Chair of Automation

Master's Thesis

Uncertainty Modelling in Sensor Systems

Serkan Ergun, BSc

November 2019

Abstract

This work presents the mathematical modelling of an integrated gradient field based Hall-angle sensor excited by a diametrically polarized disc magnet. It starts with the computation of the magnetic field of such a magnet based on the results presented in relevant publications. Coordinate transformations are then performed to simulate the assembly tolerances of the sensor and magnet. An abstraction of the real sensor has been designed and coded to simulate the data path of the sensor product. It has been shown that this simplified model delivers results consistent with more detailed and complex models. An increasing complexity of the model has little effects on further accuracy. The results of the data path are then processed using a Kalman Filter algorithm to compensate errors arising from noise as well as signal processing and transmission delays.

All models are implemented in m-code (MATLAB). These models enable the simulation of uncertainty propagation using a toolbox. This toolbox uses the approach presented in "Guide to the expression of Uncertainty in Measurement" (GUM). The results deployed by this toolbox have been validated against Monte Carlo simulations. The possibilities and limitations of this toolbox are shown for the magnetic field calculation, data path calculation and filtering using Kalman's algorithm.

The results of the simulation are consistent with experimental results conducted on uncalibrated test chips. Time series analysis has been performed on these results to remove systematical error contribution. After such thorough calibration, the angle error results reduce to values of as low as 0.05° (1 σ value), which is simply the remaining noise level.

This work enables concept engineers for such sensor products to perform fast and yet accurate uncertainty analysis. It allows statistical optimization at an early stage and therefore significantly reduces the time required for product development.

Zusammenfassung

In dieser Arbeit wird die mathematische Modellierung eines integrierten Hall-Winkelsensors auf Gradientenfeldbasis vorgestellt, der von einem diametral polarisierten Scheibenmagneten angeregt wird. Die Arbeit beginnt mit der Berechnung des Magnetfelds eines solchen Magneten auf der Grundlage der in einschlägigen Veröffentlichungen vorgestellten Ergebnisse. Anschließend werden Koordinatentransformationen durchgeführt, um die Montagetoleranzen von Sensor und Magnet zu simulieren. Eine Abstraktion des realen Sensors wurde entworfen und programmiert, um den Datenpfad des Sensorprodukts zu simulieren. Es wurde gezeigt, dass dieses vereinfachte Modell Ergebnisse liefert, die mit detaillierteren und komplexeren Modellen konsistent sind. Eine zunehmende Komplexität des Modells hat nur geringe Auswirkungen auf die weitere Genauigkeit. Die Ergebnisse des Datenpfads werden dann unter Verwendung eines Kalman-Filter-Algorithmus verarbeitet, um Fehler zu kompensieren, die sich aus dem Messrauschen und Signalverarbeitungs- sowie Übertragungsverzögerungen ergeben.

Alle Modelle sind in m-Code (MATLAB) implementiert. Dieses Modell ermöglicht die Simulation der Unsicherheitsausbreitung mithilfe einer Toolbox. In dieser Toolbox wird der in ”Guide to the expression of Uncertainty in Measurement” (GUM) beschriebene Ansatz verwendet. Die von dieser Toolbox bereitgestellten Ergebnisse wurden anhand von Monte-Carlo-Simulationen validiert. Die Möglichkeiten und Einschränkungen dieser Toolbox werden für die Magnetfeldberechnung, Datenpfadberechnung und Filterung unter Verwendung des Kalman-Algorithmus gezeigt. Die Ergebnisse der Simulation stimmen mit experimentellen Ergebnissen überein, die mit nicht kalibrierten Testchips durchgeführt wurden. An diesen Ergebnissen wurde eine Zeitreihenanalyse durchgeführt, um den systematischen Fehlerbeitrag zu beseitigen. Nach solch einer gründlichen Kalibrierung reduzieren sich die Winkelfehlerergebnisse auf Werte von 0,05° (1 σ -Wert). Dies entspricht dem verbleibenden Rauschpegel.

Diese Arbeit ermöglicht es Konzeptingenieuren solcher Sensorprodukte eine schnelle und dennoch genaue Unsicherheitsanalyse durchzuführen. Es ermöglicht eine frühzeitige statistische Optimierung und reduziert damit erheblich den Zeitaufwand für die Produktentwicklung.

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich diese Arbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt, und mich auch sonst keiner unerlaubten Hilfsmittel bedient habe.

Ich erkläre, dass ich die Richtlinien des Senats der Montanuniversität Leoben zu "Gute wissenschaftliche Praxis" gelesen, verstanden und befolgt habe.

