Bachelor Thesis

Position Estimation of BLDC Motors Using Gaussian Processes

Fabian Sordon

11. June 2019

Abstract

An important task in the area of automation technology is to record the angular position of brushless DC motors. Commonly, inbuilt Hall sensors, magnetic resolvers, and optical encoders are used for that purpose. In preliminary tests, we could identify a definite relationship between the magnetic field distribution of the rotor permanent magnets and a reference angular position. In this thesis, we develop a technique to estimate the position and the speed by measuring the magnetic field of the rotor.

For the pursued estimation approach, we combine Gaussian processes for regression with the extended Kalman filter and the unscented Kalman filter. Besides, we assemble an experimental setup to train the estimation approach with realistic operating conditions of the motor. Due to an observed dependency of the magnetic field measurement, with respect to the angular speed, we identify a GP predictive measurement function with multivariate inputs. The estimation approach is evaluated with real data from the experimental setup and the accuracy of the position and speed estimation is verified.

Eidesstattliche Erklärung

Hiermit erkläre ich, die vorliegende Bachelor Thesis selbstständig angefertigt zu haben. Die verwendeten Quellen sind im Text gekennzeichnet und im Literaturverzeichnis aufgeführt.

Karlsruhe, 11. June 2019

_____

Fabian Sordon

Contents

List of Figures

Notation

Conventions

x	Scalar
$\boldsymbol{x}$	Random variable
$\hat{x}$	Mean of random variable $\boldsymbol{x}$.
$\underline{x}$	Column vector
$\underline{\boldsymbol{x}}$	Random vector
$\hat{\underline{x}}$	Mean of random vector $\underline{\boldsymbol{x}}$.
$\mathbf{A}$	Matrix
$(.)_k$	Quantity at time step k.
$\sim$	Distribution operator.
	E.g., $\boldsymbol{x} \sim \mathcal{U}$ means $\boldsymbol{x}$ is distributed according to $\mathcal{U}$.

State Estimation

$\hat{\underline{x}}_k^-$	*A priori* estimation at time step k
$\hat{\underline{x}}_k^+$	*A posteriori* estimation at time step k
$\mathbf{P}_k^-$	*A priori* estimation-error covariance at time step k
$\mathbf{P}_k^+$	*A posteriori* estimation-error covariance at time step k
$\mathbf{F}_k$	System matrix at time step k
$\mathbf{H}_k$	Measurement matrix at time step k for linear systems,
	Jacobi-matrix of the measurement function for non-linear systems
$\underline{h}_k(\underline{x}_k)$	Measurement function at time step k
$\underline{w}_k$	System noise
$\underline{v}_k$	Measurement noise
$\mathbf{Q}_k$	System noise covariance
$\mathbf{R}_k$	Measurement noise covariance

Gaussian Processes

$\hat{f}_*$	GP predictive mean function for scalar outputs
σ_{f_*}	GP predictive covariance function for scalar outputs
$\mathbf{K}(\mathbf{X}, \mathbf{X}')$	GP kernel matrix
$k(\underline{x}, \underline{x}')$	GP kernel function
σ_n	GP hyperparameter: Noise standard deviation
σ_f	GP hyperparameter: Signal standard deviation
l	GP hyperparameter: Covariance characteristic length-scale

Abbreviations

KF	Kalman Filter
EKF	Extended Kalman Filter
UKF	Unscented Kalman Filter
GP	Gaussian Process
BLDC	Brushless Direct Current
I^{2}C	Inter-Integrated Circuit
SPI	Serial Peripheral Interface

CHAPTER 1

Introduction

In automation technology and robotics, the recording of covered distances, angular positions and speed must be permanently conducted [1]. The measurement of the angular position of electrical motors is the most important component for motor-feedback-systems. In order to accurately control the angular position and speed of an electric motor, knowledge of the instantaneous position is required. Currently, sensors such as inbuilt Hall sensors, magnetic resolvers, and optical encoders are used to sense the angular position.

In this thesis, an approach is developed to estimate the position and the speed of a brushless DC electric motor by measuring the magnetic field of the rotor.

1.1 Motivation

Typically, brushless DC electric motors (BLDC motors) are designed with two-pole-pair permanent magnets on the rotor. These permanent magnets produce a magnetic field, that can be detected at the back side of the motor. In preliminary tests, we could identify a definite relationship between measurements of this magnetic field and the reference angular position. This relationship exhibits a good signal-to-noise ratio. From this knowledge, we want to develop a technique to estimate the position and the speed of a motor using the measurement output of magnetic sensors.

In the pursued estimation approach, we intend to consider both the measurement output and the knowledge from previous estimates. For this purpose, we use a Kalman filter, in which the previous estimate is incorporated via a system equation. For the measurement output, a measurement equation is utilized.

To find the underlying nonlinear measurement function mapping the state of the motor to the magnetic field, we introduce Gaussian processes as regression technique. Furthermore, we apply the common B-splines approximation technique as a reference. Gaussian processes (GP) feature non-parametric regression and uncertainty estimates for their predictions [2]. Besides, GP could be successfully applied

to Bayesian filters in [3].

Generally, the Kalman filter is defined for linear systems. To operate with such nonlinear measurement functions, we introduce the extended Kalman filter (EKF) and the unscented Kalman filter (UKF).

In the next paragraph, we give a brief overview of a typical BLDC motor. In chapter 2, we introduce the state-of-the-art for angular position sensors. In chapter 3, the theoretical background for state estimation and the regression techniques is presented. We introduce the pursued position estimation approach in chapter 4. In chapter 5, we develop an experimental setup for the magnetic measurements. We evaluate the results in chapter 6 and conclude in chapter 7.

1.2 Overview of a BLDC Motor

An overview of a BLDC motor is given in 1.1. It consists of a stator fixed to the housing and a rotor, that features one rotational degree of freedom. Typically, four permanent magnets are mounted on the surface of the rotor. Moreover, six coils are placed on the within the stator.

The identification of the angular position of BLDC motors is required due to several reasons: Firstly, in contrast to brushed DC motors, BLDC motors are not self-commutating and, therefore, the motor control uses knowledge of the rotor position to commutate the motor

Figure 1.1: Overview of a Typical BLDC Motor

correctly. Secondly, position knowledge can be used to identify ground truth information, when the motor causes changes in the state of the technical system. For example, in vehicle applications information about the rotor position can be used to estimate the actual position of the vehicle on its path.

A common way to sense the position is to use rotation sensors such as inbuilt Hall sensors, magnetic resolvers, or optical encoders. However, instead of using sensors, it is also possible to run a motor with sensorless motor control making use of the back electromotive force (back-EMF) [5].

CHAPTER 2

State-of-the-Art: Overview of Angular Position Sensors

This chapter provides an overview of common angular position sensors. A widespread technique is the usage of inbuilt Hall sensors to detect the magnetic field of the rotor permanent magnets. The relatively new magnetic resolvers sense the angular position by measuring the magnetic field of an external permanent magnet. The most accurate sensors are optical encoders.

2.1 Inbuilt Hall Sensors

Hall sensors are magnetic sensors, that are able to detect the magnetic field based on the Hall effect. The usage of inbuilt Hall sensors is the most typical approach to sense the rotor position in order to commutate the stator coils. Therefore, in most BLDC motors, three Hall sensors are already embedded into the stationary part of the motor [5].

The Hall effect theory is summarized in [5]: "If an electric current carrying conductor is kept in a magnetic field, the magnetic field exerts a transverse force on the moving charge carriers which tends to push them to one side of the conductor. This is most evident in a thin flat conductor. A buildup of charge at the sides of the conductors will balance this magnetic influence, producing a measurable voltage between the two sides of the conductor."

In BLDC motors, one of these Hall sensors will produce a LOW-signal if a north pole passes its surrounding and a HIGH-signal for a south pole. Combining these three Hall signals, the electrical angle of the motor can be derived. However, for two pole pairs of permanent magnets on the rotor, this will give us 12 positions per rotation, meaning only every 60 degrees a new position will be detected. For middle and high-speed commutation this is still a satisfying accuracy. Nonetheless, if the motor

needs to be commutated at very low speeds or we want to use the rotor position to model the state of a technical system, this might not satisfy the requirements.

2.2 Magnetic Resolver

Analogous to inbuilt Hall sensors, the magnetic resolver is in most applications based on Hall effect sensors. But instead of measuring the magnetic field of the permanent magnets on the rotor, it senses the magnetic field of an external two pole permanent magnet mounted on one shaft end of the rotor. The position of the sensor is accurately aligned to the position of the magnet. Four quadratic patterned Hall sensors measure the magnetic field in an area of the magnet, where its induced magnetic field distribution is almost linear. This can be exemplified with the rotary magnetic position sensors from *AMS* [6].

In [7] is deduced, how the angle α can be identified from these Hall sensor signals. Generally, the Hall sensors produce four sinusoidal signals for one rotation,

$$
\begin{aligned}
U_{sin}(s) &= \overline{U}sin(\omega s) + UO_{sin}, \\
U_{-sin}(s) &= -\overline{U}sin(\omega s) + UO_{sin}, \\
U_{cos}(s) &= \overline{U}cos(\omega s) + UO_{cos}, \\
U_{-cos}(s) &= -\overline{U}cos(\omega s) + UO_{cos},
\end{aligned}
\tag{2.1}
$$

that contain the unknown offsets UO_{sin} and UO_{cos}. To overcome the offset UO_{sin} and UO_{cos} the difference signals

$$
\begin{aligned}
U_{si}(s) &= U_{sin}(s) - U_{-sin}(s), \\
U_{co}(s) &= U_{cos}(s) - U_{-cos}(s)
\end{aligned}
\tag{2.2}
$$

are derived. From these two values, the current angle α can be derived by applying the trigonometric arctan- and arccot-functions. On an evaluation board, this can be performed using the CORDIC-algorithm [8].

2.3 Optical Encoder

Optical encoders are used in applications where most demanding resolution and accuracy of position sensing is required [7]. A schematic view on a typical optical encoder is given in 2.1. An emitter sends out rays on a bright-dark-pattern of a code disc, which is projected on a photo-electrical receiver. This emitter-receiver system can be placed several times on different radial positions with respect to the rotation axis. For two shifted, but apart from that identical bright-dark-patterns on the code disc, an incremental digital output signal will be generated, also called

A/B-Interface 2.2. Each edge of the signal A or B will change the counter for the position. Due to the 90 degree shifting of the A and B signal, we can identify the direction of the rotation from the sequence of the HIGH or LOW edges. The counter divided by the total amount of increments per rotation provides the angular position.
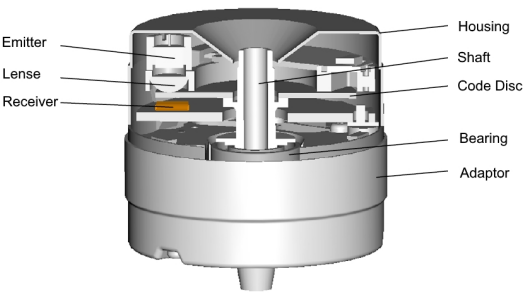
Figure 2.1: Embedded Optical Encoder [7]
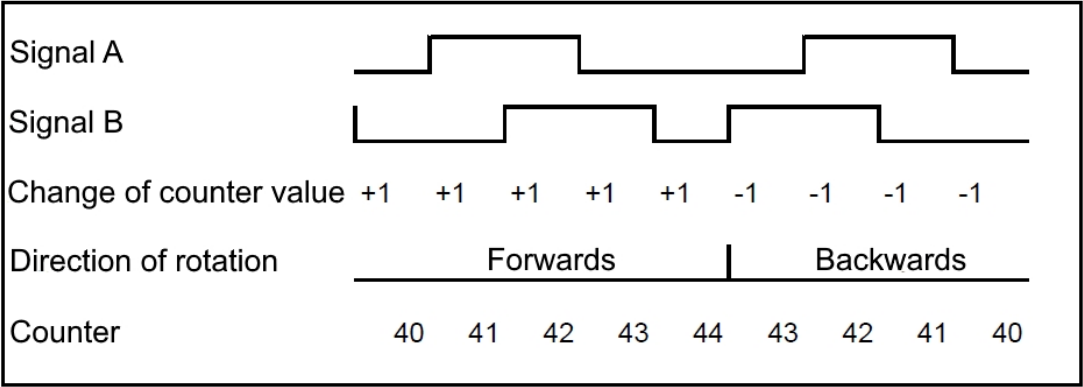
Figure 2.2: Digital Interface of an Optical Encoder [7]

CHAPTER 3

Theoretical Background

After the state of art for position estimation was described in 2, in this chapter we introduce the theoretical background required to develop a position estimation model. This model processes magnetic field measurements of the rotor to estimate the angular position and speed. For this purpose, we will introduce the Kalman filter and its extensions to nonlinear problems for state estimation. Likewise, an introduction to Gaussian processes and B-splines for regression of the measured data is given.

A good overview of state estimation namely extended and unscented Kalman filter is given in [4]. Along, [2] provides a very comprehensive background on Gaussian processes for regression. [9] and [10] introduce interpolation and B-splines approximation. An integration of Gaussian processes to Bayes filters such as the extended Kalman filter and the unscented Kalman filter is performed in [3].

3.1 State Estimation

In [4], an intuitive definition for states is given: The states of a system are variables that represent the complete status of a general system at one instant of time. Furthermore, we use state estimation to derive these unknown sought-after states either to implement a state-feedback controller or if we are interested in the states in general for further research.

Accordingly, we can apply state estimation to any problem, where we are interested in several properties of a system, but we can not measure these properties directly. For example, we want to derive the angular position and speed of a motor, but only have measurements regarding the magnetic field of the rotor. With a state estimator, we define these unknown system properties as the system state variables. The noisy measurements are defined as measurement output.

Henceforward, we introduce linear time-discrete systems and the recursive least

squares estimation, which is the basis for most state estimation approaches. Next, we give an introduction to the Kalman filter (KF). Then, we illustrate the extended Kalman filter (EKF) and the unscented Kalman filter (UKF).

3.1.1 Linear Time-Discrete Systems

In most applications, sensors can only be evaluated at discrete time points $t = t_k$ for every time interval ΔT. Therefore, we consider the discrete time series given by $t_k = t_{k-1} + \Delta T$ with the initial time t_0. In this chapter, we will only examine the time-discrete form of system state estimation.

A definition of a linear time-discrete system from [4] is given by the following equations,

$$
\begin{aligned}
\underline{x}_k &= \mathbf{F}_{k-1}\underline{x}_{k-1} + \mathbf{G}_{k-1}\underline{u}_{k-1} + \underline{w}_{k-1}, \\
\underline{y}_k &= \mathbf{H}_k\underline{x}_k + \underline{v}_k.
\end{aligned}
\tag{3.1}
$$

We call the first equation of (3.1) system equation and the second one measurement equation. The vector $\underline{x}_k$ is the system state whereas $\underline{y}_k$ is the measurement output at the discrete time point t_k. As mentioned before, the system state contains all the state variables representing the status of the system. The measurement output contains variables, that actually correspond to the properties of the measurement data processed in our system.

The vector $\underline{u}_{k-1}$ is the control vector for the previous time point t_k. The contained variables provide information about outside influences on the system that can be determined.

The matrices $\mathbf{F}_{k-1}$, $\mathbf{G}_{k-1}$ and $\mathbf{H}_k$ are called system matrix, input matrix and output matrix, respectively.

Along, $\underline{w}_{k-1}$ is the system noise and $\underline{v}_k$ is the measurement noise. These quantities of uncertainty are what makes our system a stochastic system. They correspond to outside influences that can not be determined. Besides, they provide information about how much confidence we have in certain quantities of our system model. [4]

In most applications, both noise terms $\underline{w}_{k-1}$ and $\underline{v}_k$ are seen as zero-mean uncorrelated Gaussian noise. We define

$$
\begin{aligned}
\underline{w}_{k-1} &\sim \mathcal{N}(\underline{0}, \mathbf{Q}_{k-1}), \\
\underline{v}_k &\sim \mathcal{N}(\underline{0}, \mathbf{R}_k)
\end{aligned}
\tag{3.2}
$$

as Gaussian distributed noise terms with the system noise covariance $\mathbf{Q}_{k-1}$ measurement noise covariance $\mathbf{R}_k$.

3.1.2 Recursive Least Squares Estimation

We identified a linear time-discrete system model in (3.1) but our target is to find an optimal state estimation of the ground truth system status based on noisy measurements. For that, we introduce the recursive least squares estimation technique, which is comprehensively described in [4].

From a given noisy measurement $\underline{y}_k$, we want to find the optimal estimate $\hat{\underline{x}}_k$ of the system. There are several approaches to identify the optimal estimate over a whole time series of measurements. However, in most applications, an estimation of the optimal state needs to be computed directly after a new measurement is received. In order to restrict the computation cost for each estimation step we need to find a recursive definition of optimal states dependant on the previous estimate $\hat{\underline{x}}_{k-1}$ and the new measurement $\underline{y}_k$.

For a recursive estimation we first need to assign the initial properties of the state for $k = 0$ in a probabilistic way. We define

$$
\begin{aligned}
\hat{\underline{x}}_0 &= \mathbb{E}(\underline{x}), \\
\mathbf{P}_0 &= \mathbb{E}[(\underline{x} - \hat{\underline{x}}_0)(\underline{x} - \hat{\underline{x}}_0)^T]
\end{aligned}
\tag{3.3}
$$

in which $\mathbb{E}$ determines expected values for a probability situation and $\mathbf{P}_0$ is called the estimation-error covariance for $k = 0$.

Now we can recursively define the estimator for any time index $k = 1, 2, \ldots$ as follows,

$$
\begin{aligned}
\underline{y}_k &= \mathbf{H}_k \underline{x}_k + \underline{v}_k, \\
\hat{\underline{x}}_k &= \hat{\underline{x}}_{k-1} + \mathbf{K}_k(\underline{y}_k - \mathbf{H}_k \hat{\underline{x}}_{k-1}).
\end{aligned}
\tag{3.4}
$$

The unknown matrix $\mathbf{K}_k$ is called the estimator gain matrix and $(\underline{y}_k - \mathbf{H}_k \hat{\underline{x}}_{k-1})$ is called the correction term.

The estimator gain matrix can be derived from the previous estimation-error covariance $\mathbf{P}_{k-1}$, the output matrix $\mathbf{H}_k$ and the measurement noise covariance matrix $\mathbf{R}_k$ with the equation

$$
\mathbf{K}_k = \mathbf{P}_{k-1}\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_{k-1}\mathbf{H}_k^T + \mathbf{R}_k)^{-1}
\tag{3.5}
$$

and the current estimation-error covariance $\mathbf{P}_k$ can be computed with

$$
\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k-1}(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T.
\tag{3.6}
$$

3.1.3 The Kalman Filter

The Kalman filter is a widespread state estimation approach for linear systems. It is characterized by a stochastic system model and in most applications is defined

over discrete time properties.

For the most part, the Kalman filter is based on the recursive least squares estimation. The state of a system is estimated using noisy measurements. Besides, the uncertainty of the estimated state is derived.

The main contribution achieved by the Kalman filter is to take into account, that we can actually predict the state $\underline{x}_k$ from $\underline{x}_{k-1}$ even without knowing the measurement $\underline{y}_k$. Accordingly, we can perform an *a priori* state estimation at a time step, at which the measurement is not known yet. Then, an *a posteriori* state estimation is executed taking into account the measurement.

We denote the *a priori* estimation with $\underline{x}_k^-$ and the *a posteriori* estimation with $\underline{x}_k^+$. The same is applied to the estimation error covariance denoting $\mathbf{P}_k^-$ and $\mathbf{P}_k^+$.

In [11], these two estimation steps are defined as follows: The identification of $\underline{x}_k^-$ and $\mathbf{P}_k^-$ is called *prediction step*. Afterwards, in the *filter step*, the current measurement $\underline{y}_k$ is compared with the predicted result. For the weighting between prediction and measurement, a so-called *Kalman gain* $\mathbf{K}_k$ is identified.

