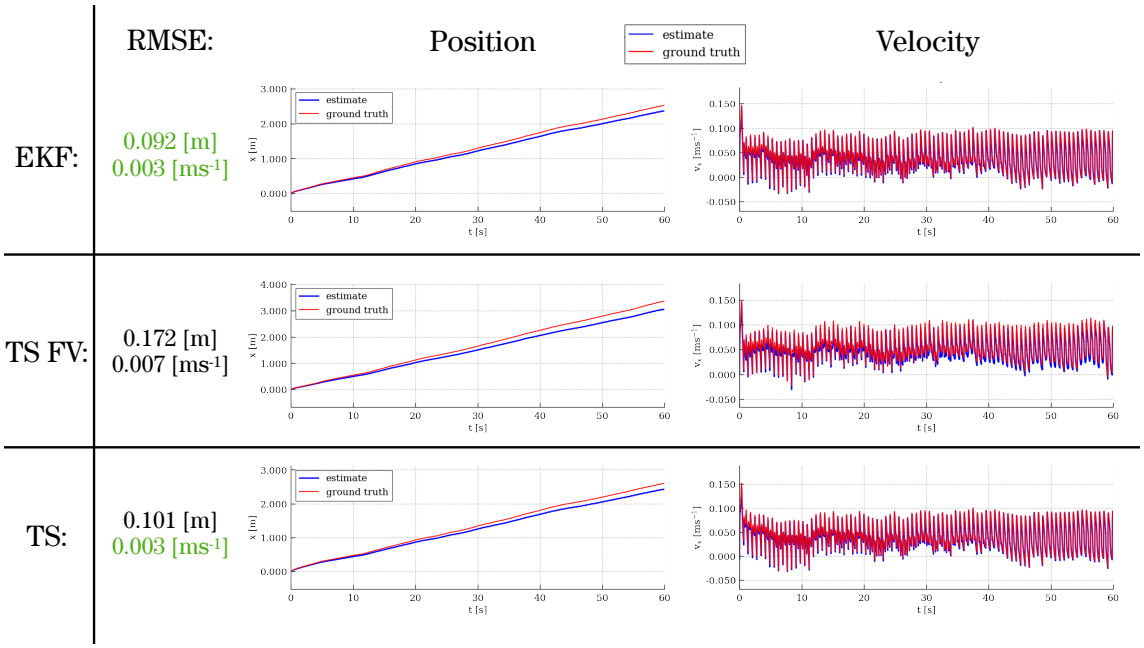

Figure 4.6: 4. Sideways in x: Comparison between estimated x position and velocity (blue) and ground truth data (red) in [m] and [m/s] which are plotted over time in [s]. RMSE values are given for each filter, with the lowest highlighted in green.