Weiters erkläre ich, dass die elektronische und gedruckte Version der eingereichten wissenschaftlichen Abschlussarbeit formal und inhaltlich identisch sind.

Datum 28.10.2019

Unterschrift Verfasser/in
Serkan, Ergun

Dedication

I dedicate this thesis to my wonderful children, Jakob and Emma.

I have learned more from you, than I ever could give back.

Acknowledgements

My sincerest gratitudes belong to everyone, who made this Master's thesis possible. I especially want to thank my mentor, Dr. Wolfgang Granig, for his constant input and feedback during my internship , Prof. Dr. Paul O'Leary for his valuable input on scientific writing and time-series analysis and Prof. Dr. Hubert Zangl for his constant support an for answering all the little questions that I came across during the last months. I also want to thank my colleagues at Concept Engineering and the department of Sense & Control of Infineon Technologies Austria AG for offering me the opportunity to write this thesis and making me feel appreciated since day one.

A special thanks is also dedicated to Dr. Michael Ortner for allowing me to use his MagPyLib as a second source for my computations of magnetic fields before official public release.

Furthermore, I want to thank Prof. Dr. Anton Gfrerrer, head of the Chair of Computational Geometry at MU Leoben for his lectures on kinematics that have proven to be a huge help in developing the mathematical model for coordinate transformations.

A big thanks deserve Dr. Andreas Spaethe and Florian Floh for proof-reading this thesis.

A special thank you is dedicated to my wife Karoline, my children Emma and Jakob and my mother-in-law Sabine, for allowing me to spend weekends at my desk writing this thesis.

Finally, I want to thank my mother, Christa, my father Haydar (who passed away in late 2015) and my brothers, Selim, Ender and Hakan for their emotional support during all these years.

Contents

List of Figures

List of Tables

Units used in this thesis

$^\circ$	Degree
A	Ampere
$^\circ$C	Degree Celsius
g	Gram
h	Hour
Hz	Hertz
K	Kelvin
kHz	Kilohertz
km	Kilometre
LSB	Least Significant Bit
m	Meter
mm	Millimetre
ms	Millisecond
mT	Millitesla
mV	Millivolt
μs	Microsecond
ns	Nanosecond
%	Per cent
ppm	Parts per million
rad	Radian
s	Second
T	Tesla
V	Volt
Ω	Ohm

Nomenclature & Abbreviations

ADC	Analog Digital Converter
AE	Angle Error
AEC	Automotive Electronics Council
AG	Air Gap
AMR	Anistropic Magneto-Resistance
AUT	An Uncertainty Toolbox
CCDF	Complementary Cumulative Distribution Function
CDF	Cumulative Distribution Function
CORDIC	Coordinate Rotation Digital Computer
cpd	continous probability distribution
CS	Coordinate System
DoF	Degree of Freedom(s)
dpd	discrete probability distribution
EMI	ElectroMagnetic Interference
EV	Electric Vehicle
FFT	Fast Fourier Transform
IC	Integrated Circuit
GMR	Giant Magneto-Resistance
GUM	Guide to the expression of Uncertainty in Measurement
OEM	Original Equipment Manufacturer (see glossary)
OOP	Object Oriented Programming
PCB	Printed Circuit Board
pdf	probability density function
PPCC	Probability Plot Correlation Coefficient
ppf	percent point function
ROI	Region of Interest
RR	Reading Radius
RV	Random Variable
TMR	Tunnel Magneto-Resistance
unc()	uncertainty object within AUT
unc_t()	uncertainty object for Monte Carlo simulations within within AUT
UT	Unscented Transform
φ	true angle (= rotation angle, mechanical angle)
$\varphi\tilde{}$	measured angle
$\Delta\varphi$	angle error (=$\varphi\tilde{}$-φ)
μ	expected value
σ	standard deviation
σ^2	variance

Additionally, the following short hand notation is being introduced:

1. Matrices are denoted by capital bold letters: $\mathbf{A}$
2. Vectors are denoted by small bold letters: $\mathbf{x}$
3. Measurements are denoted by a superscripted tilde: $\mathbf{A}^{\sim}$, $\mathbf{x}^{\sim}$
4. Estimates are denoted by a ”hat” above the symbol: $\hat{\mathbf{A}}$, $\hat{\mathbf{x}}$
5. Predictions are denoted by an apostrophe: $\mathbf{A}'$, $\mathbf{x}'$
6. And finally; Predictions of estimates are denoted by a combination of the denotations above: $\hat{\mathbf{A}}'$,$\hat{\mathbf{x}}'$

Chapter 1

Introduction

1.1 Introduction

A world without sensors would be unimaginable today. Sensors are used in almost all areas of life. There are a variety of sensors, from simple liquid thermometers to complex acceleration sensors. These two examples also lead to the first distinction of sensors. There are sensors which directly indicate the measurement result to the user in form of a observable physical quantity (temperature change leads to an expansion of the liquid) or those sensors which convert an input physical quantity to an electrical output (voltage, current, electrical resistance, charge, capacity, ...) produce. This work will focus on the latter of which.