Let us review a linear time-discrete system,

$$
\begin{aligned}
\underline{x}_k &= \mathbf{F}_{k-1}\underline{x}_{k-1} + \mathbf{G}_{k-1}\underline{u}_{k-1} + \underline{w}_{k-1}, \\
\underline{y}_k &= \mathbf{H}_k\underline{x}_k + \underline{v}_k.
\end{aligned}
\tag{3.7}
$$

The first equation is called system equation and the second one is called measurement equation. Using the system equation, $\hat{\underline{x}}_k^-$ can be predicted from previous estimates. In the measurement equation, a relationship is mapped between the system state vector $\underline{x}_k$ and the measurement output $\underline{y}_k$.

The equations of the estimator for the Kalman filter are deduced in [4] and given by

$$
\begin{aligned}
\hat{\underline{x}}_k^- &= \mathbf{F}_{k-1}\hat{\underline{x}}_{k-1}^+ + \mathbf{G}_{k-1}\underline{u}_{k-1}, \\
\mathbf{P}_k^- &= \mathbf{F}_{k-1}\mathbf{P}_{k-1}^+\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}, \\
\mathbf{K}_k &= \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1}, \\
\hat{\underline{x}}_k^+ &= \hat{\underline{x}}_k^- + \mathbf{K}_k(\underline{y}_k - \mathbf{H}_k\hat{\underline{x}}_k^-), \\
\mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-.
\end{aligned}
\tag{3.8}
$$

In (3.8), the equations for one estimation iteration are given. The first two equations take place during the prediction step. The last tree equations occur during the filter step.

3.1.4 The Extended Kalman Filter

The above definition of a Kalman filter is restricted to linear system models. How can we apply a Kalman filter if we want to model the system equation and the

output equation of the estimator with nonlinear functions?

For this purpose we introduce the extended Kalman filter (EKF) that approaches nonlinear dependencies in a system by performing a linearization of a nonlinear function using Taylor series expansion of the first order. The EKF is the most widely used state estimator for nonlinear systems [4]. There are also approaches using higher order Taylor series expansion, but here only the first order linearization will be contemplated. A comprehensive derivation of the extended Kalman filter is given in [4].

We map a typical nonlinear system with the equations

$$\begin{aligned} \underline{x}_k &= \underline{f}_{k-1}(\underline{x}_{k-1}, \underline{u}_{k-1}, \underline{w}_{k-1}), \\ \underline{y}_k &= \underline{h}_k(\underline{x}_k, \underline{v}_k). \end{aligned} \tag{3.9}$$

By undertaking a linearization of the nonlinear functions $\underline{f}_{k-1}$ and $\underline{h}_k$ we can identify partial derivative matrices, which can be processed in the estimator. The biggest concern is that an approximation of a function by linearization only indicates good results around a linearization point.

The best approximation for $\underline{f}_{k-1}$ can be found by using the state estimation from the previous time instance $\hat{\underline{x}}_{k-1}^+$ as a linearization point. For $\underline{h}_k$ we use the predicted state $\hat{\underline{x}}_k^-$ as a linearization point. For a given previous state estimate $\hat{\underline{x}}_{k-1}^+$ and estimation-error covariance $\mathbf{P}_{k-1}^+$ we can perform recursively an estimation.

Firstly, we can derive the matrices

$$\mathbf{F}_{k-1} = \left. \frac{\partial \underline{f}_{k-1}}{\partial \underline{x}} \right|_{\hat{\underline{x}}_{k-1}^+}, \quad \mathbf{L}_{k-1} = \left. \frac{\partial \underline{f}_{k-1}}{\partial \underline{w}} \right|_{\hat{\underline{x}}_{k-1}^+} \tag{3.10}$$

with a given $\hat{\underline{x}}_{k-1}^+$. Then, with these matrices, we can, secondly, identify the predicted state $\hat{\underline{x}}_{k-1}^+$ and the predicted estimation-error covariance $\mathbf{P}_k^-$ in a time update,

$$\begin{aligned} \mathbf{P}_k^- &= \mathbf{F}_{k-1} \mathbf{P}_{k-1}^+ \mathbf{F}_{k-1}^T + \mathbf{L}_{k-1} \mathbf{Q}_{k-1} \mathbf{L}_{k-1}^T, \\ \hat{\underline{x}}_k^- &= \underline{f}_{k-1}(\hat{\underline{x}}_{k-1}^+, \underline{u}_{k-1}, \underline{w}_{k-1}). \end{aligned} \tag{3.11}$$

Thirdly, we can use the computed predicted state $\hat{\underline{x}}_k^-$ to calculate the derivatives

$$\mathbf{H}_k = \left. \frac{\partial \underline{h}_k}{\partial \underline{x}} \right|_{\hat{\underline{x}}_k^-}, \quad \mathbf{M}_k = \left. \frac{\partial \underline{h}_k}{\partial \underline{v}} \right|_{\hat{\underline{x}}_k^-}. \tag{3.12}$$

Finally, we have all needed terms to undertake a measurement update for the estimator. We compute

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{M}_k \mathbf{R}_k \mathbf{M}_k^T)^{-1}, \\ \hat{\underline{x}}_k^+ &= \hat{\underline{x}}_k^- + \mathbf{K}_k(\underline{y}_k - \underline{h}_k(\hat{\underline{x}}_k^-, 0)), \\ \mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-. \end{aligned} \tag{3.13}$$

and we can use $\mathbf{P}_k^+$ and $\hat{\underline{x}}_k^+$ to recursively execute the next estimation step.

3.1.5 The Unscented Kalman Filter

Even though the EKF is the most widespread estimation approach, its estimation quality depends on the accuracy of the linear approximation of the nonlinear function. If the function values in the direct environment of the linearization point vary largely, the EKF might give poor estimation results.

Therefore, we introduce the unscented Kalman filter (UKF) as a second approach for estimation of nonlinear systems. Though the linearization error can be increased, the UKF typically requires a higher computation time than the EKF [3]. Likewise, we do not claim completeness of the theory in this work and refer to [4] for comprehensive instructions on UKF.

The basic concept of the UKF is to transform the probability distribution of a given state to a corresponding probability distribution when it is processed in the nonlinear system model. This is performed by using the unscented transformation. In [4] is described that the unscented transformation is based on two fundamental principles:

(1) A nonlinear transformation of a single point can easily be made.

(2) The true probability density function of a state vector can be approximated by a set of single points.

For this purpose $2n$ sample points referred as "sigma points" $\underline{x}^{(i)}$ are symmetrically located around the mean $\overline{\underline{x}}$ of a given state $\underline{x}$. The state $\underline{x}$ is a $n \times 1$ vector which means for every state variable we compute two more sigma points. The distance between sigma points and mean is selected in such a way, that their distribution yields the given estimation-error covariance $\mathbf{P}$. Accordingly, we define

$$
\begin{aligned}
\underline{x}^{(i)} &= \overline{\underline{x}} + \widetilde{\underline{x}}^{(i)} \quad \text{with} \quad i = 1, ..., 2n, \\
\widetilde{\underline{x}}^{(i)} &= \left(\sqrt{n\mathbf{P}}\right)_i^T \quad \text{with} \quad i = 1, ..., 2, \\
\widetilde{\underline{x}}^{(n+i)} &= -\left(\sqrt{n\mathbf{P}}\right)_i^T \quad \text{with} \quad i = 1, ..., 2,
\end{aligned}
\tag{3.14}
$$

That features the advantage that only for this set of points the mapping of the nonlinear function, namely the state transition function and the measurement function, needs to be computed.

The distribution of the sigma points, that are transformed in such a way, provides information about the new state estimate and estimation-error covariance. However, at this point we will not deepen the mean and covariance approximation with the aid of the transformed sigma points.

3.2 Regression Techniques

In the previous chapter, we introduced the Kalman filter and its extensions to non-linear estimation systems. Forthwith, we present techniques to identify regression functions for a set of training data. With these nonlinear functions, we aim at mapping training input data to noisy measurement output data.

In [2] two basic approaches to deal with such a supervised learning problem are introduced:

(1) We restrict the considered class of functions, for example, we assume only linear or polynomial functions of the input.

(2) We assign a prior probability to any possible function. For example, a smoother function could be considered to be more likely than an uneven function with respect to some prior knowledge.

In order to both find the underlying mean function mapping the input data to the most probable output value and also its corresponding uncertainty, we will introduce Gaussian processes for regression. We then shortly give an overview on basis splines (B-splines) as an alternative regression technique.

B-splines can be classified to the first approach whereas Gaussian processes can be assigned to the second approach.

3.2.1 Gaussian Processes for Regression

Gaussian processes (GP) are a regression technique, that identifies a probability distribution over a function. This means that both a prediction mean and its uncertainty is provided over the input space. Besides, they are a non-parametric approach that can approximate function parameters such as smoothness and noise standard deviation from the training set. [2]

Yet, [2] gives a comprehensive theoretical background on GP for both regression and classification problems. In this chapter, we summarize the basis concept of Gaussian process for regression.

At the bottom, let us define the training set $\mathcal{D}$ as

$$\mathcal{D} = \{(\underline{x}_i, y_i)|i = 1, ..., n\} \tag{3.15}$$

with input vector $\underline{x}_i$ of the dimension D and y_i referred as scalar output. We can also write the training set in the form $\mathcal{D} = (\mathbf{X}, \underline{y})$ with the design matrix $\mathbf{X}$ containing all the training inputs.

We model the real process $f(\underline{x})$ as a random variable and define the noisy measurement equation

$$y = f(\underline{x}) + \epsilon \qquad (3.16)$$

assuming additive Gaussian noise ϵ with the variance σ_n^2.

Now we want to predict the mean and covariance of the measurement function for a test input matrix $\mathbf{X}_*$. This leads us to the key prediction equations for GP regression for the predictive mean

$$\hat{\underline{f}}_* = \mathbf{K}(\mathbf{X}_*, \mathbf{X})[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_n^2\mathbf{I}]^{-1}\underline{y} \qquad (3.17)$$

and the predictive covariance

$$\Sigma_{\mathbf{f}_*} = cov(\underline{f}_*) = \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) - \mathbf{K}(\mathbf{X}_*, \mathbf{X})[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_n^2\mathbf{I}]^{-1}\mathbf{K}(\mathbf{X}, \mathbf{X}_*). \qquad (3.18)$$

As well, we define the kernel matrix $\mathbf{K}(\mathbf{X}, \mathbf{X}')$ as the evaluation of the kernel function $k(\underline{x}_i, \underline{x}_j')$ for each combination of $\underline{x}_i$ and $\underline{x}_j'$,

$$\mathbf{K}(\mathbf{X}, \mathbf{X}') = \begin{pmatrix} k(\underline{x}_1, \underline{x}_1') & \cdots & k(\underline{x}_1, \underline{x}_{N'}') \\ \vdots & \ddots & \vdots \\ k(\underline{x}_N, \underline{x}_1') & \cdots & k(\underline{x}_N, \underline{x}_{N'}') \end{pmatrix}. \qquad (3.19)$$

The kernel function $k(\underline{x}_i, \underline{x}_j')$ can be chosen dependent upon the regression problem, however, the most common one is the Gaussian kernel or squared exponential kernel

$$k(\underline{x}, \underline{x}') = \sigma_f e^{-\frac{1}{2l^2}||\underline{x} - \underline{x}'||^2}. \qquad (3.20)$$

The hyperparameters of such a Gaussian process are the covariance characteristic length scale l, the signal standard deviation σ_f^2 and the noise standard deviation σ_n^2. They can be approximated by minimizing the negative logarithmic marginal likelihood.

With the prediction equations (3.17) and (3.18) we can model nonlinear functions in the system and measurement equation of Kalman filters. Nonetheless, for the extended Kalman filter we need to determine the partial derivatives of the nonlinear functions in order to perform a linearization. Accordingly we need to derive the Jacobi-matrix of the functions.

Applied on GP, the partial derivatives of the predictive mean function, with respect to the test input variables, need to be identified.

In [3] the derivatives of the GP predictive mean function for an input vector $\underline{x}_*$ is given by

$$\frac{\partial \overline{\underline{f}}_*(\underline{x}_*)}{\partial \underline{x}_*} = \frac{\partial \mathbf{K}(\underline{x}_*, \mathbf{X})}{\partial \underline{x}_*} [\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}]^{-1} \underline{y}. \tag{3.21}$$

The partial derivatives of the kernel matrix is given by the partial derivatives of the Gaussian kernel function

$$\frac{\partial k(\underline{x}_*, \underline{x})}{\partial \underline{x}_*[i]} = -\frac{1}{2l^2} \sigma_f(\underline{x}_*[i] - \underline{x}[i]) e^{-\frac{1}{2l^2} ||\underline{x}_* - \underline{x}||^2}. \tag{3.22}$$

3.2.2 Splines Approximation

In this thesis, we compare the GP regression with the common splines approximation. Splines are piecewise polynomial functions, that typically map from a univariate or multivariate input to a univariate output. The crossover area between two piecewise polynomial functions can be steadily differentiated [9].

A comprehensive theoretical background of splines and Bezier-techniques can be found in [10]. Hereinafter, we only refer to basis-splines (B-splines). A good overview of B-splines is given in [9].

CHAPTER 4

The Position Estimation Approach

From the theoretical background in the last chapter, we develop an approach to estimate the position and speed of a BLDC motor by measuring the magnetic field of the rotor. In 4.1 is shown a basic block diagram for this approach. A magnetic sensor measures the magnetic flux density at one position behind the motor. This raw measurement is processed in a preprocessing step mainly performing a mean adjustment adding an offset to the magnetic field measurement. This measurement is then fed into the Kalman filter, which estimates the position, the speed, and the estimation-error covariance. This estimation is also considered for the next estimation step, here illustrated with a feedback loop. Yet, to reasonably process the magnetic field measurement in the Kalman filter, a system equation and a measurement equation need to be specified. The system equation can be assumed to be linear but the measurement equation will be modeled with an unknown nonlinear measurement function. We use several regression approaches to identify the nonlinear measurement function. For training the measurement function we need both the magnetic field measurements and a ground truth position and speed for reference. Accordingly, a reference sensor will be used providing a raw position measurement. In another preprocessing step both the ground truth position and the speed of the motor can be identified from this measurement.

Firstly we will explain the preprocessing of the sensor data. Next, the Kalman filter will be specified for the extended Kalman filter and the unscented Kalman filter. Lastly, several approaches will be used to train the measurement function of the Kalman filter. The first approach will assume a measurement function with a scalar input namely the position. The second approach will be a multi-input function with respect to both the position and the speed of the motor. We will use Gaussian processes and B-splines for regression.
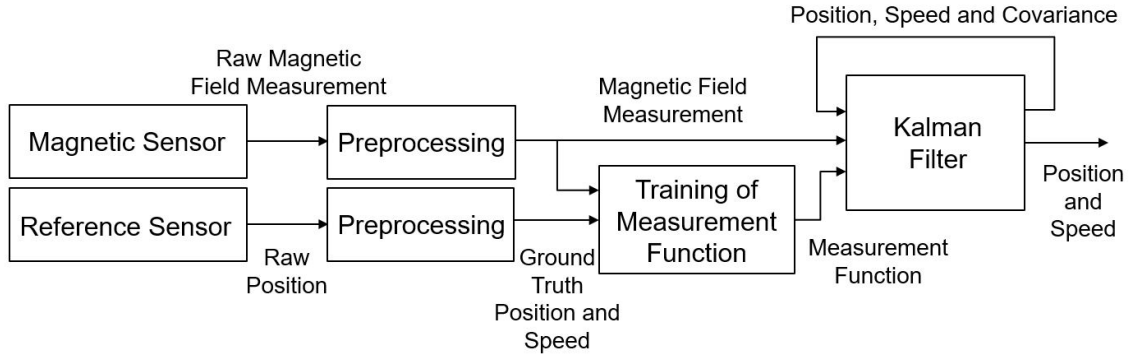
Figure 4.1: Block Diagram of the Position Estimation Approach

4.1 Preprocessing

Preprocessing for the Magnetic Sensor

Here we deal with the subject that it is easier to work with measurement data in further steps if we align the dimensions of the measurements to the desired quantity. We define one magnetic measurement at time $T = T_k$ as $\underline{b}'_k = (b_{k1}, b'_{k2}, ..., b'_{kn})^T$ for n being the amount of measurement outputs. b'_{ki} is one magnetic measurement in the measurement direction $i = 1, ..., n$.

In a later step we will apply several regression approaches on the magnetic measurements. Gaussian processes, as defined in 3, assume a zero-mean output value for the predictive mean function. Thus, we adjust the mean value of the magnetic measurement b'_{ki} with the offset

$$OFF_i = \frac{1}{m} \sum_{j=1}^{m} b'_{ji} \tag{4.1}$$

with m being the length of the offset training set. Besides, we want to scale the measured values into the range of -1 to 1. Consequently, we compute the preprocessed magnetic measurement b_{ki} as

$$b_{ki} = \frac{2(b'_{ki} - OFF_i)}{b'_{i,max} - b'_{i,min}}. \tag{4.2}$$

Preprocessing for the Reference sensor

To preprocess the measured ground truth position θ'_k we get the position in radian dimension θ_k by computing

$$\theta_k = \frac{2\pi \theta'_k}{\theta'_{max} - \theta'_{min}}. \tag{4.3}$$

Withal, if the position measurement is very accurate and performed for infinitely small time intervals, we can also derive the speed. We assume that a measurements is provided at the time instance T_k and the time interval between two succession of measurement sets is $\Delta T = T_k - T_{k-1} = constant$. From this assumption the ground truth rotation speed ω_k can be derived from the ground truth positions θ_k and θ_{k-1} by computing

$$\omega_k = \frac{\theta_k - \theta_{k-1}}{\Delta T}. \tag{4.4}$$

4.2 The Kalman Filter

In this section, we will first determine the system equation and the measurement equation for the position estimation approach. Later, we will specify the extended Kalman filter and the unscented Kalman filter for our approach.

4.2.1 System Model

The most basic knowledge about the examined system is that the target estimation variables are the angular position θ and the angular speed ω and the output measurement is given by variables containing the magnetic flux density. In the previous section, we considered the common case for n measurement directions. However, below we specify to have two measurement variables b_x and b_y. We begin by making assumptions about the considered system:

(1) The system equation can be seen as a linear system.

(2) The control vector $\underline{u}_k$ of the system is equal to zero.

(3) The system noise and the measurement noise are zero-mean additive noise

(4) The covariance of the system and measurement noise are constant matrices.

(5) The measurement function is independent of the measurement noise.

How can we validate these assumptions? For (1) exist several examples, where the system equations for modeling of motors are assumed to be linear. [12]
For (2) a possible control input vector could be for example the current in the stator coils of the motor. For this approach, we did not realize any interface between the motor control and the estimator, which could be a setup for future work. Hence, the control vector is assumed to be equal to zero.
The reason for the assumptions (3), (4) and (5) is mostly to reduce the computing time of one estimation step. However, later it will be illustrated that indeed the

variances of the noise terms are almost constant for all possible system states. We review the linear time-discrete system equation

$$\underline{x}_k = \mathbf{F}_{k-1}\underline{x}_{k-1} + \mathbf{G}_{k-1}\underline{u}_{k-1} + \underline{w}_{k-1} \tag{4.5}$$

equal to the one in (3.7). For our system we assume $\underline{u}_{k-1} = 0$.

The target estimation variables of the Kalman filter are the angular position θ and the angular speed ω. So we define the state $\underline{x}_k$ as

$$\underline{x}_k = \begin{pmatrix} \theta_k \\ \omega_k \end{pmatrix} \tag{4.6}$$

for the time T_k. How can we predict a state $\underline{x}_k^-$ only knowing the previous state estimate $\underline{x}_{k-1}^+$?