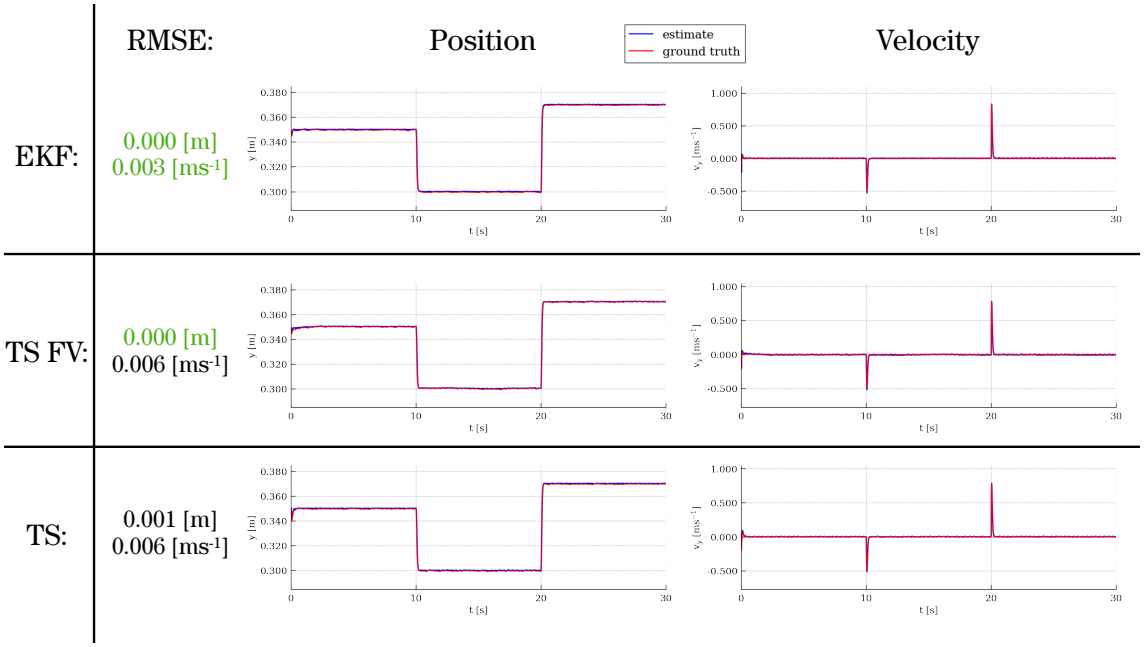

Figure 4.7: 5. Variation in y: Comparison between estimated y position and velocity (blue) and ground truth data (red) in [m] and [m/s] which are plotted over time in [s]. RMSE values are given for each filter, with the lowest highlighted in green.

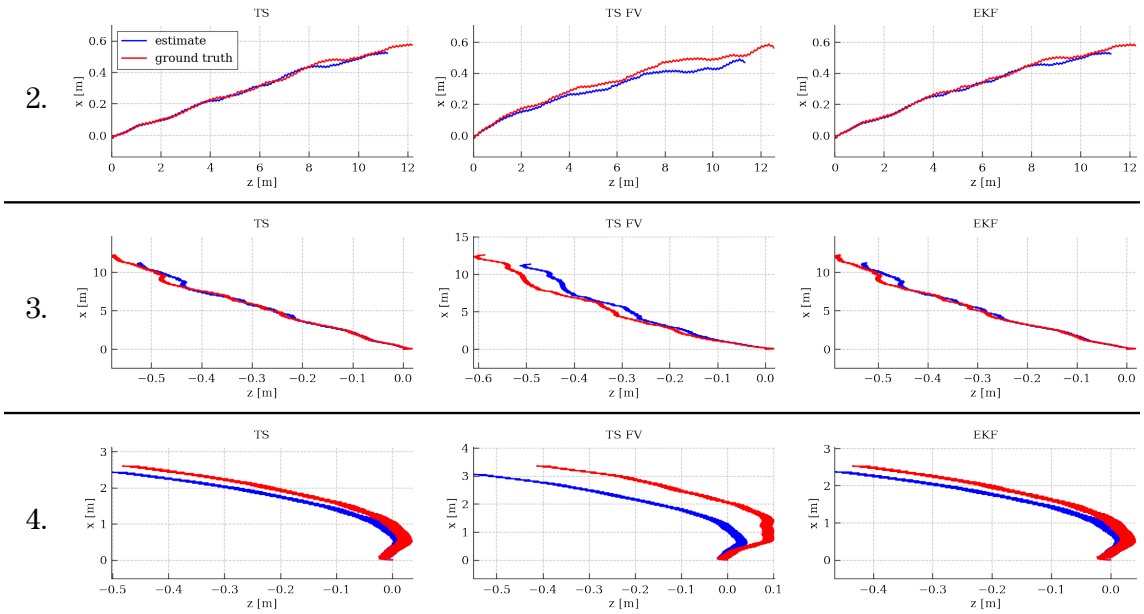

Figure 4.8: Trajectory map in z and x in [m] for the three locomotion scenarios (2., 3. and 4.). The estimated trajectory is plotted in (blue) and the ground truth data in red for all three position and velocity filters.

Chapter 5

Discussion

Building on previous work, the objective of this thesis was to implement and evaluate different approaches to state estimation for legged robots. The challenges lie in the inherent nonlinearity of orientation estimation, as well as the corruption of measurements by sensor noise and biases, as is expanded upon in section 2.1.3. The success of an estimation approach is dependent on its ability to handle these tasks well and produce estimates with high accuracy. At the same time, this thesis further aims at achieving preferably low complexity of implementation. Derived from these criteria, the performance evaluation of a Two Stage Estimation approach combining an augmented Classical Kalman Filter with different orientation filters was focused on. A state-of-the-art Extended Kalman Filter approach was further employed to offer comparable values for the determined testing scenarios. The scenarios were tested in simulation using measurements disturbed by noise and biases to provide realistic estimation results.