One of the core tasks is to design these sensors. This is not a trivial task which takes many months or years of development, as a sensor is no longer exclusively used to detect and convert measurable variables but, in the case of integrated sensors, encloses the supply of the sensitive elements and digitization and performing evaluation algorithm to directly calculate the quantity of interest. In most cases an error compensation is performed onboard too. One of the core customers of such sensor products is the automotive industry.

This Master's thesis will put close attention to angle sensors. They are used in a wide range of applications. Some of the most prominent applications for angle sensors include steering angle detection, commutating electric motors and detection of throttle and valve positions. [7]

Increasing requirements in functional safety drives sensor system suppliers of mass vehicle manufacturers to develop accurate, reliable and robust sensor products. Magnetic field based sensors are very well known to fulfil these requirements. Unfortunately the continuous electrification of the propulsion causes electromagnetic interference to components sensitive to magnetic fields. Gradient field based sensing principles can suppress these effects and have gained large popularity in the last decade.

Concept engineers of such sensor products may want to have an adequate mathematical model at an early stage of development to reduce unnecessary overhead in simulations and experiments to make assumptions about the accuracy and reliability of their product. Delivering such results at this early stage forces engineers to consider the concept of uncertainty propagation during this phase.

Therefore this work presents a model of such a gradient field based sensor with Hall plates using technological data from existing products. The uncertainty is propagated through the signal processing path using an existing toolbox developed by the Alpen Adria University of Klagenfurt [1]. The results of this model are validated against experimental results. Evident conclusions summarize the possibilities and restrictions of the mathematical model and the uncertainty propagation tools. An outlook for future analysis such as methods for statistical optimization of the sensor product based on these conclusions is presented.

1.2 Target application

A roadmap of this Master's thesis is shown in Figure 1.1. This project aims to model an entire "datapath" of a gradient field (which is stray field robust if some prerequisites are met) magnetic angle sensor (section 1.4.2) using MATLAB ®. This datapath starts with the computation of 3D- field solution created by a diametrically polarized permanent disc magnet. It moves on to an abstracted mathematical model of an Integrated Hall sensor, which includes signal acquisition, digitalization and offset and amplitude corrections aiming to return an estimate of the angular position of a shaft. In the final step, a Kalman Filter is applied to allow predictions of future state values.

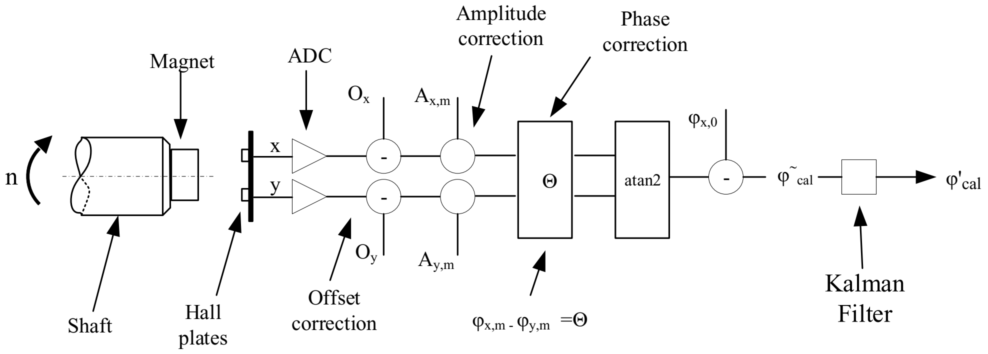

Figure 1.1: Roadmap of this Master's thesis. The signal processing path consists of the calculation of the magnetic field, modelling Hall plates, Analog-Digital Converters (ADC), signal corrections (amplitude and phase), angle calculation via an arctangent function and future state prediction with Kalman Filter.

Additionally, the uncertainty propagation was considered for this model using a MATLAB based toolbox developed by the University of Klagenfurt for evaluating uncertainties. This toolbox was tested for its applicability for stray field robust gradient based angle sensors and validated against Monte Carlo simulations and experimental results of test chips.