Assuming constant angular speed, we can predict $\omega_k^- = \omega_{k-1}^+$. Furthermore, with the constant time interval ΔT between two measurements, we can predict the angular position $\theta_k^- = \theta_{k-1}^+ + \Delta T \omega_{k-1}^+$. This will give us the linear system equation

$$\underline{x}_k = \begin{pmatrix} \theta_k \\ \omega_k \end{pmatrix} = \mathbf{F}\underline{x}_{k-1} + \underline{w}_{k-1} = \begin{pmatrix} 1 & \Delta T \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \theta_{k-1} \\ \omega_{k-1} \end{pmatrix} + \underline{w}_{k-1}. \tag{4.7}$$

For small ΔT and a good previous estimate the simplification $\omega_k^- = \omega_{k-1}^+$ is valid because indeed the variation of ω_k will be quite small. A similar example for the system equation of a motor is given in [12].

However, the system equation will be blind towards changes in ω_k over a longer series of time and, therefore, the state estimation needs to consider the actual measurement output as well. For the measurement equation a linear assumption is not possible. We define the measurement output vector $\underline{y}_k$ as

$$\underline{y}_k = \begin{pmatrix} b_{kx} \\ b_{ky} \end{pmatrix} \tag{4.8}$$

containing the magnetic field measurement variables b_{kx} and b_{ky}. Assuming the non-linear measurement function h to be independent of the time and the measurement noise $\underline{v}_k$, we define the measurement equation

$$\underline{y}_k = \begin{pmatrix} b_{kx} \\ b_{ky} \end{pmatrix} = \underline{h}(\underline{x}_k) + \underline{v}_k. \tag{4.9}$$

The measurement function $h(\underline{x}_k)$ can be determined from a regression approach using testing data.

Now we want to specify the noise terms $\underline{w}_k$ and $\underline{v}_k$. We already assumed zero mean Gaussian distribution and constant covariance matrices $\mathbf{Q}$ and $\mathbf{R}$, resulting in

$$\underline{w}_k \sim \mathcal{N}(\underline{0}, \mathbf{Q}), \quad \underline{v}_k \sim \mathcal{N}(\underline{0}, \mathbf{R}). \tag{4.10}$$

The measurement noise covariance $\mathbf{R}$ can be identified from the Gaussian process regression approach, which provides the noise standard deviation hyperparameter σ_n for each measurement direction. Consequently we define

$$\mathbf{R} = \begin{pmatrix} \sigma_{n,x}^2 & 0 \\ 0 & \sigma_{n,y}^2 \end{pmatrix}. \tag{4.11}$$

Accordingly, $\mathbf{R}$ is a constant diagonal matrix. This can be assumed, if, indeed, the measurement noise is approximately constant for each measurement variable. We will verify this assumption in 6.

For the system noise covariance $\mathbf{Q}$ we will test the resolution of the Kalman filter for a range of noise values and select the optimal system noise covariance. In this thesis we use *The Nonlinear Estimation Toolbox* by Steinbring to setup the structure for the Kalman filter [13].

4.2.2 The Exteded Kalman Filter

In the theoretical background, a most general case for the extended Kalman filter was considered. The system function $\underline{f}_k$ maps from the previous state estimation, the previous control input and the previous system noise to the new predicted state estimation. In an analogous manner, the measurement function $\underline{h}_k$ maps from the state estimation and the measurement noise to the measurement output.

Due to the assumptions made in the previous section we identified the system equation (4.7) and the measurement equation (4.9). Accordingly, a linearization must only be performed for the nonlinear measurement function. Due to the measurement function assumed to be independent with respect to the measurement noise, the equations for the extended Kalman filter can be simplified. An algorithm containing all required equations to perform one estimation step is given in 4.1.

Algorithm 4.1 Algorithm of the Extended Kalman Filter

---

input : $\underline{x}_{k-1}^{+}$: Previous state estimation

input : $\mathbf{P}_{k-1}^{+}$: Previous estimation-error covariance

input : $\underline{y}_k$: Current measurement output

Output: $\underline{x}_k^{+}$: New state estimation

Output: $\mathbf{P}_k^{+}$: New estimation-error covariance

1 $\mathbf{P}_k^{-} = \mathbf{F}\mathbf{P}_{k-1}^{+}\mathbf{F} + \mathbf{Q}$

2 $\underline{x}_k^{-} = \mathbf{F}\underline{x}_{k-1}^{+}$

3 $\mathbf{H}_k = \left.\frac{\partial h_k}{\partial \underline{x}}\right|_{\hat{\underline{x}}_k^{-}}$

4 $\mathbf{K}_k = \mathbf{P}_k^{-}\mathbf{H}_k^{T}(\mathbf{H}_k\mathbf{P}_k^{-}\mathbf{H}_k^{T} + \mathbf{R})^{-1}$

5 $\underline{x}_k^{+} = \underline{x}_k^{-} + \mathbf{K}_k[\underline{y}_k - \underline{h}(\underline{x}_k^{-})]$

6 $\mathbf{P}_k^{+} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^{-}$

The algorithm recieves the previous state estimate $\underline{x}_{k-1}^{+}$, the previous estimation error covariance $\mathbf{P}_{k-1}^{+}$ and the current measurement output $\underline{y}_k$ as an input. Firstly, the predicted estimation-error covariance $\mathbf{P}_k^{-}$ is computed. Next, the predicted state $\underline{x}_k^{-}$ is derived from $\underline{x}_{k-1}^{+}$. We then perform a linearization of the measurement function using $\underline{x}_k^{-}$ as a linearization point. This will give us the Jacobi-matrix $\mathbf{H}_k$. Now the Kalman gain $\mathbf{K}_k$ can be identified. Lastly, the new state estimation $\underline{x}_k^{+}$ and the new estimation-error covariance $\mathbf{P}_k^{+}$ can be computed representing the algorithm outputs.

4.2.3 The Unscented Kalman Filter

In [4], the advantage of the UKF was deduced over the EKF for highly non-linear models. However, in most applications the UKF also requires a higher comuting time per estimation step [3]. In this thesis, we also want to perform tests using the UKF. This makes it easier in the end to validate each of the estimation models.
We use the same assumptions from the EKF, namely constant time-independent noise covariances. In addition, we do not need to compute the Jacobimatrix of the measurement function with respect to the state variables. Instead, we identify the probability distribution of the output values by transforming the probability distribution of the given state. This is done by selecting the sigma points of the UKF.
A general algorithm of the UKF can be found in [4]. For our purpose we just need to define the system model as linear and the noise matrices as constants.

4.3 Regression Approach for the Measurement Function

We identified the specifications for a Kalman filter that can estimate the position and speed of a motor using magnetic field measurements. However, the nonlinear measurement function mapping the state of the system to the magnetic measurement is still unknown. Consequently, we use two regression approaches to identify this measurement function. For that, we need a training set $\mathbf{D}$ containing both magnetic measurements and the dedicated ground-truth state over a time series. The state is defined here as both the angular position and the speed of the rotor.

We pursue two different approaches for estimation: The first approach expects the measurements of the magnetic field to be only dependent on the position of the rotor. Thence, we map the angular position to the magnetic field measurement. The realization of such a measurement function is performed in the following using Gaussian processes and B-splines.
In the second approach the measurement function maps both the position and the speed to the measurement output. Consequently, we need to find a regression function over multidimensional input space. This is addressed in the last subsection.

4.3.1 Gaussian Processes

Measurement Function

We consider a scalar input regression function mapping the ground truth position θ to the magnetic field (b_x, b_y). We define the training set $\mathcal{D}$ for regression of the measurement function as

$$\mathcal{D} = (\underline{\theta}, \mathbf{Y}) \tag{4.12}$$

with measurement input $\underline{\theta}$ and measurement output $\mathbf{Y} = (\underline{b}_x, \underline{b}_y)$.
We assume that the measurement noise covariance is a constant matrix. Later on, in 6, this assumption is validated experimentally. Accordingly, we will only consider the GP predictive mean function in the measurement equation. Recapitulating the measurement equation of the Kalman filter (4.9), we get

$$\underline{y}_k = \begin{pmatrix} b_{kx} \\ b_{ky} \end{pmatrix} = \underline{h}(\theta_k) + \underline{v}_k = \begin{pmatrix} \hat{f}_{*,x}(\theta_k) \\ \hat{f}_{*,y}(\theta_k) \end{pmatrix} + \underline{v}_k. \tag{4.13}$$

with the GP predictive mean $\overline{f}_{*,x}$ for the magnetic field in x-direction and $\overline{f}_{*,y}$ for the magnetic field in y-direction, respectively. We apply the predictive mean function for GP regression from (3.17) to define

$$\begin{aligned} \hat{f}_{*,x}(\theta_*) &= \mathbf{K}(\theta_*, \underline{\theta})[\mathbf{K}(\underline{\theta}, \underline{\theta}) + \sigma_{n,x}^2 \mathbf{I}]^{-1} \underline{b}_x, \\ \hat{f}_{*,y}(\theta_*) &= \mathbf{K}(\theta_*, \underline{\theta})[\mathbf{K}(\underline{\theta}, \underline{\theta}) + \sigma_{n,y}^2 \mathbf{I}]^{-1} \underline{b}_y \end{aligned} \tag{4.14}$$

in two output dimensions for the scalar training input θ_*. Consequently, two different GP predictive mean functions entirely define the measurement function $\underline{h}(\theta)$. The hyperparameters (l, σ_f, σ_n) in both output dimensions can be approximated by minimizing the negative logarithmic marginal likelihood. For that we use the *GPML Library* from Rasmussen [14].

Requirement of the EKF and the UKF

Now the nonlinear measurement function is given for GP regression, but we still need to determine the Jacobi-matrix $\mathbf{H}_k$ for the linearization with the EKF. From (3.12) we get

$$\mathbf{H}_k = \frac{\partial \underline{h}(\theta_k)}{\partial \underline{x}} = \begin{pmatrix} \frac{\partial \hat{f}_{*,x}(\theta_k)}{\partial \theta} & 0 \\ \frac{\partial \hat{f}_{*,y}(\theta_k)}{\partial \theta} & 0 \end{pmatrix}. \tag{4.15}$$

Applying equation (3.21), the derivatives of the GP predictive mean functions can be evaluated via

$$\frac{\partial \hat{f}_*(\theta_*)}{\partial \theta_*} = \frac{\partial \mathbf{K}(\theta_*, \underline{\theta})}{\partial \theta_*} [\mathbf{K}(\underline{\theta}, \underline{\theta}) + \sigma_n^2 \mathbf{I}]^{-1} \underline{b}_x,$$

$$\frac{\partial \hat{f}_*(\theta_*)}{\partial \theta_*} = \frac{\partial \mathbf{K}(\theta_*, \underline{\theta})}{\partial \theta_*} [\mathbf{K}(\underline{\theta}, \underline{\theta}) + \sigma_n^2 \mathbf{I}]^{-1} \underline{b}_y. \tag{4.16}$$

Using the UKF, the only requirement is to evaluate the measurement function additionally for two sigma points.

4.3.2 B-Splines

In this thesis, we use B-splines mostly to compare the used Gaussian processes with a widespread reference approximation tool. For B-splines, we use the *Curve Fitting Toolbox* [15]. Using this Matlab library, it is both possible to derive a smoothing splines function in the B-form from a noisy data set and also to identify its derivative. For training of the B-spline we use the identical training set $\mathcal{D}$ provided to the GP. However, a more accurate way to determine B-splines functions for approximation can be possible by selecting a training set different to the GP training set. Mostly, the utilization of B-splines enables a better validation of the performed GP in the end of this thesis.

4.3.3 Measurement Function with Multivariate Input using Gaussian Processes

So far, the previous techniques identify a measurement function mapping only the angular position to the magnetic field. Hereinafter, we also want to derive a measurement function for the case that we indeed observe a dependency of the magnetic

field measurement with respect to the speed of the rotor.

To give an idea why we follow up both approaches, we forestall some general observations from the evaluation in 6: In the beginning of the experimental process, testing on the magnetic field could not be executed at higher speeds than 200 rpm. This occurred due to a magnetic sensor providing measurements only at a low sampling rate. Therefore, a dependency of the measurement output on higher speeds could not be investigated. We expected the magnetic field to be independent from the rotation speed. Using a new sensr with a higher sampling rate, tests at higher rotation speeds could be executed. We then observed a nonlinear relationship between the measurement output and the rotation speed and aligned the estimation approach accordingly. A detailed analysis for different rotation speeds will be presented in chapter 6.

We define the training set for regression of the measurement function as $\mathbf{D} = (\mathbf{X}, \mathbf{Y})$ with measurement input $\mathbf{X} = (\underline{\theta}, \underline{\omega})$ and measurement output $\mathbf{Y} = (\underline{b}_x, \underline{b}_y)$.

So far the measurement function with multivariate is only realized using Gaussian processes for regression. However, a realization using B-splines is also possible in future work.

The measurement function with multivariate inputs can be derived analogously to the univariate Gaussian processes. The only difference is that the measurement function maps both the position and the speed to the magnetic field. Similarly to (4.13) we describe the measurement equation as

$$\underline{y}_k = \begin{pmatrix} b_{kx} \\ b_{ky} \end{pmatrix} = \underline{h}(\underline{x}_k) + \underline{v}_k = \begin{pmatrix} \hat{f}_{*,x}(\underline{x}_k) \\ \hat{f}_{*,y}(\underline{x}_k) \end{pmatrix} + \underline{v}_k. \tag{4.17}$$

We define the GP predictive mean functions

$$\begin{aligned} \hat{f}_{*,x}(\underline{x}_*) &= \mathbf{K}(\underline{x}_*, \mathbf{X})[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_{n,x}^2 \mathbf{I}]^{-1} \underline{b}_x, \\ \hat{f}_{*,y}(\underline{x}_*) &= \mathbf{K}(\underline{x}_*, \mathbf{X})[\mathbf{K}(\mathbf{X}, \mathbf{X}) + \sigma_{n,y}^2 \mathbf{I}]^{-1} \underline{b}_y \end{aligned} \tag{4.18}$$

for the multivariate test input $\underline{x}_*$. Lastly, we define the Jacobi-matrix $\mathbf{H}_k$ as

$$\mathbf{H}_k = \frac{\partial \underline{h}(\underline{x}_k)}{\partial \underline{x}} = \begin{pmatrix} \frac{\partial \hat{f}_{*,x}(\underline{x}_k)}{\partial \theta} & \frac{\partial \hat{f}_{*,x}(\underline{x}_k)}{\partial \omega} \\ \frac{\partial \hat{f}_{*,y}(\underline{x}_k}{\partial \theta} & \frac{\partial \hat{f}_{*,y}(\underline{x}_k}{\partial \omega} \end{pmatrix}. \tag{4.19}$$

CHAPTER 5

Experimental Setup

In this chapter, we want to present the experimental setup for the position estimation approach. A block diagram for this experimental setup is given in 5.1.
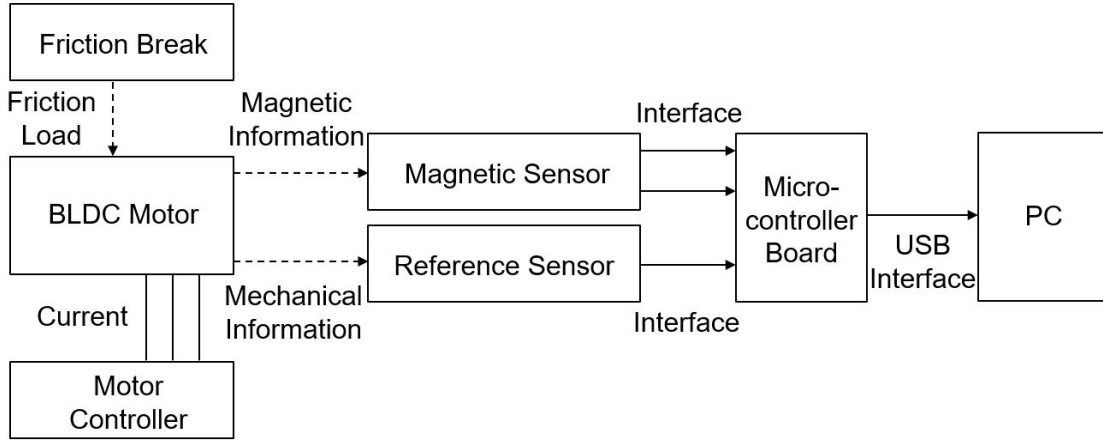

Figure 5.1: Block Diagram of the Hardware Setup

Primarily, a magnetic sensor is placed on the back side of the BLDC motor to measure the magnetic field of the rotor. A reference sensor is then used to measure the ground truth position. These sensors are then connected to a microcontroller board, that can transmit the measurement data to a PC via a digital USB interface. With such an experimental setup, we can both identify a training set for the measurement function and also receive data to test and evaluate the state estimation executed in the Kalman filter.

To ensure that such a training set is valid for realistic operating conditions of the motor in a use case, we realize several operation conditions of the motor while we measure the data. Consequently, we run the motor at a wide range of angular speeds by commutation with a motor controller. Besides, we design a friction break system to apply an approximately constant load torque.

In the first section, we will discuss the hardware selection concerning mostly the mag-

netic sensor and the optical encoder. In the second section, we will explain, how the operational conditions for the BLDC motor are realized with a motor controller and a friction brake system. We will then set up the recording of the measurement data points using a microcontroller board. Finally, by using the measurements, we will perform a magnetic field analysis to find the optimal position for the magnetic sensor.

5.1 Hardware Selection

Analogously to 5.2 a schematic sketch of the hardware setup is specified in 5.2. The magnetic sensor measures the magnetic field of the rotor (here simplified visualized as red magnetic field head curve) in two spatial axis $\underline{b}_x$ and $\underline{b}_y$ and sends the measurements via two analog signals to an Arduino Due microcontroller board. On the output shaft of the motor, an optical encoder is mounted detecting the ground truth position for reference. The optical encoder outputs ground truth in ABZ interface and the Arduino Due is programmed to read the information in the given format. To commutate the BLDC motor, a motor controller from STM is used. For the brake system, we selected conventional bicycle break shoes. Two of those brake shoes press with a constant, but adjustable normal force against an aluminum disc, which is linked to the output shaft.
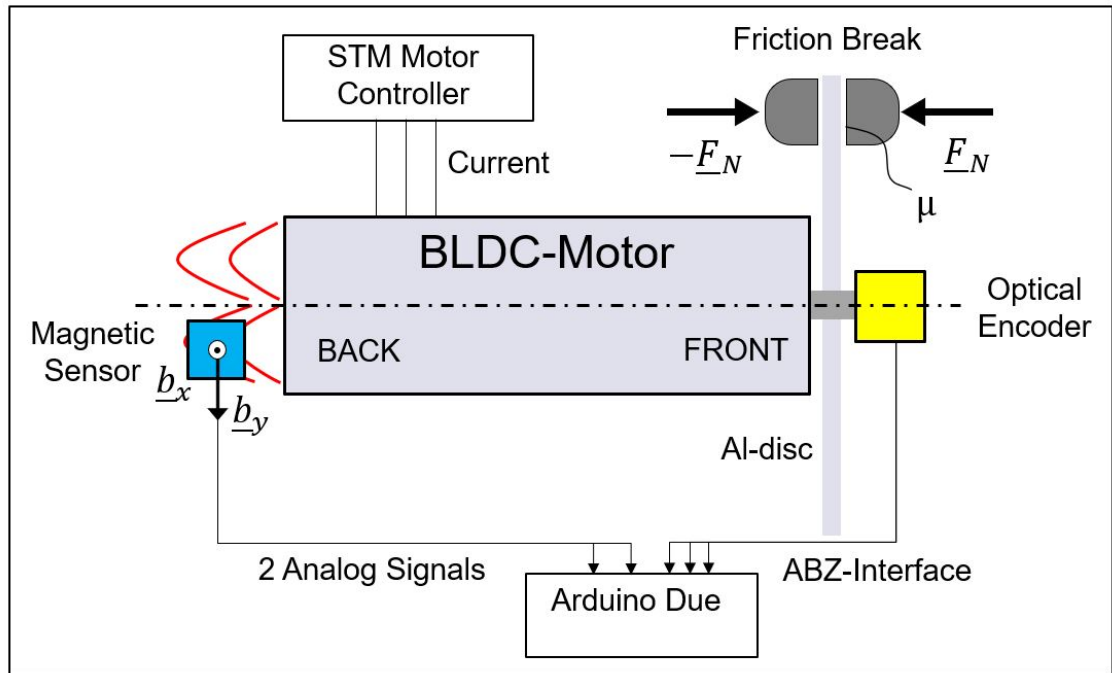

Figure 5.2: Schematic Sketch of the Hardware Setup

5.1.1 Arduino Due

The Arduino Due is a high-end MCU board, which is based on the ARM Cortex-M3 CPU running with a 84 MHz clock [16]. For our application, the Arduino Due is suitable due to the wide range of interfaces supported by the pins. For example, the Arduino Due contains a 12-bit analog-to-digital converter (ADC) that runs at 0.167 MHz. Additionally, digital interfaces such as a serial peripheral interface (SPI) or a inter-integrated circuit (I^2C) are supported. During operation, the Arduino Due can also communicate with a PC via USB serial interface.