In the following, the performance of the implemented filters is discussed and compared based on the results presented in the prior section. Further, the limitations of these results obtained in simulation is reflected upon.

5.1 Orientation Estimation

Accuracy

The results that were achieved in section 4.2.1 by all four employed filters clearly show their capability of providing accurate orientation estimates. Across the board, all filters performed within equal ranges of precision, except for yaw angle estimation where the CF and both UKFs showed a slight edge over the EKF. However, it should be noted that, as Bloesch et al. [10] discuss in their proposal of the EKF, the yaw angle remains unobservable and thus estimation drift in this variable was to be expected.

While the CF was expected to perform worse than the other more sophisticated filters, this was the case only on a minimal scale, with the CF even outperforming all other filters in yaw estimation for the rotation scenario. Its approach of efficiently combining the gyroscope and accelerometer readings with only one parameter offered for tuning performed impressively well. It should be noted however, that the accurate calculation of this parameter was only possible due to the prior knowledge of the gyroscope measurement error, which might not apply to all scenarios of employment in real hardware.

On the other hand, the EKF was expected to perform better than the other filters, due to its incorporation of the sensor biases into the filter state [10]. This allows the EKF to attempt to compensate any biases present in the intertial measurements, which could be expected to lead to improved estimates as compared to other filters without bias compensation. This was not the case,

suggesting either impressive performance by the other filters or the need for further parameter tuning within the EKF implementation.

Proposed in this project as an alternative to the EKF to avoid cumbersome linearization, the UKF showed very promising results in terms of accuracy. The scenarios tested showed no disadvantages in its approach as compared to the two more common orientation estimation techniques. Further, by comparing the results of the UKF and the UKF AV, it becomes clear that, at least in simulation, the UKF AV shows no benefits over the UKF. This would indicate that the simpler UKF implementation, neglecting angular velocity estimation, is sufficient to achieve accurate results. At this point, real hardware experiments are needed to confirm these observations.

Implementation Complexity

In terms of complexity of implementation, the EKF is clearly the least preferable option. The necessity of linearization for the approach is cumbersome and hampers the process of implementation and debugging. The CF on the other hand offers a compact solution which can easily be adapted and implemented into most any orientation estimation scheme. The minimal differences in accuracy discussed above do not justify this large difference in implementation complexity.

However, as the CF is a very compact solution, it can hardly be extended to improve estimation accuracy if needed. Here the EKF clearly offers more possibilities in terms of parameter tuning, bias estimation and other augmentations to the estimation process. To bridge this gap between high adaptability and relatively low complexity, the UKF was proposed as an alternative approach to orientation estimation. While it is not as simple as the CF, the sampling and propagating of selected sigma points produces a more sophisticated algorithm than the CF, with simultaneously less complexity than the EKF. At the same time, the adaptability of a Kalman Filtering approach is maintained, offering possible state extensions or parameter and sampling scheme tuning. These benefits, along with the achieved accuracy, identify the UKF as the preferable orientation estimation approach by the standards set and scenarios tested in this thesis.

5.2 Position and Velocity Estimation

Accuracy

The stance and height variation experiments tested in simulation give evidence of the high accuracy of all position and velocity filters for these tasks. Further, all filters provided rather smooth estimates across all tested scenarios, with no apparent discontinuities visible in the resulting plots. The locomotion experiments however helped determine some differences in precision. Surprisingly, the TS FV performed considerably worse than the two other filters which use only the relative foot position extensions to the update step. The drift accumulated in the position estimates for the TS FV in forward and sideways motion indicate that the additional inclusion of relative foot velocity into the model does not improve the estimate. On the contrary, it results in declined accuracy for the scenarios tested in this project. This observation could imply that relative foot velocity as calculated by forward kinematics is not a valuable additional measurement for pose estimation. However, as a successful implementation exists [6], it rather implies the necessity of either tuning or adapting the process noise model employed for this update component.

The almost identical results achieved for scenarios 2. and 3. provide evidence for the rotational invariance of the filters employed. Changing the trajectory direction from z to x and initializing the robot facing the x direction does not result in any changes in the estimation performance of the filters, which is in accordance with noise and bias generation model used.

Lastly, the very similar performance of the TS and the EKF validates the TS as an viable alternative to the EKF approach. While an improvement of the TS over the EKF was expected, since no

linearization is involved in the estimates provided by the TS, this could not be ascertained in this project.