This Master's thesis aims to build an accurate model of such sensor devices, allowing Concept Engineers to conduct fast computations of the performance of stray field robust magnetic sensors at an early stage and providing the basis for the adaption to other sensor devices and reducing significant time in sensor development.

1.3 Automotive sensor requirements

The following listing contains some general requirements of automotive sensors. Sensor system manufactures need to fulfil these requirements when selling their products to the automotive industry. The information in this list was gathered using both hard-copy books and online sources.

1. Temperature range: $-40°$C up to $+125°$C ($+150°$C for sensors in the engine compartment, or even up to $+180°$C closer to the engine) [2–7]. Wheel-speed sensors placed close to the brakes require operation temperatures of up to $+190°$C [8].
2. Maximum failure rate varying between less than 1 ppm [3] and 5 ppm [4]
3. Failure probabilities $< 10^{-9}$ $failures/h$ for safety critical sensors [7, 9]
4. Minimal service life of 15 years or 200.000 km mileage [3]
5. Durability against water, steam, oil, fuel, salt, dust, vibrations (in engine compartment) and even ozone [3, 7]
6. Redundancy of safety critical sensors [10]

The Automotive Electronics Council (AEC) is the standardization body which aims to establish standards for electronic components with high quality and reliability in vehicles. Their developed standards are publicly available from the AEC website [11]. The council is made up by leading companies of automotive OEM's and leading semi-conductor companies. Sensor products for the automotive industry are tested in accordance with AEC Q-100 [12].

1.4 Magnetic field sensors and their application in the automotive sector

Magnetic field sensors are probably one of the oldest sensors invented by mankind. Earliest documented use of compasses dates back to the Ancient Greek Empire around 500 BC [13]. The effect of magnetism was already known but could not be quantified at that time. It took another $2,300$ years until the first real magnetometer was built by Gauss in 1832. His rather simple device was made up by a permanent magnet and was suspended in midair by a fiber [14]. This magnetometer was able to measure the absolute magnetic intensity.

But not only humans have learned to use the effects of the terrestrial magnetic field. Recent studies show that migratory birds can sense the terrestrial magnetic field with their eyes to navigate during their travel periods [15].

State of the art magnetic field angular sensors are quite commonly used in automotive applications suffering from harsh environmental impact such as the engine compartment.

Magnetic field sensors do not suffer from wear and are quite robust against most kinds of contamination [7, 16, 17]. Application fields range from sensing small angles from single digit degrees for torque measurements via angular excursions over sensors used for headlight levelling up to multi-turn angular detection sensors for steering wheels. [7, 18]

Today two different physical principles are used to detect magnetic fields. The probably best known is the Hall effect, which has been discovered at the end of the 19^{th} century and magneto-resistive effects such as Anisotropic Magneto Resistors (AMR), Giant Magneto Resistors (GMR) and most recently Tunnel Magneto Resistors (TMR). The Hall effect relies on detecting the value of the magnetic field, whereas magneto-resistive sensors detect the direction (changes) of the field and do not measure the field values itself.
Most of the magnetic field sensors manufactured today still use the Hall effect [19]. The reason for this is despite their much lower sensitivity against GMR's and TMR's that they are not only easy and cheap to built but have also high linearity up to high magnetic fields without hysteresis effects. The properties of Hall sensors are well known and documented in various books and publications.

1.4.1 Sensing principles for magnetic field based angle sensors

Four different sensing principles are commonly used for magnetic field based angle sensors. In all four options a permanent magnet is attached to a rotating shaft. Their application is typically determined by the mechanical conditions. Whenever a shaft end is available an end-of-shaft or in-shaft solution will be chosen. Whenever the shaft end is not accessible for mounting magnets, a through-shaft or out-of shaft solution is the best way to go. In-shaft sensors have the best performance in general [20] but are also very expensive in manufacturing. End-of-shaft solutions are the cheapest as only cheap and simple magnets are needed and the setup is quite easy (no drilling required). Through-shaft and out-of-shaft solution need larger and thus more expensive magnets (depending on the shaft diameter). All four sensing principles are illustrated in Figure 1.2.