5.1.2 Optical Encoder as Ground Truth

Let us shortly summarize the requirement for the reference sensor. On the one hand, we need the sensor to derive a measurement function for the Kalman filter and, on the other hand, to evaluate in the end the estimation results of the approach. Hence, we need a very high accuracy due to the measurement is assumed to be the ground truth position of the motor. Furthermore, the accuracy must be even that high, that we can still derive a reliable speed from the position.

As a reference sensor we selected the TMCS-2 optical incremental encoder shown in 5.3. This is a relatively low-cost and small size model but achieving an enormous resolution of 10000 pulses per revolution [17].

Figure 5.3: Optical Encoder [17]

Reviewing the knowledge from the state-of-the-art about incremental encoder, we know that the output signal is mainly given by two 90 shifted pulse signals A and B. Evaluating the signals for every edge, we even get a resolution of 40000 steps per revolution which corresponds to an accuracy of 0.009 deg. In addition to the A and B output, this encoder contains a Z output, that will only give a HIGH-signal for one step per rotation to give a reference if the step counter contains an error. This can be also seen as a zero-position.
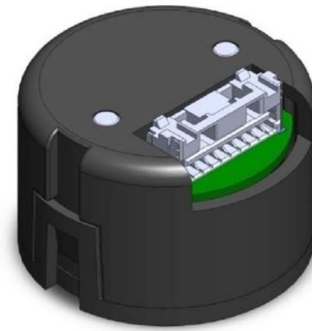
5.1.3 Magnetic Sensor

The most important section of the experimental setup is the selection of the magnetic sensor. On this, the success of the measurement approach mostly depends on.
We define the requirement for the magnetic sensor as follows:

(1) The magnetic sensor should measure in several directions in space. We want to take the magnetic measurement only from one position. Due to a circular movement of the rotor, we expect periodical measurements over one rotation. To identify the direction of the motion we at least need to have two shifted measurement outputs.

(2) The magnetic sensor should detect a wide intensity range of the magnetic flux density. In most cases of designing a technical system, the installation space is limited. Accordingly, the sensor should be positioned close to the motor housing, where the intensity of the magnetic flux density is expected to be relatively high.

(3) The magnetic sensor should measure with high accuracy. The success of the position estimation depends on how accurate the magnetic sensor can detect the magnetic field of the rotor.

(4) The magnetic sensor should provide the measurements at a high sampling rate. We want to apply the position estimation approach on both high-speed and low-speed operational conditions. If the sensor outputs data at a low sampling rate, the estimator will easily lose track of the actual position of the motor.

We started the experimental setup by using a three-axis magnetic compass with a digital interface. However, during the procedure, we realized that this sensor could only be used for low-speed applications due to its low sampling rate. Hence, we changed the experimental setup by using a two-axis magnetic sensor with an analog interface. Both the type of magnetic sensors are briefly covered below.

Three-Axis Magnetic Compass with Digital Interface

As a first magnetic sensor, we selected the ASD2613-R TinyShield 3-axis compass. It features an adequate field range of +/-8 gauss and a 2 mgauss field resolution for each output axis. A big advantage of this sensor is that it is available as an evaluation board with an integrated 12-bit ADC and the measurements can be comfortably read via I^{2}C-interface. [18]

This directly leads to the disadvantage of the sensor due to its maximum sampling rate of 160 Hz. That means one magnetic measurement can only be detected every 6.25 ms. For low-speed applications,
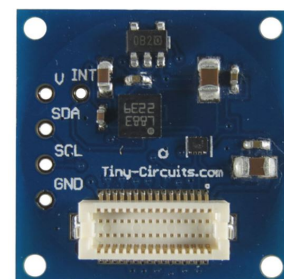
Figure 5.4: 3-Axis Compass [18]

this is not critical. For example, for rotation speeds of $200\,\mathrm{rpm}$, 48 measurements can be taken per rotation, which is sufficient to conduct an estimation. For higher speed application this sampling rate is by far too small.

Nonetheless, we could perform successful tests with this sensor for low-speed operation up to $200\,\mathrm{rpm}$ and also implemented an estimator that can process a three-dimensional measurement output. This works completely analogously to the presented estimator by just adding a third dimension to the measurement function and vector.

Two-Axis Magnetic Sensor with Analog Interface

For rotational speeds higher than $200\,\mathrm{rpm}$, we need a sensor with a higher sampling rate. We first investigated on other digital evaluation boards similar to the previous one but with a higher sampling rate. Hence, most of such kind of products are designed to detect the earth magnetic field and are not laid-out for high-speed evaluation. Besides, we investigated if there are evaluation boards available with an analog output. Such kind of available sensors are sold at a very high price.

Finally, we decided to use the Honeywell 2-axis magnetoresistive sensor HMC1052L-TR [19]. Even though it only measures the magnetic field

Figure 5.5: 2-Axis Analog Magnetic Sensor on Adapter

in two spatial directions and its field range is limited to $+/-6\,\mathrm{gauss}$, it features two key advantages over the former sensor:

(1) A magnetic field resolution of $0.28\,\mathrm{mgauss}$

(2) An analog output signal with a bandwidth of $3\,\mathrm{MHz}$

Consequently, this sensor can provide us a high-resolution measurement via an analog signal which we only need to evaluate with an ADC sampling at a high frequency. However, this sensor has the disadvantage that it does not come with an evaluation board. The sensor has the dimensions $3 \times 3 \times 1$ millimeters and needs to be soldered on an adapter board. The sensor on such an adapter board is shown in 5.5.

Before the measurements can be read, amplification must be performed according to the application notes of the sensor [19]. For both measurement directions, we receive an OUT+ and OUT- differential signal that will be supplied to the OP-Amplifier. In the application notes, a resistance of $500k\Omega$ is used in the feedback loop. Instead,
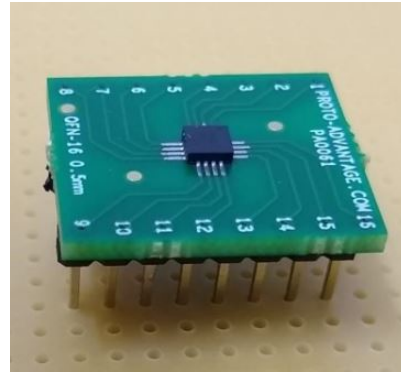
we use a potentiometer covering the range $200 - 1200k\Omega$. With that we are able to adjust the gain of the output signal. Lastly, we need to setup a set/reset-circuit to demagnetize the sensor at the initial time.

A block diagram showing the setup of the magnetoresistive sensor is given in 5.6.
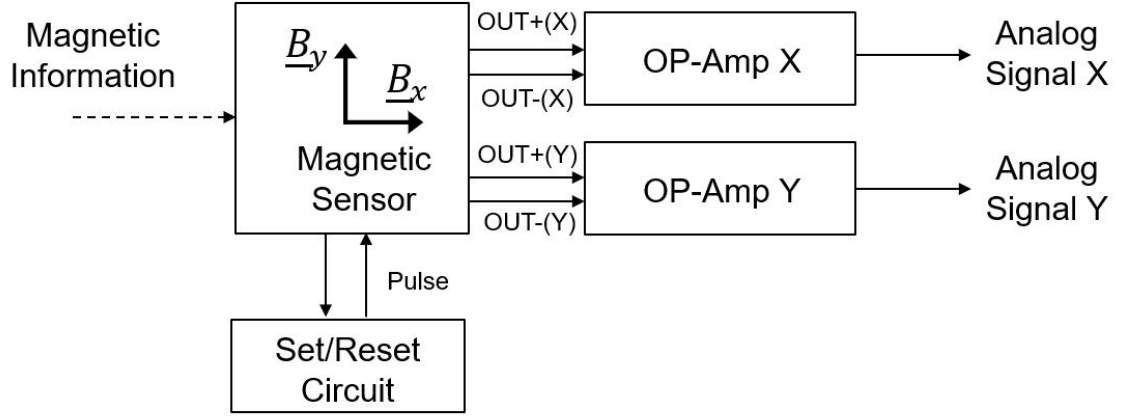

Figure 5.6: Block Diagram of Setup of Magnetoresistive Sensor

5.1.4 BLDC Motor

The probed Motor is a 24 V three-phase BLDC motor with a 10:1 gearbox. The BLDC motor is rated for the maximum torque 2.68 Nm. It works analogously to the example discussed in the introduction of this thesis. We have two-pole-pair surface mounted permanent magnets on the rotor and six stator coils. Also, the motor contains inbuilt Hall sensors being able to detect the rotor position with an accuracy of 60 deg.

For commutating the motor the three cables U, V, and W are available. To read the Hall sensors we have the output cables HU, HV, and HW, which require a 5 V supply voltage.

Earlier, we mentioned that we want to mount the reference sensor on the output shaft of the motor. If we have a 10:1 gearbox the resolution of the optical encoder per rotation of the rotor will decrease. Thus, we dismounted the gearbox and now only consider the rotation speed of the rotor.

5.1.5 Motor Controller

To commutate the motor we employ the STEVAL-SPIN3201 motor controller from STM. This will provide the motor a 24 V supply voltage and is able to control the stator coils using the field-oriented control.

We only use the motor controller to run the motor at different rotation speeds to take magnetic measurements at more realistic operational conditions. In this thesis, we do not aim to integrate the estimation approach into a motor controller. Accordingly, as position feedback, the inbuilt Hall sensors of the BLDC motor are connected to the motor controller.

The STM motor controller features an USB interface and the set speed can comfortably be piloted from a PC using the ST Motor Control Workbench software.

The tests are performed adjusting the rotation speed of the motor with such an interface between $-2000\,\mathrm{rpm}$ and $2000\,\mathrm{rpm}$ and at the same time taking measurements with the magnetic sensor and the reference sensor.

5.1.6 Friction Break System

In a real operational scenario, the motor will not always operate at no load torque. We do not know yet if this will influence the measurements of the magnetic sensor. In order to validate the position estimation approach for load torque operation, we realize a friction brake system being able to replicate such operational conditions. A schematic sketch of the friction brake system is given in 5.7.
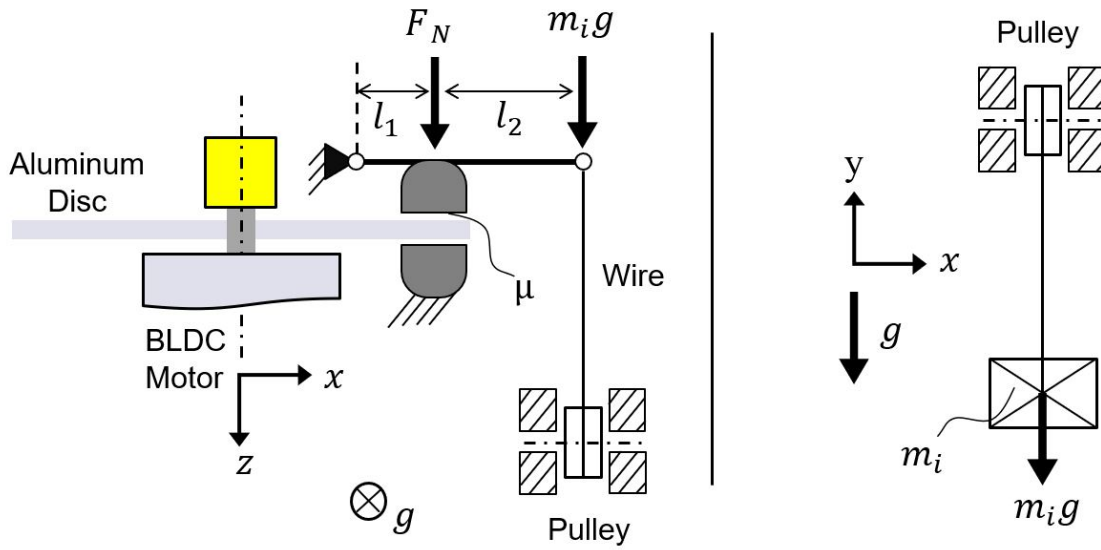

Figure 5.7: Schematic Sketch of the Friction Break System

An aluminum disc is linked to the output shaft of the motor. Laterally one brake shoe with the friction coefficient μ presses the aluminum disc with a normal force F_N against a stationary second brake shoe. This normal force F_N is exerted by a lever arm, which is pulled by a wire towards a pulley. The rope is then connected to a weight m_i which is accelerated due to gravity g.

With such a setup we can apply approximately constant load torques on the motor

and adjust this torque by selecting different weights m_i. We can derive the load torque M_L with the equation

$$M_L = r * \mu * g * m_i * \frac{l_1 + l_2}{l_1},$$ (5.1)

where r is the distance between rotation axis and point of action of the break shoe. Considering the 10:1 gear box ratio we will get the effective load torque

$$M_{L,eff} = 10 M_L.$$ (5.2)

For our setup we have the following constants:

(1) Lever arm $l_1 = 0.022\,\mathrm{m}$

(2) Lever arm $l_2 = 0.078\,\mathrm{m}$

(3) Radius $r = 0.045\,\mathrm{m}$

(4) Slide friction coefficient for aluminum and vulcanite $\mu = 0.6$ [20]

(5) Acceleration due to gravity $g = 9.81\,\mathrm{m\,s^{-2}}$

We chose the weights to be $m_1 = 0\,\mathrm{kg}$, $m_2 = 0.03\,\mathrm{kg}$ and $m_3 = 0.06\,\mathrm{kg}$. With that, we can realize operational conditions with load torques listed in 5.1.

m_i/kg	F_N/N	M_L/Nm	$M_{L,eff}$/Nm
0	0	0	0
0.03	1.34	0.036	0.36
0.06	2.68	0.072	0.72

Table 5.1: Load Torque Operational Conditions

The BLDC motor is rated for $2.68\,\mathrm{Nm}$ from the manufacturer and here we only test until $0.72\,\mathrm{Nm}$. This is mostly performed for reasons of safety due to the fast rotating aluminum disc. However, the initial torque is higher because for low speeds a static friction coefficient is applied instead of a sliding friction coefficient. Generally, the static friction is higher than the sliding friction.

5.2 Recording Data Points

We previously selected the Arduino Due to readout the measurement data and transfers it to the PC via USB interface. In the following, we summarize how the data points are recorded with such a hardware setup.

The optical encoder provides the position via an incremental interface with the output pins A, B, and Z. The ARM Cortex M3, on which the Arduino Due is based on, contains a quadrature decoder *QDEC*, which can either derive the angular position or the angular speed from incremental interfaces at high speeds [21]. It is not possible to get both position and speed from the quadrature decoder.

The position is the more valuable information for our application. Therefore, we identify the angular speed indirectly from the angular position. However, this reduces the accuracy of the speed measurement.

To receive measurements from the digital three-axis compass, we use an I^{2}C interface. It was already mentioned that the sampling rate is limited to 160 Hz. We found a way to speed up the sampling rate to 200 Hz.

The analog two-axis magnetic sensor is interfaced via two analog signals. The Arduino contains a 12-bit analog-to-digital converter. It takes 16 μs to read two analog inputs and the position from the optical encoder. Therefore, the sampling rate is 62.5 kHz. However, the limiting factor for such a high sampling rate is now the Serial USB interface between the Arduino and the PC. It has been observed that such sensor data can be only transferred at a maximum serial rate of 455 Hz using a serial baud rate of 115200.

In this work, we did not yet find a faster way to provide the sensor data to the PC with an Arduino. Nonetheless, regarding this high sampling rate, much better results could be achieved if the data is directly evaluated with the estimation approach on the microcontroller. For running the motor at 2000 rpm we can record 6 data points per revolution using the digital sensor and 13.6 data points per revolution using the analog sensor. The results are listed in the table 5.2.

Magnetic Sensor	Sampling Rate /kHz	Serial Rate (USB) /kHz	Points at 2000RPM /rev
Digital 3-Axis	0.2	0.2	6
Analog 2-Axis	62.5	0.455	13.6

Table 5.2: Sampling Rate of the Data Recording

On the PC these data points are evaluated in a Matlab environment. The implementation of the position estimation approach is as well realized in Matlab. The tests are executed on an Intel i5 CPU running at 2.2 GHz.

5.3 Magnetic Field Analysis for Sensor Positioning

Now we are able to record data points from the selected experimental setup and we can derive the optimal positions of the magnetic sensors. We define the targets for optimal positioning:

(1) We achieve a minimal measurement noise.

(2) We utilize the magnetic field range supported by the magnetic sensors while leaving some buffer if we eventually detect a higher magnetic field than expected.

(3) The measurement outputs should be shifted in such a way that we can derive the direction of rotation uniquely.

In 5.8 we see a simulation of the magnetic flux density in a cross-section of a BLDC motor during operation. We observe that within the area of the rotor the magnetic field curves are shaped very steadily due to the permanent magnets. In the stator area, the magnetic field is induced mostly by the current in the stator coils. Here, we observe a highly nonuniform run in the magnetic field curves.

Therefore, we take measurements of the magnetic field produced by the rotor permanent magnets. The most direct access to this field is at the back side of the motor, because at the front side commonly a gearbox is assembled. The measurements are taken at different po-

Figure 5.8: FEM Simulation of the Sectional Magnetic Flux Density in a BLDC Motor [22]

sitions in radial direction $\underline{r}$ and axial directions $\underline{z}$ towards the back side of the motor housing. A schematic diagram showing the sensor positions is given in 5.9.