Implementation Complexity

As mentioned throughout this thesis and discussed in section 3.2, the Two Stage Estimation approach allows for using a Classical Kalman Filter for estimating position and velocity. There is no doubt that the implementation of the TS and TS FV is of lower complexity than that of an EKF. A reduced state vector, as the orientation is estimated separately, no linearization and accordingly no Jacobian calculation lead to an objectively more straightforward filter algorithm.

Further, the TS and TS FV can be paired with any orientation filter, due to the employed Two Stage Estimation approach. This enables greater modularity, allowing for switching between different orientation filters either for purposes of research or to find a combination optimal to the desired application. Such a differentiation is not possible when employing a filter that does not separate these two distinct steps of the estimation process, as was the case for the EKF employed.

As was the case with the UKF, the higher complexity of the EKF is not justified by the comparison between the estimation performance of the EKF and the TS. The TS provides estimates with accuracy on par with the EKF in all scenarios, while being considerably simpler and modular. These benefits emphasize the TS as the preferable choice for position and velocity estimation.

5.3 Limitations

The results, upon which the observations and statements made in this section are made, were obtained in simulation. A realistic noise and bias model was employed that, based on hardware parameters, aimed at generating relevant measurements. However, the results were never confirmed by real hardware testing, which does introduce some limitations into the evaluation process.

The terrain in the simulation environment was even and no foot slippage occurred. Further, apart from the noise and biases, no external disturbances were inflicted upon the robot. Also, the speed with which the locomotion tasks were performed was kept relatively low at $0.2\,^{\mathrm{m}}/_{\mathrm{s}}$ and no highly dynamic gaits were tested.

For testing position and velocity estimation performance, ground truth orientation data was provided to the TS and TS FV, as well as in the measurement update of the EKF, as was mentioned in section 3.6.2. This is surely not possible in hardware experiments and thus does not extensively represent a realistic testing scenario. While the main objective of these experiments was definitely the position and velocity estimation performance, using an orientation estimate from a nonlinear filter instead of ground truth data is a logical next step to validate the results.

Nevertheless, these limitations certainly do not invalidate the observations made, as all filters were tested within the same environment and the comparison between the filters was the main objective of this thesis. Further, it would be unreasonable to commence hardware experiments without previously testing the filters in simulation, both for reasons of safety and feasibility. However, the drawbacks of testing in simulation reflected upon here do imply the advantage of further evaluation of the filter performances in real hardware experiments. Also, introducing new testing scenarios to evaluate estimation performance in highly dynamic gaits, as well as on uneven terrain would help highlight potential flaws and refinement possibilities.

Chapter 6

Conclusion

This thesis aimed at determining a pose estimation approach for legged robots that combines high accuracy and low implementation complexity. This was done by implementing, testing and evaluating the estimation performance of different filters within a Two Stage Estimation framework and comparing them to a state-of-the-art EKF implementation. Through this work, the UKF for orientation estimation and the TS for position and velocity estimation were identified as very capable filters that best meet these criteria.

As was highlighted throughout this thesis, the UKF offers a simpler alternative for orientation estimation on par with the performance of the EKF, while avoiding any linearization. Simultaneously, the advantages of using a Kalman Filter approach are maintained, with extensions to the state and thus the estimation algorithm being straightforward to implement.

For position and velocity estimation, the TS, ie. the Classical Kalman Filter, was identified as performing similarly accurate as the EKF while being simple to implement, lightweight and very easy to adapt. As proposed by prior work, the foot contact position was included into the state to improve estimation performance.

The work presented here can be directly extended to include comparisons to more recent estimation approaches, such as the InEKF, as well as improved implementations of the TS and UKF. These improvements might include bias estimation and different sigma point sampling methods for the UKF, as well as more extensive parameter tuning for the TS. Lastly, as mentioned in chapter 5, real hardware experiments could be done to further validate the identifications made in this project.

Bibliography

[1] Mohammad O.A. Aqel, Mohammad H. Marhaban, M. Iqbal Saripan, and Napsiah Bt Ismail. Review of visual odometry: types, approaches, challenges, and applications. *SpringerPlus*, 5(1), 2016.

[2] ETHZ ASL. Kalibr: Imu noise model. *https://github.com/ethz-asl/kalibr/wiki/IMU-Noise-Model*, 2016.