The digital sensor measures the magnetic field in the three spatial directions $\underline{e}_x$, $\underline{e}_y$, and $\underline{e}_z$. The analog sensor measures the magnetic field in the directions $\underline{b}_A$ and $\underline{b}_B$, where as $\underline{b}_A = \underline{e}_y$ and $\underline{b}_B = -\underline{e}_y$. Radial and axial distances $\underline{r}_1$ and $\underline{z}_1$ characterize the position of the digital sensor and the distances $\underline{r}_2$ and $\underline{z}_2$ characterize the position of the analog sensor. The angle α between $\underline{r}_1$ and $\underline{e}_x$ or $\underline{r}_2$ and $\underline{e}_y$ is approximately 45 deg. Accordingly, we expect a similar shape of the measured curve in $\underline{e}_x$ and $\underline{e}_y$ direction which is shifted about approximately 90 deg.
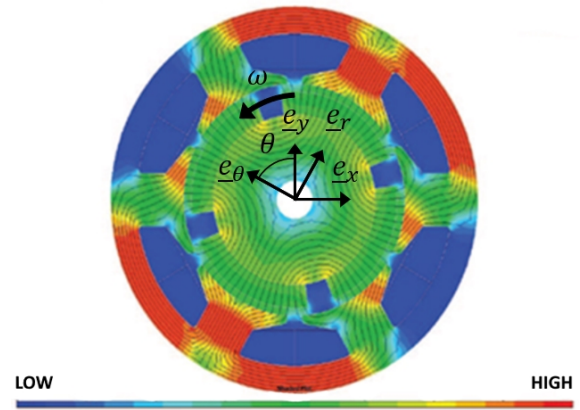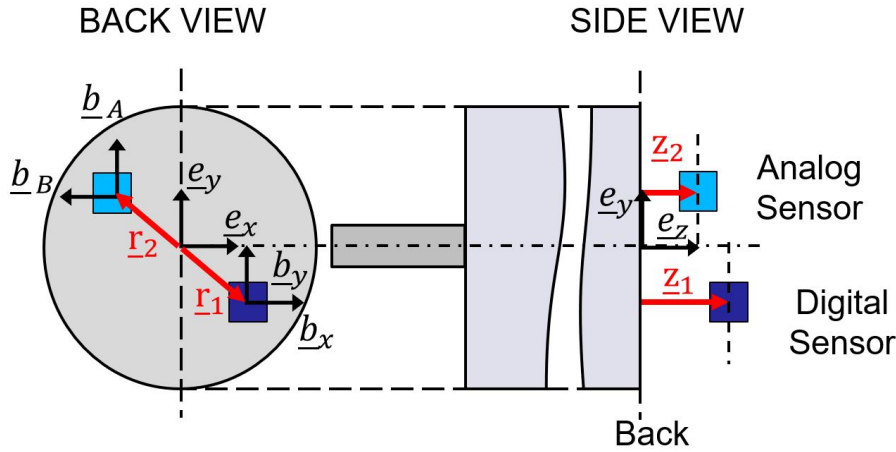
Figure 5.9: Overview of the Magnetic Sensor Positioning

5.3.1 Positioning of the Digital Magnetic Sensor

We observed mainly two tendencies concerning the dependency of the measurement with respect to the axial position:

(1) The higher the distance the lower the measurement amplitude.

(2) The higher the distance the higher the system noise.

In 5.10 the axial positioning of the digital sensor is visualized for a radial position $r = 12\,\text{mm}$, low speeds and zero torque load. In these graphs, the ground truth position is mapped to the magnetic measurement in x-, y- and z-direction. The left graph plots the measurement data taken at $50\,\text{mm}$ distance to the motor housing. We observe here a very high measurement noise and a small signal amplitude over one revolution.

The right graph plots the measurement data for a $20\,\text{mm}$ distance. We observe a lower noise but a signal amplitude that extends beyond the field range of the sensor. In the central graph, the optimal axial distance at $30\,\text{mm}$ is presented. At this distance, a good compromise among low noise and a reasonable dynamic range. Considering measurements at other axial positions, we select an axial distance of $30\,\text{mm}$ as the optimal position for the digital sensor.

Note that in this case, the field range of the digital sensor is $+/-1.3\,\text{gauss}$. This is the default adjustment of the sensor, which achieves the best resolution for such a field range. Yet, with an adjustment to a higher field range also closer axial positions could be realized with the digital sensor.

For the radial positioning we observed the following tendencies:

(1) The higher the distance to the rotation axis (or the closer we get to the stator) the higher the signal noise and the smaller the amplitude.

(2) The smaller the distance to the rotation axis, the smaller the signal amplitude.

In 5.11, we see an analogous visualization of the radial positioning for an axial distance z=30mm, low speeds, and zero torque load. Here, we exemplarily plot the measurement data for the radial distances 6 mm, 12 mm and 24 mm. The optimal position can be found at 12 mm radial distance. If we get closer to the rotation axis, the amplitude will decrease. If we get further away and closer to the stator area, the amplitude decreases and also the measurement noise gets higher.
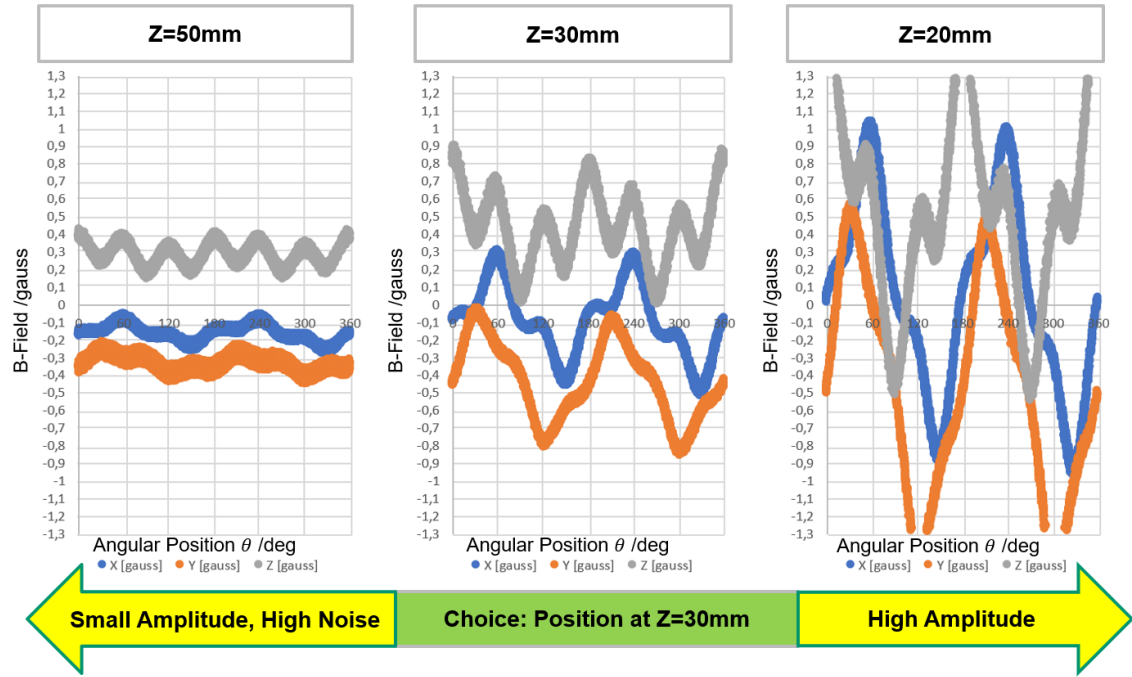
Figure 5.10: Axial Positioning of the Digital Sensor (Low Speed, Zero Load Torque, $r = 12mm$)

5.3.2 Positioning of the Analog Magnetic Sensor

For the optimal positioning of the analog sensor, we can use the observations made for the digital sensor. The optimization problem in radial direction is assumed to be analogously valid for the analog sensor. Therefore, we position the sensor mirrored to the digital sensor in radial direction utilizing the symmetry of the motor.
Due to constraints of the fixation setup for the sensors, we select a radial position of $r = 10.6$ mm instead of 12 mm. We also observed, regarding the digital sensor, that the closer the sensor is positioned to the housing in axial direction, the lower the noise. The analog sensor features the big advantage that the resistance in the feedback loop of the amplifier can be adjusted using the potentiometer. Accordingly, we can optimize the dynamic range of the sensor. We select a small axial distance of
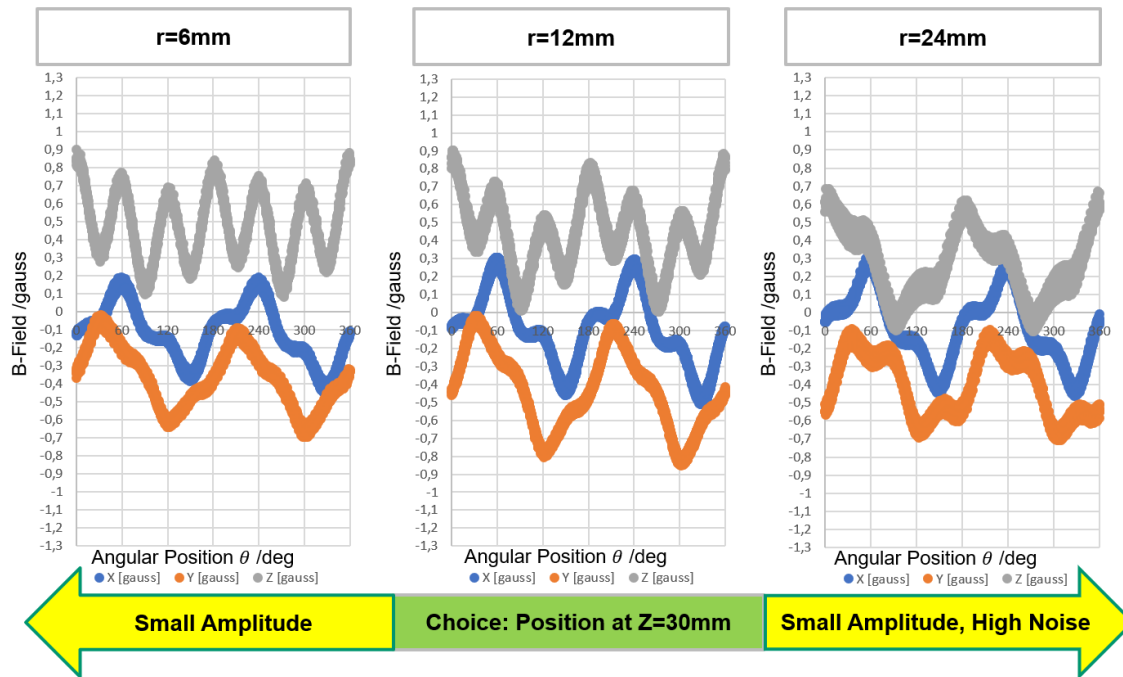
Figure 5.11: Radial Positioning of the Digital Sensor (Low Speed, Zero Load Torque, $z = 30mm$)
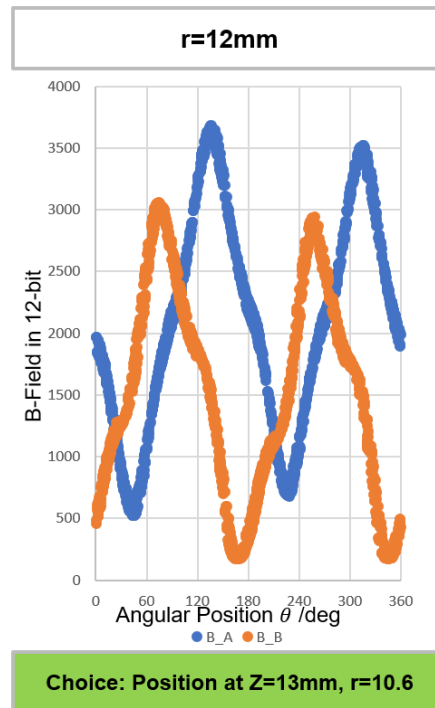

Figure 5.12: Positioning of 2-Axis Analog Magnetic Sensor (Low Speed, Zero Load Torque, Resistance=700kΩ

$z = 13\,\mathrm{mm}$ and, consequently, adjust the potentiometer to a feedback resistance of $700\,\mathrm{k\Omega}$. In 5.5 the optimal position of the analog sensor is visualized for low-speed and zero load torque.

CHAPTER 6

Evaluation

In this chapter, we present the results of the position estimation approach evaluated for real data from the experimental setup. First, we present the results using the digital three-axis compass at low rotation speeds. Afterwards, we present the results using the analog magnetic sensor for higher rotation speeds. Besides we will apply the measurement function with multivariate inputs.

6.1 Results of the Position Estimation Approach Using the Digital Compass

To evaluate the results using the digital compass, we first present the measurement function using Gaussian processes and B-splines. With this measurement function, we specify the measurement model of the Kalman filter. Then, the estimator is evaluated by comparing the combinations GP-EKF, Splines-EKF, GP-UKF, and Splines-UKF with the ground-truth position and speed. Besides, we compare the estimates of the angular position and speed with the ground truth.

6.1.1 Measurement Function

In chapter 4 we defined the regression approach for a measurement output $\mathbf{Y} = (\underline{b}_x, \underline{b}_y)$ in two spatial directions following the characteristics of the analog sensor. For the digital compass we just add a third direction $\underline{b}_z$ resulting in $\mathbf{Y}_{digital} = (\underline{b}_x, \underline{b}_y, \underline{b}_z)$. We already explained the restriction of the digital compass with respect to the rotation speed. Accordingly, the training set $\mathcal{D}_{digital} = (\underline{\theta}_{digital}, \mathbf{Y}_{digital})$ is taken for speeds from 0 to 200 rpm and contains 880 data points.

Gaussian Processes

Using the training set $\mathcal{D}_{digital}$ we first determine the hyperparameters of the Gaussian process by minimizing the negative logarithmic marginal likelihood. For that, we use the *minimize* Matlab function from the *GPML Library* by Rasmussen [14]. The hyperparameters l, σ_f and σ_n in the three measurement directions are given by

$$
\begin{aligned}
\underline{l} &= [0.2931\,\text{rad}, 0.2889\,\text{rad}, 0.4072\,\text{rad}]^T, \\
\underline{\sigma}_f &= [0.1731, 0.1742, 0.3982]^T, \\
\underline{\sigma}_n &= [0.0146\,\text{gauss}, 0.0147\,\text{gauss}, 0.0246\,\text{gauss}]^T.
\end{aligned}
\tag{6.1}
$$

With these hyperparameters we can specify the Gaussian process using the equations for prediction (3.17) and (3.18). The predicted mean and covariance functions for such a training set are shown in 6.1 for the measurement outputs in x-, y- and z-direction. We see that the GP mean functions (here referred as $B_{x,mean}$, $B_{y,mean}$ and $B_{z,mean}$) smoothly run through their associated noisy measurement points. The GP covariance functions are indicated by different color schemes around the mean function. In this case, 95% confidence bounds are visualized.
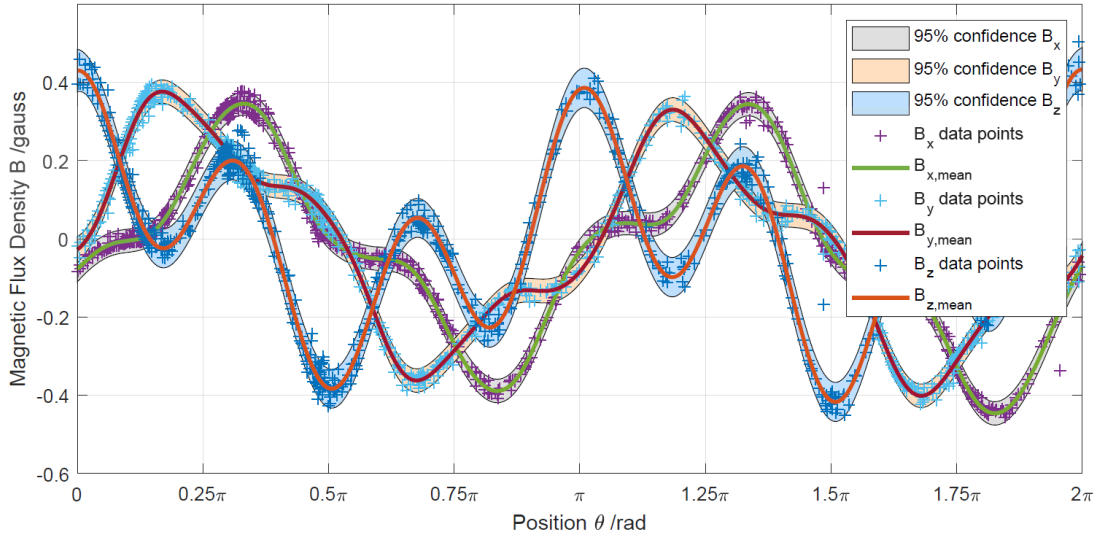

Figure 6.1: GP Predictive Mean and Covariance for Digital Compass Measurement Function with Respect to the Reference Position θ

We are interested if the predicted covariance function changes with respect to the reference position, or if it is approximately constant. In 6.2 the GP covariance is visualized for the three measurement outputs. We observe that the covariance functions are almost constant for all three measurement directions. Also, this constant value is quite accurately given by the hyperparameter noise standard deviation in all three directions $\underline{\sigma}_n = (0.0146\,\text{gauss}, 0.0147\,\text{gauss}, 0.0246\,\text{gauss})$. Accordingly, we

assume, that the measurement noise covariance is a diagonal constant matrix specified by the squared noise standard deviations of the GP. The measurement noise covariance is given by

$$\mathbf{R} = \begin{pmatrix} (0.0146\,\text{gauss})^2 & 0 & 0 \\ 0 & (0.0147\,\text{gauss})^2 & 0 \\ 0 & 0 & (0.0246\,\text{gauss})^2 \end{pmatrix}. \tag{6.2}$$
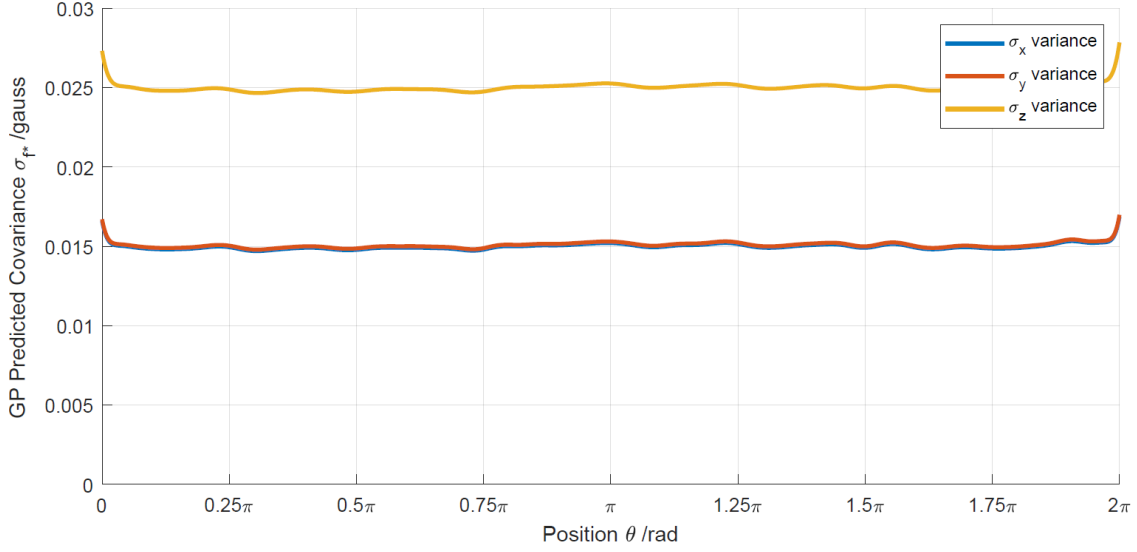

Figure 6.2: GP Predicted Covariance with Respect to the Reference Position θ

B-Splines

As a reference regression approach to Gaussian processes we apply B-splines for approximation. The same training set $\mathcal{D}_{digital}$ is utilized. We use the *spaps()*-function from the *Curve Fitting Toolbox* [15].

The B-splines, identified with the *spaps()*-function, need to be specified with a smoothing tolerance. This tolerance indicates, whether the regression function is overfitting the noisy data points, or the smoothing spline misses the actual data points. Using the hyperparameter σ_n of the GP, the tolerance of the B-splines was too high. Consequently, we identified the B-splines tolerance with an iterative approach.