[3] Timothy D. Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017.

[4] Axel Barrau and Silvere Bonnabel. The invariant extended Kalman filter as a stable observer. *IEEE Transactions on Automatic Control*, 62(4):1797–1812, 2017.

[5] Axel Barrau and Silvère Bonnabel. Invariant Kalman Filtering. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):237–257, 2018.

[6] Gerardo Bledt, Matthew J. Powell, Benjamin Katz, Jared Di Carlo, Patrick M. Wensing, and Sangbae Kim. MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot. *IEEE International Conference on Intelligent Robots and Systems*, (December):2245–2252, 2018.

[7] Gerardo Bledt, Patrick M. Wensing, Sam Ingersoll, and Sangbae Kim. Contact Model Fusion for Event-Based Locomotion in Unstructured Terrains. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 4399–4406, 2018.

[8] Michael Bloesch. State Estimation for Legged Robots - Kinematics, Inertial Sensing, and Computer Vision. 44(23), 2017.

[9] Michael Bloesch, Christian Gehring, Peter Fankhauser, Marco Hutter, Mark A. Hoepflinger, and Roland Siegwart. State estimation for legged robots on unstable and slippery terrain. *IEEE International Conference on Intelligent Robots and Systems*, pages 6058–6064, 2013.

[10] Michael Bloesch, Marco Hutter, Mark A. Hoepflinger, Stefan Leutenegger, Christian Gehring, C. David Remy, and Roland Siegwart. State estimation for legged robots: Consistent fusion of leg kinematics and IMU. *Robotics: Science and Systems*, 8:17–24, 2013.

[11] Silvere Bonnabel. Left-invariant extended Kalman filter and attitude estimation. *Proceedings of the IEEE Conference on Decision and Control*, pages 1027–1032, 2007.

[12] Marco Camurri, Maurice Fallon, Stephane Bazeille, Andreea Radulescu, Victor Barasuol, Darwin G. Caldwell, and Claudio Semini. Probabilistic Contact Estimation and Impact Detection for State Estimation of Quadruped Robots. *IEEE Robotics and Automation Letters*, 2(2):1023–1030, 2017.

[13] T. Chai and R. R. Draxler. Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, 7(3):1247–1250, 2014.

[14] John L. Crassidis. Sigma-point Kalman filtering for integrated GPS and inertial navigation. *IEEE Transactions on Aerospace and Electronic Systems*, 42(2):750–756, 2006.

[15] Mamadou Doumbia and Xu Cheng. State estimation and localization based on sensor fusion for autonomous robots in indoor environment. *Computers*, 9(4):1–15, 2020.

[16] T. Flayols, A. Del Prete, P. Wensing, A. Mifsud, M. Benallegue, and O. Stasse. Experimental evaluation of simple estimators for humanoid robots. *IEEE-RAS International Conference on Humanoid Robots*, pages 889–895, 2017.

[17] Naman Gupta. State Estimation for Legged Robots Using Proprioceptive Sensors. 2019.

[18] Ross Hartley, Maani Ghaffari, Ryan M. Eustice, and Jessy W. Grizzle. Contact-aided invariant extended Kalman filtering for robot state estimation. *International Journal of Robotics Research*, 39(4):402–430, 2020.

[19] Marco Hutter, Christian Gehring, Michael Bloesch, Mark A. Hoepflinger, C. David Remy, and Roland Siegwart. Starleth: A compliant quadrupedal robot for fast, efficient, and versatile locomotion. *Adaptive Mobile Robotics - Proceedings of the 15th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, CLAWAR 2012*, pages 483–490, 2012.

[20] Marco Hutter, Christian Gehring, Dominic Jud, Andreas Lauber, C. Dario Bellicoso, Vassilios Tsounis, Jemin Hwangbo, Karen Bodie, Peter Fankhauser, Michael Bloesch, Remo Diethelm, Samuel Bachmann, Amir Melzer, and Mark Hoepflinger. ANYmal - A highly mobile and dynamic quadrupedal robot. *IEEE International Conference on Intelligent Robots and Systems*, 2016-November:38–44, 2016.

[21] Jemin Hwangbo, Carmine Dario Bellicoso, Péter Fankhauser, and Marco Hutter. Probabilistic foot contact estimation by fusing information from dynamics and differential/forward kinematics. *IEEE International Conference on Intelligent Robots and Systems*, 2016-November:3872–3878, 2016.

[22] InvenSense Incorporated. MPU-9150 Specification Product 4.3 Revision. 1(408):1–50, 2013.

[23] Simon J Julier and Jeffrey K Uhlmann. A New Extension of the Kalman Filter to Nonlinear Systems. Technical report.

[24] Simon J. Julier and Jeffrey K. Uhlmann. Unscented filtering and nonlinear estimation. In *Proceedings of the IEEE*, volume 92, pages 401–422, mar 2004.

[25] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering, Transactions of the ASME*, 82(1):35–45, 1960.

[26] Kenji Kaneko, Fumio Kanehiro, Shuuji Kajita, Hirohisa Hirukaka, Toshikazu Kawasaki, Masaru Hirata, Kazuhiko Akachi, and Takakatsu Isozumi. Humanoid robot HRP-2. *Proceedings - IEEE International Conference on Robotics and Automation*, 2004(2):1083–1090, 2004.

[27] Joonyoung Kim, Taewoong Kang, Dongwoon Song, and Seung Joon Yi. Design and control of a open-source, low cost, 3d printed dynamic quadruped robot. *Applied Sciences (Switzerland)*, 11(9), 2021.

[28] Youngjoo Kim and Hyochoong Bang. Introduction to Kalman Filter and Its Applications. In *Introduction and Implementations of the Kalman Filter*. IntechOpen, may 2019.

[29] Manon Kok, Jeroen D. Hol, and Thomas B. Schön. Using inertial sensors for position and orientation estimation. *Foundations and Trends in Signal Processing*, 11(1-2):1–153, 2017.

[30] Edgar Kraft. A quaternion-based unscented Kalman filter for orientation tracking. *Proceedings of the 6th International Conference on Information Fusion, FUSION 2003*, 1:47–54, 2003.

[31] Sebastian O.H. Madgwick, Andrew J.L. Harrison, and Ravi Vaidyanathan. Estimation of IMU and MARG orientation using a gradient descent algorithm. *IEEE International Conference on Rehabilitation Robotics*, 2011.

[32] Peter S. Maybeck. *Stochastic models, estimation and control*. Academic Press, 1979.

[33] Corey Montella. The Kalman Filter and Related Algorithms A Literature Review. *Research Gate*, (May):1–17, 2011.

[34] Parag Narkhede, Shashi Poddar, Rahee Walambe, George Ghinea, and Ketan Kotecha. Cascaded complementary filter architecture for sensor fusion in attitude estimation. *Sensors*, 21(6):1–18, 2021.

[35] Simon P. Neill and M. Reza Hashemi. *Ocean Modelling for Resource Characterization*. 2018.

[36] Joern Rehder, Janosch Nikolic, Thomas Schneider, Timo Hinzmann, and Roland Siegwart. Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes. *Proceedings - IEEE International Conference on Robotics and Automation*, 2016-June:4304–4311, 2016.

[37] Maria Isabel Ribeiro. Kalman and Extended Kalman Filters : Concept , Derivation and Properties. *Institute for Systems and Robotics Lisboa Portugal*, (February):42, 2004.

[38] Nuno Ferrete Ribeiro and Cristina P. Santos. Inertial measurement units: A brief state of the art on gait analysis. *ENBENG 2017 - 5th Portuguese Meeting on Bioengineering, Proceedings*, (January), 2017.

[39] L. Tamas, Gh. Lazea, R. Robotin, C. Marcu, S. Herle, and Z. Szekely. State estimation based on kalman filtering techniques in navigation. In *2008 IEEE International Conference on Automation, Quality and Testing, Robotics*, volume 2, pages 147–152, 2008.

[40] Nikolas Trawny and Stergios I. Roumeliotis. Indirect Kalman filter for 3D Attitude Estimation: A Tutorial for Quaternion Algebra. *Technical Report Number 2005-002, Rev. 57*, (2005-002):1–25, 2005.

[41] Eric A Wan and Rudolph Van Der Merwe. The unscented Kalman filter for nonlinear estimation. In *IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium, AS-SPCC 2000*, pages 153–158, 2000.

[42] Greg Welch and Gary Bishop. An Introduction to the Kalman Filter. *In Practice*, 7(1):1–16, 2006.