Finally, we chose the tolerances for the three B-splines as

$$\underline{tol}_{splines} = [0.0024, 0.0015, 0.0044]^T. \tag{6.3}$$

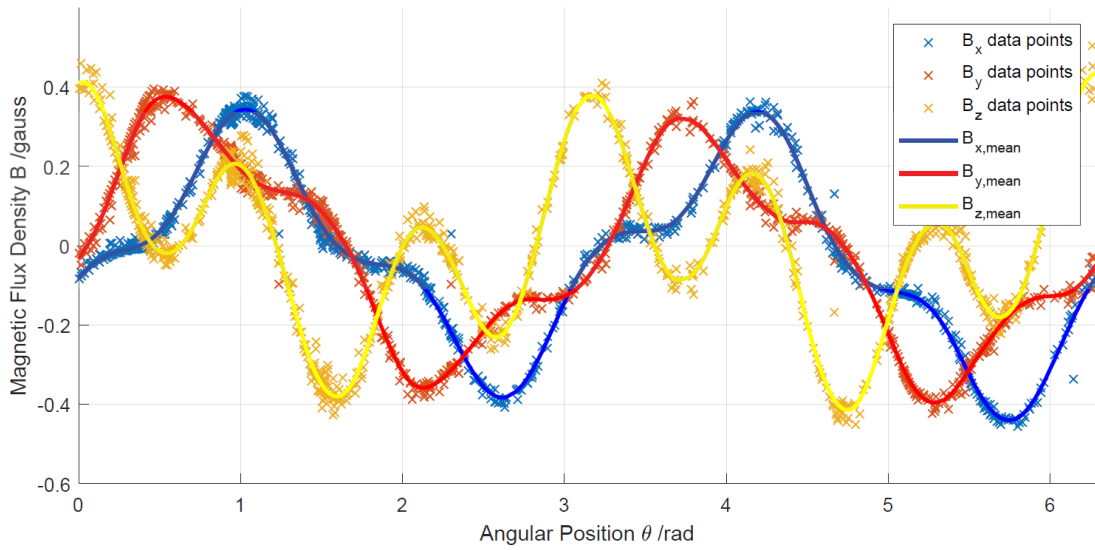The regression functions of the B-splines are shown in 6.3.

Figure 6.3: B-Splines Regression Functions for Digital Compass Measurement Function with Respect to the Reference Position θ

6.1.2 Results of the Position Estimation Approach

With the measurement functions from the previous regression approaches, we can now proceed to evaluate the position estimator. We specify the Kalman filter in the following way:

(1) The measurement function $\underline{f}(\theta)$ is given by the regression approaches.

(2) The measurement covariance $\mathbf{R}$ is given in (6.1).

(3) The system noise covariance $\mathbf{Q}$ will be derived with an iterative approach.

(4) The time interval between two measurements is $\Delta T = 0.005\,\text{s}$.

(5) The initial state is set to $\underline{x}_0 = [0, 0]^T$ with a high initial covariance $\mathbf{P}_0 = diag((4\pi\,\text{rad})^2, (40\,\text{rad/s})^2)$.

To identify the system noise, we performed estimations for all combinations of 20 different values for the position noise and 12 different values for the speed noise. We selected the combination that provided the best average position estimate over several random estimation tests. The selected system noise is

$$\mathbf{Q} = \begin{pmatrix} (0.025\,\text{rad})^2 & 0 \\ 0 & (1.5\,\text{rad/s})^2 \end{pmatrix}. \tag{6.4}$$

With these specifications, we run the estimation approach by combining GP or B-splines regression with EKF or UKF. We visualize the tracking results for 100 test points, but evaluate each combination with tracking results from 5000 testing points

taken at rotation speeds up to 200 rpm.

First, we present the results of a combination of GP regression and EKF. In 6.4 the position estimation results are visualized. We see that due to a high initial estimation-error covariance the position estimate very quickly tracks the ground truth position. After around 12 time-steps, the GP-EKF provides a very accurate estimation result while the estimation error gets very small.

In 6.5 the speed estimation results are presented. The speed estimation is much less accurate compared to the position estimation. In this case, this can be related to the fact that the measurement function does not provide any relationship with respect to the rotation speed. After around 12 time-steps the speed estimation tracks the ground truth speed sufficiently though featuring an estimation latency.

We compare these results with a combination of B-splines and EKF. We use the same training set containing 100 data points. The results of the position estimation are given in 6.6 and the results of the speed estimation are visualized in 6.7. The position estimation using B-splines-EKF is also quite accurate, though it takes around 20 time steps to reliably track the ground truth. The same observation is valid for speed estimation.
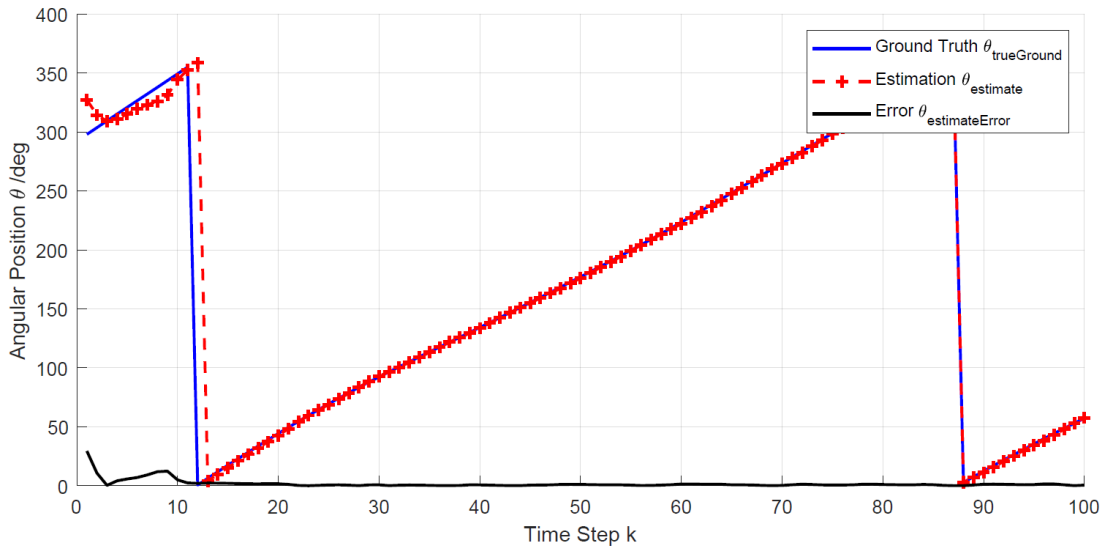

Figure 6.4: GP-EKF Position Estimation

For a GP-UKF and B-splines-UKF combination an analogous distribution is observed to the B-splines-EKF combination, featuring an initial tracking of around 20 time-steps.

Now we quantitatively evaluate the different combinations with respect to the position estimate, the speed estimate and the computing time for a test set of the size 1000. To eliminate the impact of the initial tracking we ignore the estimates of the first 50 time steps.
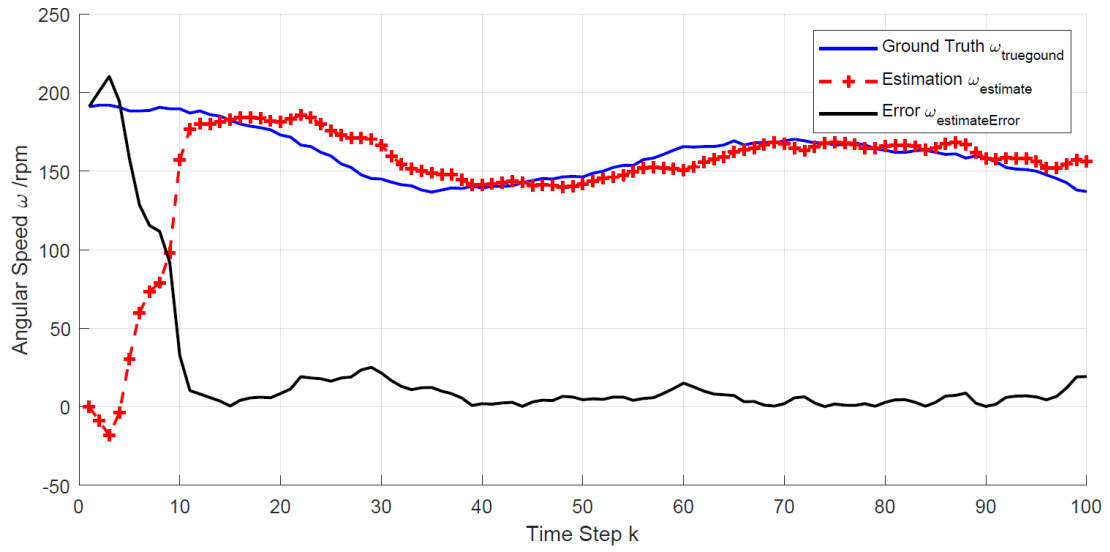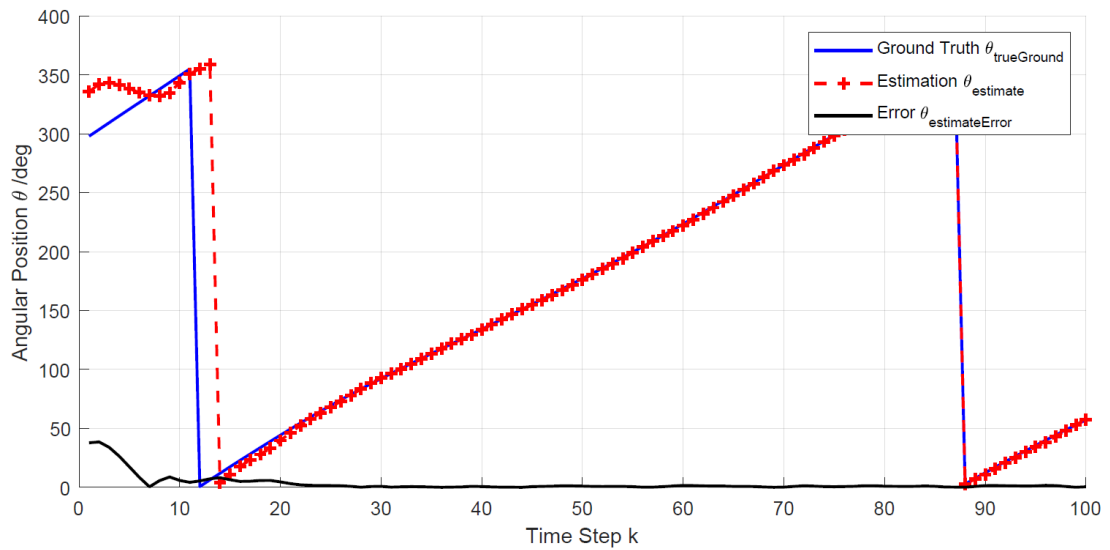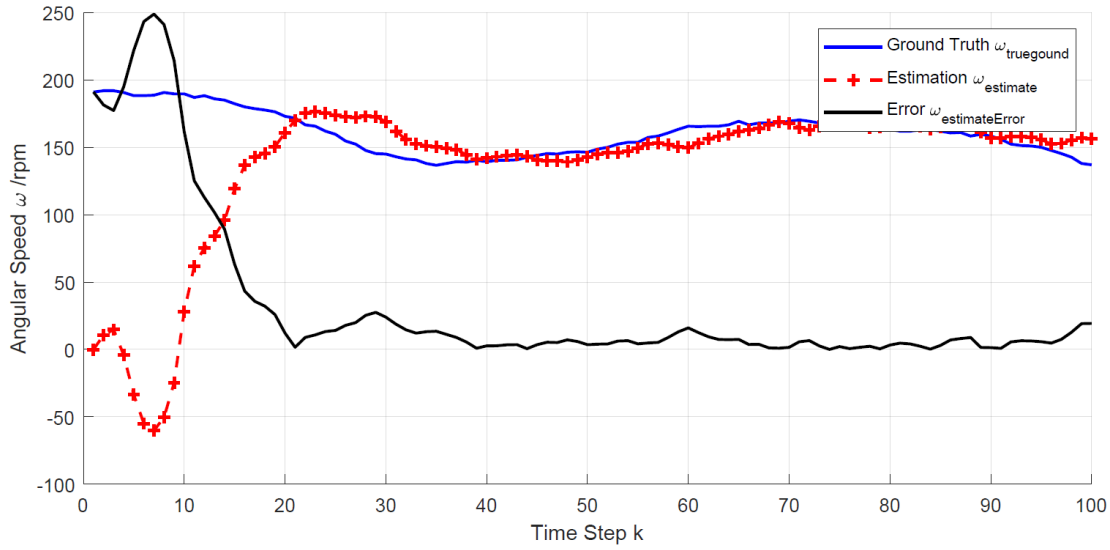
Figure 6.5: GP-EKF Speed Estimation

Figure 6.6: B-Splines-EKF Position Estimation

Figure 6.7: B-Splines-EKF Speed Estimation

These results are presented in 6.1, where the standard deviation of the position estimate error $\sigma_{err,pos}$, the standard deviation of the speed estimate error $\sigma_{err,speed}$ and the computing time per estimation step are listed for each estimation model. All models feature a position estimate error of around 1 deg and a speed estimate error around 12.5 rpm. The best position accuracy is achieved by the Splines-EKF model with 0.88 deg closely followed by GP-EKF with 0.90 deg.

However the GP-EKF model surprisingly needs the smallest computing time of 1.1 ms. It also provides the smallest speed estimation error of 12.37 rpm. Consequently, the GP-EKF model is selected to be the best estimation approach.

Estimation Model	Position Error $\sigma_{err,pos}/\text{deg}$	Speed Error $\sigma_{err,speed}$ /rpm	Comp. Time T_c /ms
GP-EKF	0.90	12.37	1.1
Splines-EKF	0.88	12.38	2.4
GP-UKF	1.08	12.51	1.6
Splines-UKF	1.16	12.63	2.1

Table 6.1: Quantitative Evaluation of the Estimation Models

A position estimation error of around 0.9 deg is already a good result for this approach. For most motor controllers, an accuracy of 60 deg is already enough to commutate a motor. The more critical point is the sampling rate of the sensor. The estimation approach is required to provide reliable results for any operational condition such as high angular speeds. At 2000 rpm the sensor can barely provide 5 measurements over one rotation for operation at 160 Hz. This makes it impossible

to reliably estimate the position and speed for all operating conditions.

We also observe a second problem. Due to the symmetric magnetic field distribution of the rotor, the position estimation approach occasionally tracks a position which is 180 deg shifted towards the ground-truth position. In 6.8 we see such a case, where the estimator detects the shifted symmetric position.
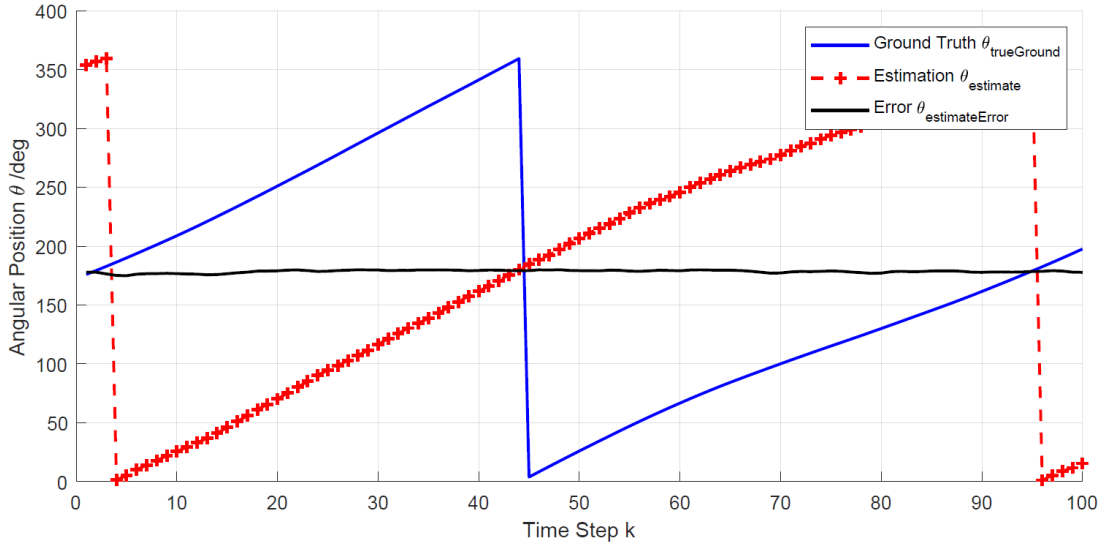

Figure 6.8: GP-EKF Position Estimation 180 deg Shifted

However, the magnetic field is not perfectly symmetric. Accordingly, the measurement model, which is based on the ground truth position, maps the shifted position not as accurate to the magnetic field.

This splits up the tracking of the position in two cases. Either the estimator follows the ground truth position (65% of the cases) or it follows the 180 deg shifted position (35% of the cases). For the first case, we will get the estimation accuracy from 6.1. For the second case, the position estimation error is around twice as high than in 6.1. The speed estimation error approximately remains the same.

This problem is not critical if the position estimation is used as a motor-feedback-system. Due to the two-pole-pair rotor, the position of the rotor can be identified with an electrical angle. The electrical angle is periodical over 180 deg instead of 360 deg of the mechanical angle. If the mechanical angle is required, for example, to estimate the pose of the technical system, this shifting might be critical. However, the 180 deg shifting only occurs at the startup time of the estimator. The relative angular position estimation remains accurately.

6.2 Results of the Position Estimation Approach Using the Analog Magnetic Sensor

The estimation approach using the digital compass cannot be validated for higher speeds due to the small sampling rate of the sensor board. Hence, we extend the evaluation to the second analog magnetic sensor providing a higher sampling rate. We first identify the measurement function. We use Gaussian processes with a univariate measurement input for different rotation speeds. Due to an observed dependency of the magnetic field with respect to the angular speed, we apply Gaussian processes with multi-variate measurement input. The training set $\mathcal{D}_{analog}$ will be used containing the ground truth position and speed. Then, we evaluate the position estimation approach by testing with different angular speeds and comparing the estimation with the ground truth.

6.2.1 Measurement Function with Univariate Input

So far, we assumed that the magnetic field measurement is independent with respect to the angular speed. With the analog sensor, we now have the possibility to detect the magnetic field at high speed to investigate this assumption. Therefore, we computed several GP measurement functions using training sets with different rotation speeds. If this assumption is correct, all these measurement functions should be almost identical.

We use several training sets $\mathcal{D}_{i,analog}$ with 600 data points. Each of this training set is taken at a constant rotation speed between $-2000\,\text{rpm}$ and $2000\,\text{rpm}$. A positive speed indicates a counterclockwise rotation in the direction of the output shaft of the motor.

We present the noise standard deviation σ_n for some exemplary GP regression functions in 6.2. The noise standard deviation σ_n gives information about the measurement noise of the function. The magnitudes of the noise standard deviations in x-direction $\sigma_{n,x}$ and in y-direction $\sigma_{n,y}$ are relatively low for any of the GP functions. Even for 2000 rpm, the correlated measurement noise would be only around 0.5% of the measurement amplitude.

Speed ω/rpm	-2000	-500	+500	+2000
$\sigma_{n,x}$	0.0052	0.0034	0.0043	0.0080
$\sigma_{n,y}$	0.0065	0.0039	0.0054	0.0083

Table 6.2: Noise Standard Deviation for Different GP Regression Functions

The curves of the GP functions for different angular speeds can be seen in 6.9. The

figure shows that we observe a dependency of the magnetic field to the angular speed. The displayed functions map the angular position θ to the magnetic field. In this case, the magnetic field is a dimensionless quantity from -1 to 1. This is caused by an assembled amplification circuit for the analog sensor with adjustable potentiometers. Accordingly, a magnetic field unit such as *gauss* is not validated. The functions for the same rotation speed but in different directions (such as $-2000\,\mathrm{rpm}$ and $2000\,\mathrm{rpm}$) are displayed in the same color. The function shifted to the right direction is always the one with a positive angular speed.

Mainly, the following observations can be made:

(1) The measurement curve is shifted up to $20\,\mathrm{deg}$ for positive angular speeds. The curve is shifted up to $-20\,\mathrm{deg}$ for negative angular speeds

(2) The higher the speed either in positive or in negative direction the smaller the amplitude of the measurement curve. The amplitude decreases up to 22% of the original magnitude.

(3) Lastly, the shape of the curve changes. For positive speeds, the bending of the curve in between a minimum and a maximum increases. For negative high speeds, such a bending almost disappears.

What could be the reason for this dependency? We draw three hypotheses to explain the observation:

(1) The microcontroller reads the sensors with an additional latency that we, yet, did not consider. This could cause a shifting of the curve.

(2) The magnetic sensor could have limitations for very fast changing magnetic fields.

(3) The measured magnetic field indeed reflects the actual magnetic field distribution at the back side of the motor. The dependency with respect to the speed may appear due to the material in between the sensor and the rotor. Other effects that occur strongly at high speeds can also influence the magnetic field.

The first hypothesis can be disproved. Primarily, the time interval required to measure both the reference sensor and the two analog outputs could be restricted to $16\,\mu\mathrm{s}$. At $2000\,\mathrm{rpm}$, the rotor can turn to a maximum of $0.2\,\mathrm{deg}$ for such a time interval. Therefore, this effect can not explain the shifting of up to $20\,\mathrm{deg}$. Additionally, the curve is shifted in positive direction for positive speeds. If the shifting was caused by latency, the curve should be shifted in negative direction.

For the second hypothesis we can only refer to the advertised bandwidth of $3\,\mathrm{MHz}$ supported by the sensor. However, we can not preclude that the sensor setup somehow influences the measurement results. Another influence on the measurement
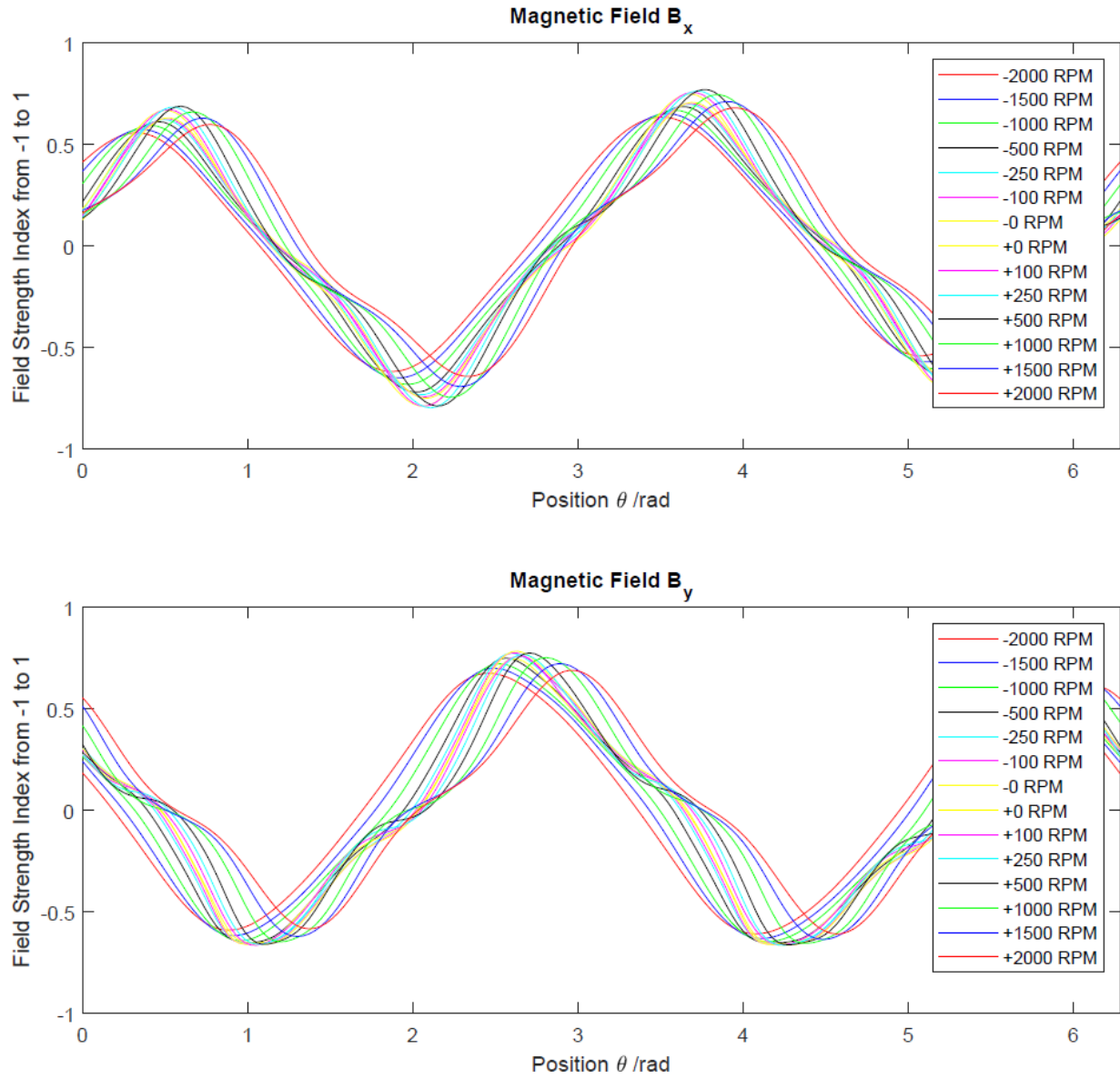
Figure 6.9: GP Regression Functions for Different Rotation Speeds

output could be the magnetic bias of the sensor.

The third hypothesis can be motivated by several effects. For example, Eddy currents could occur in the material at the back side of the motor due to the time-variant magnetic field.

In this work we could not gain certainty, which effect causes the dependency on the speed. Also it is not clear, why the shape of the measurement curve changes differently for positive and negative speeds.

However, we can still work with such an effect. Even though the shape of the measurement curve changes for different speeds, the noise standard deviation σ_n of each GP predictive function for constant speed is very low. Therefore, the dependency of the magnetic measurement, with respect to the angular position and speed, can be mapped.

6.2.2 Measurement Function with Multivariate Input

Previously, we observed that the magnetic field measurements depend both on the angular position and speed. Accordingly, we derive a measurement function mapping the angular position θ and speed ω to the magnetic field. We use the training set $\mathcal{D}_{analog} = \langle \mathbf{X}, \mathbf{Y} \rangle$ containing 1070 data points taken at $-2000\,\mathrm{rpm}$ to $2000\,\mathrm{rpm}$ angular speed. The training input $\mathbf{X} = [\underline{\theta}, \underline{\omega}]$ is a matrix containing the ground truth position and speed. Moreover, the training output $\mathbf{Y} = [\underline{b}_x, \underline{b}_y]$ is a matrix containing the magnetic measurements in two spatial directions.

For such a training set we first determine the hyperparameters of the multivariate input Gaussian process. The noise standard deviation is $\sigma_{n,x} = 0.0103$ in x-direction and $\sigma_{n,y} = 0.0098$ in x-direction. Consequently, the measurement noise of the multivariate input GP covering all the speeds is only a little bit higher than the measurement noise of the univariate GP functions with constant speed.

The GP predictive mean function is given in 6.10 in x-direction and in 6.11 in y-direction. The functions map the angular position θ and the angular speed ω to the measurement output. We observe that the GP predictive mean functions smoothly approximate the data points over the multivariate input space.

Note that the speed ω is divided by a factor 200. This is performed due to the non-equal properties of the input values. With this factor, the boundaries of the position ($0\,\mathrm{rad}$ to $2\pi\,\mathrm{rad}$) and the speed ($-210\,\mathrm{rad/s}$ to $210\,\mathrm{rad/s}$) are aligned. Similar results could be achieved by introducing further hyperparameters to the kernel function of the GP.
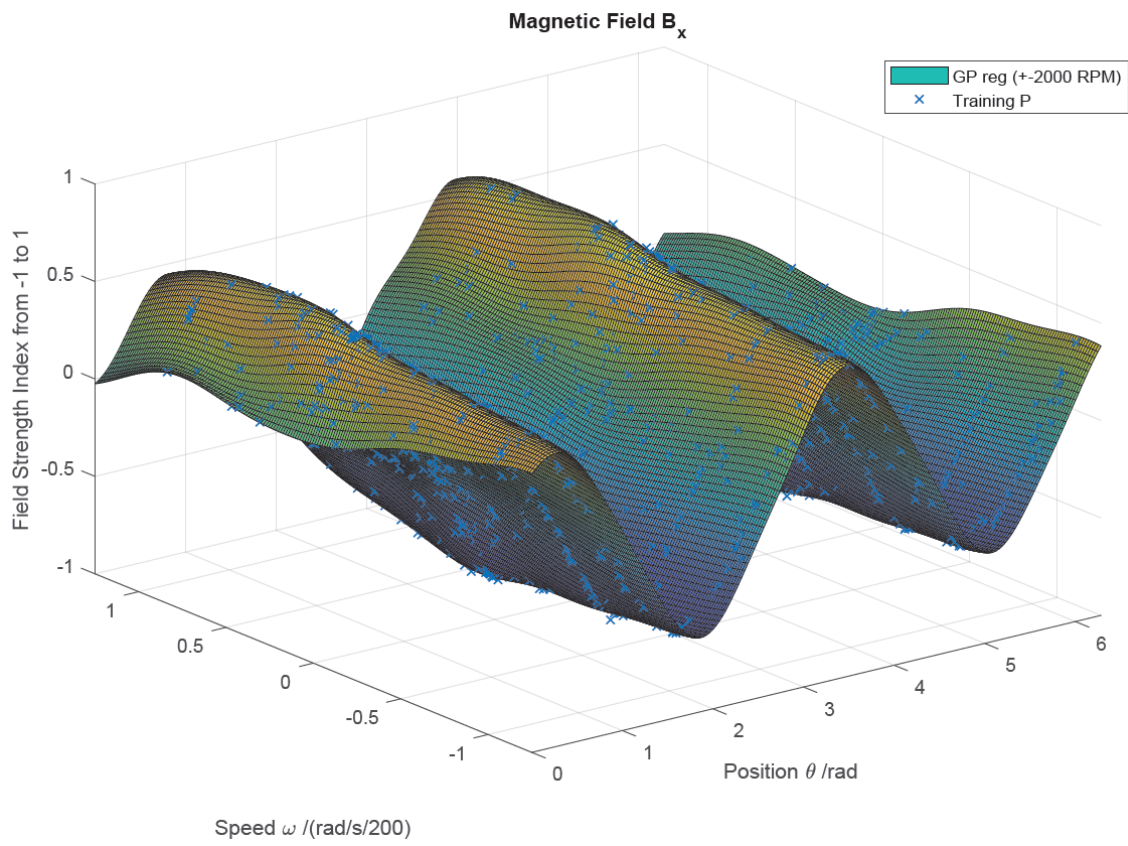
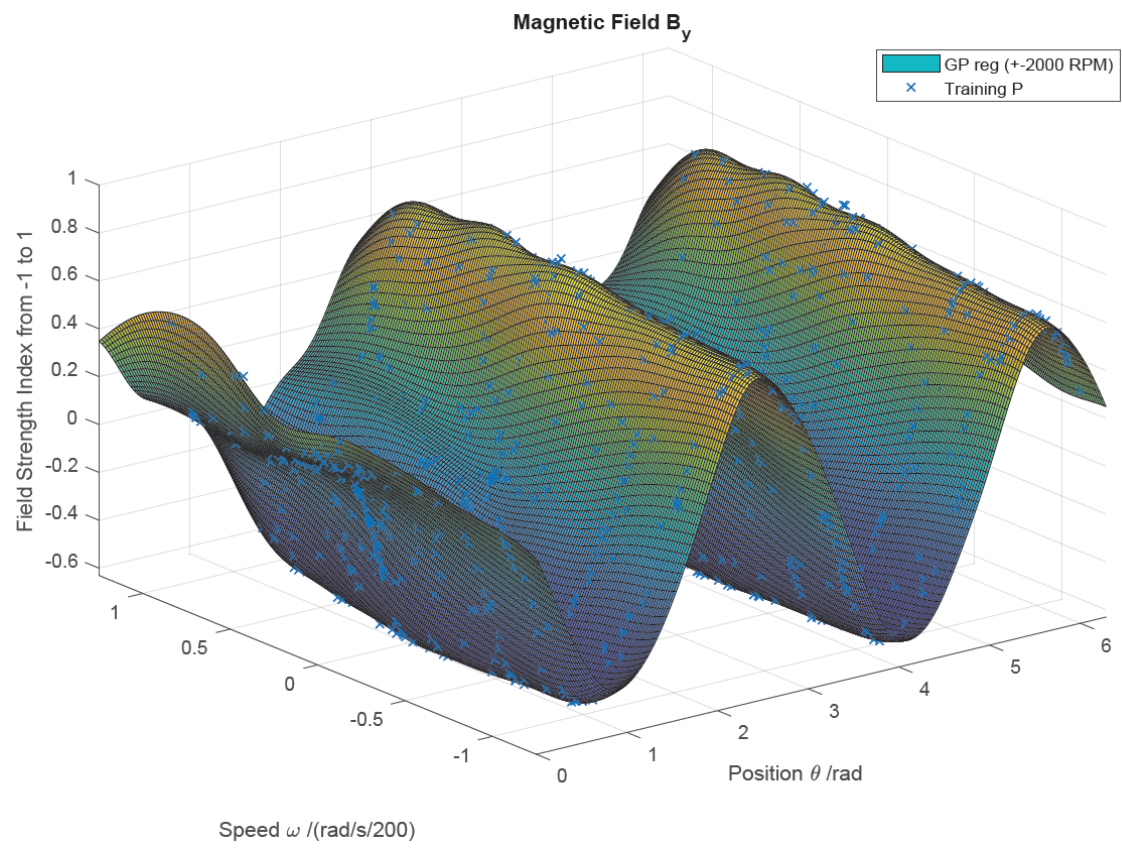**Figure 6.10:** GP Predictive Mean Function in B_x for Multivariate Inputs

Figure 6.11: GP Predictive Mean Function in B_y for Multivariate Inputs

6.2.3 Results of the Position Estimation Approach

Using a multivariate measurement function the properties of the Kalman filter do not change much compared to a univariate approach. We list the specified features in the following:

(1) The measurement function $\underline{h}(\theta, \omega)$ is now dependant on both state variables θ and ω.

(2) The measurement covariance is given by $\mathbf{R} = diag(\sigma_{n,x}^2, \sigma_{n,y}^2)$.

(3) The system noise covariance is $\mathbf{Q} = diag((0.01\,\mathrm{rad})^2, (6\,\mathrm{rad/s})^2)$.

(4) The time interval between two measurements is $\Delta T = 0.0022\,\mathrm{s}$.

(5) The initial state is set to $\underline{x}_0 = [0, 0]^T$ with the covariance
$\mathbf{P}_0 = diag((4\pi\,\mathrm{rad})^2, (200\,\mathrm{rad/s})^2)$.

Similarily to the estimation approach using the digital compass, we simplify in the measurement noise covariance in (2) to be a constant diagonal matrix. This is not valid in general. We could use the GP predictive covariance to identify such a non-linear measurement noise covariance containing non-diagonal cross terms. However, we could verify that this simplified assumption approximates the measurement noise covariance well. Besides, applying this simplification allows us to save a lot of computing time. Especially for extensive training sets, this may be the case. The system noise covariance in (3) is selected experimentally analogous to the one for the digital compass.

We can now evaluate this estimation approach with several testing data for different angular speeds. For a first overview, we show a typical tracking process for angular speeds around 1700 rpm at zero load torque. We here combine the GP measurement function with an EKF. The results of the position and speed estimation for the first 200 time-steps are visualized in 6.12 and 6.13. We see that, in order to reliably track the ground truth position, the estimator needs around 30 time-steps from the initial state. This is achieved even for very high initial ground truth speeds around 1700 rpm. Thenceforward, the position estimation is very close to the ground truth position.

However, the speed estimation exhibits a quite high estimation error for high angular speeds. The speed estimation curve is also a bit shifted with respect to the ground truth speed. This is mostly associated with a high-frequency alteration of the ground truth speed. The reason for this could be a poor commutation technique with the motor controller or some nonlinear friction effect. We could solve this observed shifting by applying a higher sampling rate. Therewith, the estimator could react faster on high-frequency changes of the angular speed.
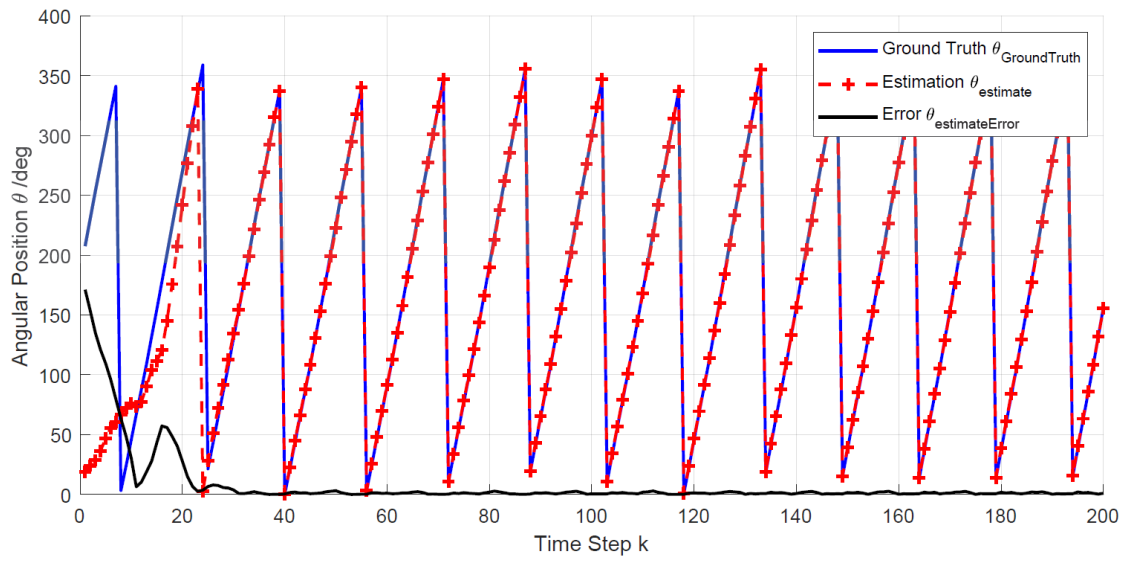
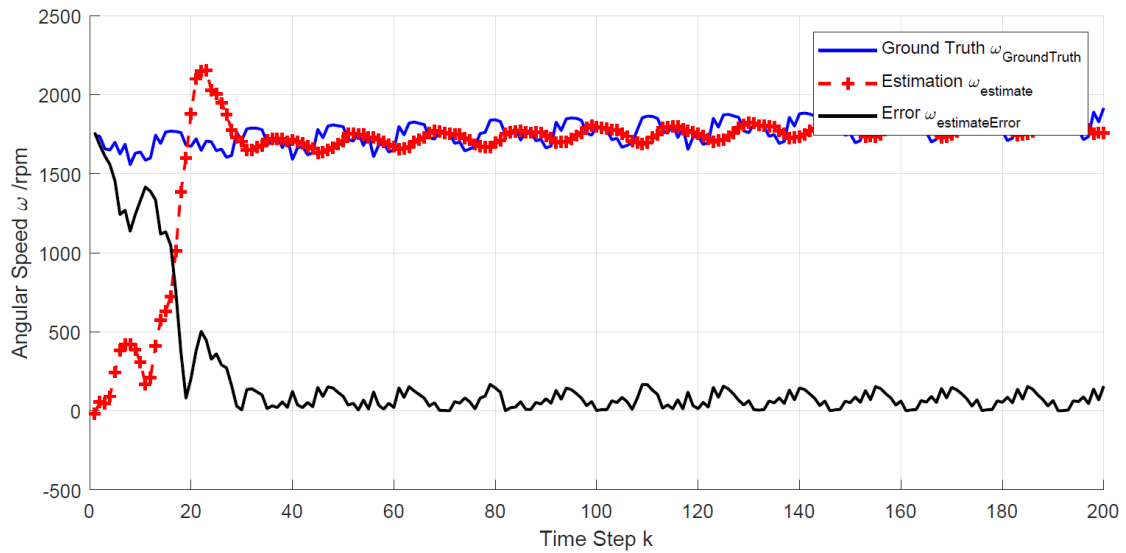**Figure 6.12:** Multivariate Input GP-EKF Position Estimation

Figure 6.13: Multivariate Input GP-EKF Speed Estimation

We now want to quantitatively evaluate the estimation approach for testing data at different rotation speeds with 1000 data points. The tests are performed for operational conditions at zero, mid and high load torque M_{load}. Tests at 100 rpm can only be executed at zero load torque because the motor is not strong enough to rotate at low speeds with the applied load torques. Along, we evaluate at 500, 1000, 1500 and 2000 rpm. The evaluated parameters are the standard deviation of the position estimate error $\sigma_{err,pos}$, the standard deviation of the speed estimate error $\sigma_{err,speed}$ and the computing time T_{comp}. We use both the GP-EKF and the GP-UKF estimation approach. The results are given in 6.3 for operation at zero load, in 6.4 at mid load, and in 6.5 at high load.

Operational Condition with Zero Load $M_{L,eff} = 0\,\mathrm{Nm}$				
Estimation Model	Testing Speed $\omega_{test}/\mathrm{rpm}$	$\sigma_{err,pos}/\mathrm{deg}$	$\sigma_{err,speed}/\mathrm{rpm}$	T_{comp}/ms
GP-EKF	100	0.756	7.89	1.3
GP-EKF	500	0.760	12.36	1.3
GP-EKF	1000	0.742	35.68	1.3
GP-EKF	1500	1.121	50.22	1.3
GP-EKF	2000	1.213	69.73	1.3
GP-UKF	100	0.798	6.80	1.6
GP-UKF	500	0.712	12.17	1.6
GP-UKF	1000	0.723	35.22	1.6
GP-UKF	1500	1.109	49.49	1.6
GP-UKF	2000	1.197	69.42	1.6

Table 6.3: Evaluation of the Estimation Approach for Zero Load

Both the GP-EKF and the GP-UKF approach feature acceptable computation times for one estimation step. Still, the GP-EKF is with 1.3 milliseconds per step a little bit faster than the GP-UKF. On the other hand, the GP-UKF estimation features a slightly better accuracy than the GP-EKF for almost any operational condition. A reason for that is that the linearization of the EKF can give poor results for highly nonlinear curves. This effect occurs stronger for higher speeds and a constant sampling rate.

We now want to take a closer look at the position estimate error. We visualize the quantitative evaluation results for the position estimation error in 6.14.

In this graph, we map the angular speed of the motor to the position estimation error. In any case, the UKF (dashed lines) provides a more accurate estimate than the EKF (dotted line). At zero and mid load torque, the position estimation error increases for higher rotation speeds. This is expected because for higher rotation

Operational Condition with Mid Load $M_{L,eff} = 0.36\,\mathrm{Nm}$				
Estimation Model	Testing Speed ω_{test}/rpm	$\sigma_{err,pos}$/deg	$\sigma_{err,speed}$/rpm	T_{comp}/ms
GP-EKF	100	—	—	—
GP-EKF	500	0.929	22.69	1.3
GP-EKF	1000	1.236	46.90	1.3
GP-EKF	1500	1.370	72.95	1.3
GP-EKF	2000	1.453	88.35	1.3
GP-UKF	100	—	—	—
GP-UKF	500	0.818	20.67	1.6
GP-UKF	1000	1.109	45.96	1.6
GP-UKF	1500	1.219	70.85	1.6
GP-UKF	2000	1.464	90.15	1.6

Table 6.4: Evaluation of the Estimation Approach for Mid Load

Operational Condition with High Load $M_{L,eff} = 0.72\,\mathrm{Nm}$				
Estimation Model	Testing Speed ω_{test}/rpm	$\sigma_{err,pos}$/deg	$\sigma_{err,speed}$/rpm	T_{comp}/ms
GP-EKF	100	—	—	—
GP-EKF	500	1.267	25.33	1.3
GP-EKF	1000	1.114	46.29	1.3
GP-EKF	1500	1.472	67.90	1.3
GP-EKF	2000	1.580	73.27	1.3
GP-UKF	100	—	—	—
GP-UKF	500	1.250	25.00	1.6
GP-UKF	1000	1.045	45.97	1.6
GP-UKF	1500	1.229	60.30	1.6
GP-UKF	2000	1.581	73.10	1.6

Table 6.5: Evaluation of the Estimation Approach for High Load

speeds the number of measurements per rotation decreases due to a constant sampling rate.

However, at high load torque the position estimation error is relatively high for 500 rpm rotation speed. We assume that this appears because of the limited abilities of the friction brake system. Especially for low speeds, the high load torque applied from the brake system on the motor varies largely. That results in fast acceleration and then a sudden deceleration of the motor at 500 rpm and high load. This effect also declines the accuracy of the estimation.
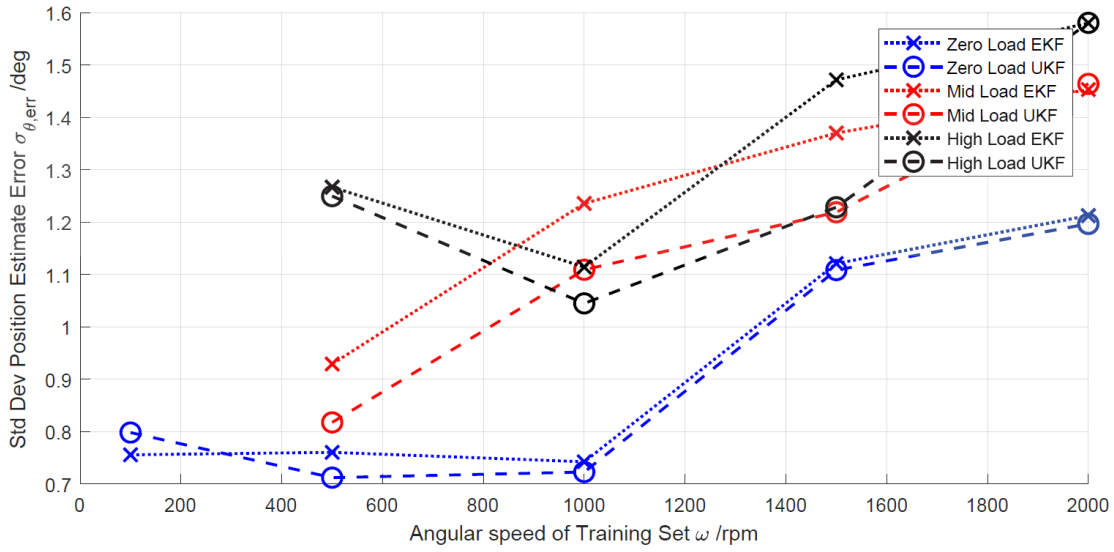
Figure 6.14: Position Estimation Error for Different Operational Conditions

After all, for all operational conditions we can verify a position estimation error below 1.6 deg. Besides, for low torques and angular speeds below 1000 rpm, we can even reduce the position estimation error to less than 0.8 deg. In the same way, we visualize the speed estimation results in 6.15.
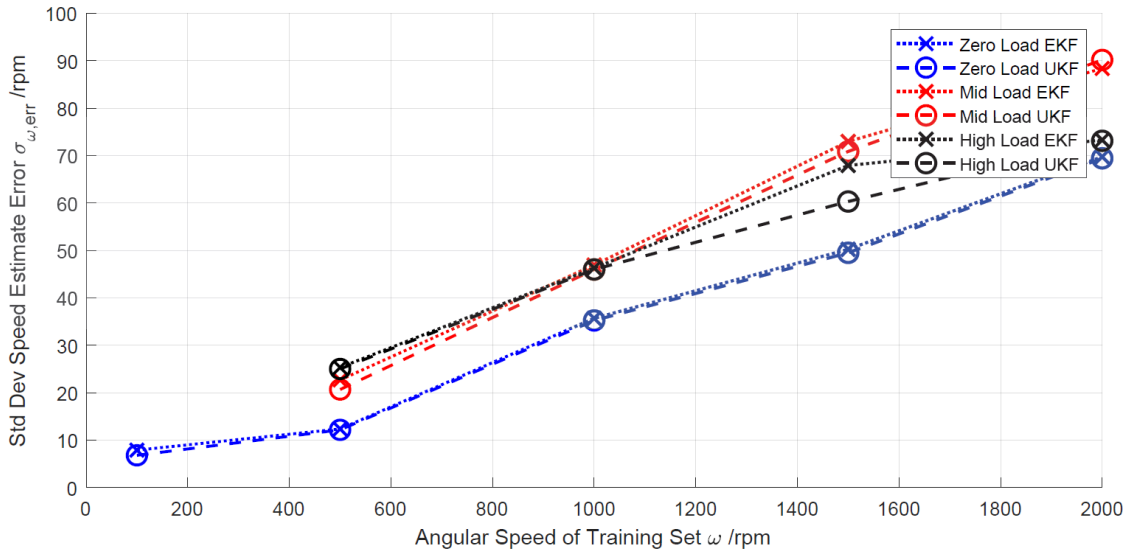

Figure 6.15: Speed Estimation Error for Different Operational Conditions

Here, we map the angular speed of the training set to the standard deviation of speed estimation error. The results of the UKF and EKF are approximately identical. We observe that the speed estimation error almost linearly increases for higher rotation speeds. The operation with load torques scales the position error only insignificantly.

At speeds below 500 rpm and zero load torque, the speed estimation error is around 10 rpm. For high speed and high load torque operating conditions, the speed estimation error increases to 90 rpm. This is a relatively large speed estimation error. Yet, we can compute the relative speed estimation error with respect to the angular speed. This is performed in 6.16. We observe that the speed estimation error is below 5% for any operating condition. For zero load torques, the speed accuracy remains below 3.5%. Nonetheless, for operation at low speeds (100 rpm), the speed estimation error increases until 8%.
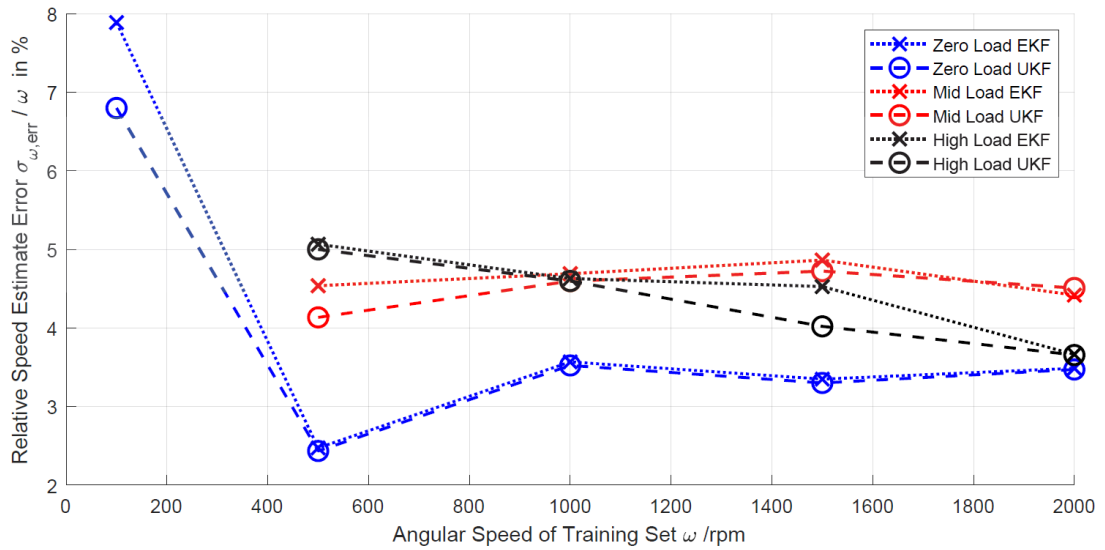

Figure 6.16: Speed Estimation Error for Different Operational Conditions

A reason for a relatively high position error could be the still quite small sampling rate of 455 Hz both for the magnetic sensor and the reference sensor. A higher sampling rate could be realized by using a microcontroller that provides a faster operating serial USB interface. Alternatively, the position estimation approach could be directly implemented on a microcontroller, on which a high sampling rate of 62.5 kHz is already available.

Secondly, though the reference sensor provides a high resolution angular position, the angular speed is just computed from this position. Therefore, the derived angular speed could indeed contain a relatively high noise. Considering this angular speed to be the ground truth, the position estimation error could increase.

Another open issue is that in some cases the estimator tracks a 180 deg shifted position with respect to the ground truth position. In such cases, the accuracy of the position and speed estimation decreases.

CHAPTER 7

Conclusion

In the field of robotics and control systems, the position and speed of electrical motors need to be identified to facilitate high performance control applications. In several literature it has been shown, that Gaussian process regression models can be successfully integrated into Bayesian filters such as the EKF and the UKF [3].

In this thesis, we developed and evaluated a method to estimate rotor position and angular velocity for BLDC motors. This method is based on the measurement of magnetic field at a fixed position of the stator produced by the rotor. We used a Kalman filtering algorithm for state estimation. The corresponding measurement function, mapping the rotor state to the observed values of the magnetic field, was obtained by GP predictive regression functions, using a measured training data set. Due to a highly nonlinear nature of the measurement function, we integrated the system model into EKF and UKF framework.

An experimental setup was assembled, in which two different magnetic sensors were placed on the back side of a BLDC motor. For that, we identified a position of placement for the magnetic sensors with an maximal signal-to-noise ratio. Besides, we conducted tests at realistic operating conditions by building a friction brake system to record measurements for different torque loads and angular speeds.

We expected the magnetic measurements to be only dependent on the angular position of the rotor. For measurements at high angular speeds, we observed a dependency of the magnetic field measurement with respect to the angular speed. Therefore, we identified a multivariate GP predictive measurement function, additionally taking the angular speed into account.

We achieved less than 1.6 deg of error in the position estimation using our approach. Along with positions, our approach also estimates the angular speed. For low angular speeds, the speed estimates exhibited an error around 10 rpm (8%), while this error increased to 90 rpm (5%) for high speed applications. The computing time for each estimation step was less than 1.6 ms. The evaluation of EKF and UKF approaches in terms of accuracy and computational performance shows, that UKF

achieves more accurate results than EKF. However, this comes at a cost of increased computing time.

With this thesis, several fields for further research can be investigated. Firstly, a study analyzing the dependency of the magnetic field, with respect to the angular speed, could be executed. The influence of Eddy-currents on the magnetic field distribution can be detected.

Secondly, based on this experimental setup, a research could optimize the performance of the estimation approach by installation of a second magnetic sensor. Recently, a promising paper about magnetic odometry has been publish concerning such a setup with a magnetic sensor array [23].

Yet, other technical issues can be optimized. The recording of data points can be conducted with a higher sampling rate. Besides, a more accurate way can be established to measure the reference angular speed.

Bibliography

[1] Herbert Bernstein. *Messelektronik und Sensoren.* Springer Vieweg, Wiesbaden, 2014.

[2] C. E. Rasmussen, C. K. I. Williams. *Gaussian Processes for Machine Learning.* the MIT Press, Massachusetts, 2006.

[3] Jonathan Ko, Dieter Fox. *Bayesian Filtering Using Gaussian Process Prediction and Observation Models.* In: Autonomous Robots, Volume 27, pp. 75–90, Springer Science+Business Media, 2009.

[4] Dan Simon. *Optimal State Estimation.* Wiley Interscience, Hoboken, New Jersey, 2006.

[5] Padmaraja Yedamale. *Brushless DC (BLDC) Motor Fundamentals.* Microchip Technology Incorporated, USA, 2003.

[6] ams AG. *AN5000Rotary Magnetic Position Sensors - Magnet Selection Guide.* `https://ams.com/documents/20143/36005/AnglePositionOnAxis_AN000271_2-00.pdf/`, [Online; accessed 05-June-2019].

[7] Ekbert Hering. *Sensoren in Wissenschaft und Technik.* Springer Vieweg, Wiesbaden, 2018.

[8] Pramod K. Meher. *50 Years of CORDIC: Algorithms, Architectures,and Applications.* In: IEEE Transactions on Circuits and Systems I, Regular Papers, Volume 56, pp. 1893 - 1907, IEEE, 2009.

[9] Uwe D. Hanebeck. *Informationsverarbeitung in Sensornetzwerken.* Lecture Script, Karlsruhe Institute of Technology, Karlsruhe, 2019.

[10] Hartmut Prautzsch, Wolfgang Boehm, Marco Paluszny. *Bezier- and B-spline techniques.* Springer, 2002.

[11] Uwe D. Hanebeck. *Stochastische Informationsverarbeitung.* Lecture Script, Karlsruhe Institute of Technology, Karlsruhe, 2019.

[12] Dan Simon. *Kalman Filtering*. In: Embedded Systems Programming, pp 72-79, 2001.

[13] Jannik Steinbring. *The Nonlinear Estimation Toolbox*. `https://nonlinearestimation.bitbucket.io/`, [Online; accessed 05-June-2019].

[14] Carl Edward Rasmussen and Chris Williams. *Documentation for GPML Matlab Code version 4.2.* `http://www.gaussianprocess.org/gpml/code/matlab/doc/`, [Online; accessed 05-June-2019].

[15] MathWorks Inc. *How to Construct Splines*. `https://de.mathworks.com/help/curvefit/examples/how-to-construct-splines.html`, [Online; accessed 05-June-2019].

[16] Arduino. *Arduino Due*. `https://store.arduino.cc/due`, [Online; accessed 05-June-2019].

[17] Trinamic Motion Control. *TMCS-28HardwareManual*. `https://www.trinamic.com/fileadmin/assets/Products/Encoder_Documents/TMCS-28_datasheet_Rev1.00.pdf`, [Online; accessed 05-June-2019].

[18] Honeywell. *3-Axis Digital Compass IC HMC5883L*. `https://github.com/TinyCircuits/TinyCircuits-TinyShield-Compass-ASD2613/raw/master/res/HMC5883L_3-Axis_Digital_Compass_IC_Datasheet.pdf`, [Online; accessed 05-June-2019].

[19] Honeywell. *1, 2 and 3 Axis Magnetic SensorsHMC1051/HMC1052/HMC1053*. `https://aerospace.honeywell.com/en/~/media/aerospace/files/datasheet/hmc_1051-1052-1053_data_sheet.pdf`, [Online; accessed 05-June-2019].

[20] Carsten Bielmeier. *Fahrradphysik*. Julius-Maximilians-Universität, Würzburg, 2012.

[21] Atmel. *SAM3X/SAM3A SeriesAtmel SMART ARM-based MCU DATASHEET*. `http://ww1.microchip.com/downloads/en/devicedoc/atmel-11057-32-bit-cortex-m3-microcontroller-sam3x-sam3a_datasheet.pdf`, [Online; accessed 05-June-2019].

[22] D. Ishak, N.A.A Manap, M.S. Ahmad and M.R. Arshad. *Electrically Actuated Thrusters for Autonomous Underwater Vehicle*. In: Advanced Motion Control, pp. 619 -624 IEEE, Nagaoka, Japan 2010.

[23] Skog, Hendeby, Gustafsson. *Magnetic Odometry - A Model-Based Approach Using A Sensor Array*. In: 21st International Conference on Information Fusion (FUSION), pp. 794-798, 2018.