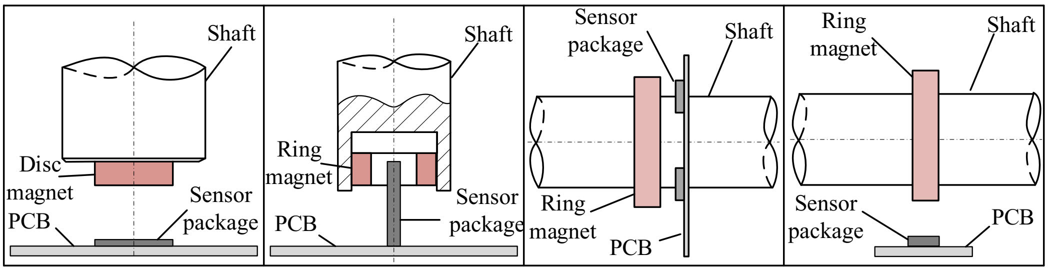

Figure 1.2: Most common sensing principles for magnetic angle sensors. From Left to Right: End-of-shaft,in-shaft, through-shaft and out-of-shaft sensing. PCB: Printed Circuit Board.

The sensor analysed in this Master's thesis is used in an end-of shaft sensing application.

1.4.2 Stray field robust magnetic angle sensing

The recent paradigm change in the automotive industry moving from combustion engines to the electrification of propulsion (electric vehicles (EV) with batteries as well as hydrogen fuelled vehicles and hybrids) caused new challenges for magnetic sensors as high power cables are causing Electromagnetic Interference (EMI) to magneto-sensitive devices. One possible solution to overcome this challenge are gradiometric based magnetic sensors for end-of-shaft solutions, which have been introduced by [21], with early ideas dating back to 1994 [22]. The basic idea is it to place four Hall elements on a circle with *reading radius RR*. Two opposite Hall elements built up one gradiometer. This setup detects the gradients of the axial magnetic field components with respect to the x- and y-axis.

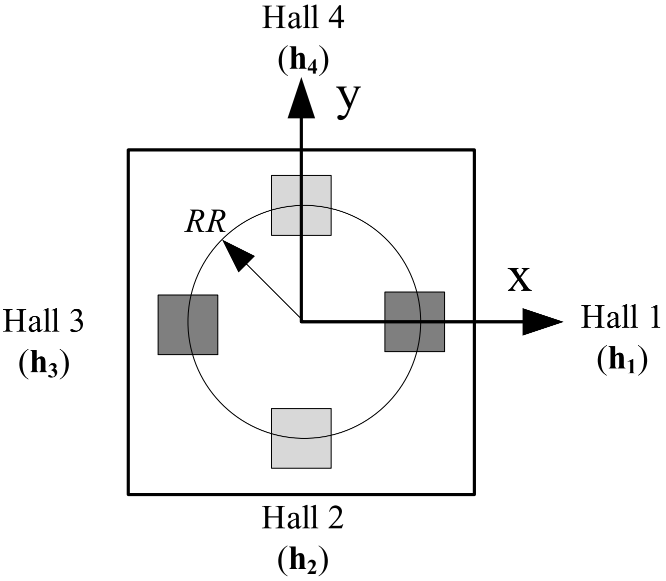

Figure 1.3: Sensing principle of gradient field based Hall effect sensor. Four Hall plates (Hall 1 to 4) are placed on a circle around the center. A gradiometer is made up by two opposite Hall plates [63].

This sensing principle is able to cancel out homogeneous disturbances if some precautions are met (such as non-ferrous shafts). The system cannot cancel out inhomogeneous disturbances, but since their effects are small, they are neglectable.

The following example explains how homogeneous disturbances are cancelled out by the sensor system. The rotating magnet ideally induces a sinusoidal magnetic signal at each Hall plate ($B_{H\,1}$ to $B_{H\,4}$). Each Hall plate has an equal amplitude of B_{amp} and a phase shift of 90° ($\frac{\pi}{2}$ in radians) if mounted equiangular on the die. An ideal homogeneous stray field B_{off} affects all Hall plates equally. With this the magnetic field at each Hall plate can be calculated by:

$$
\begin{aligned}
B_{H\,1} &= B_{amp} \quad sin(\varphi+\pi) \quad +B_{off}, &(1.1)\\
B_{H\,2} &= B_{amp} \quad sin(\varphi+\tfrac{\pi}{2}) \quad +B_{off}, &(1.2)\\
B_{H\,3} &= B_{amp} \quad sin(\varphi) \quad +B_{off}, &(1.3)\\
B_{H\,4} &= B_{amp} \quad sin(\varphi-\tfrac{\pi}{2}) \quad +B_{off}. &(1.4)
\end{aligned}
$$

The difference of Hall 1 and Hall 3 makes up the ”x” signal x_{Sig} of the sensor and the difference of Hall 2 and Hall 4 makes up the ”y” signal y_{Sig}:

$$x_{Sig} = B_{H\,1} - B_{H\,3} \tag{1.5}$$

and

$$y_{Sig} = B_{H\,2} - B_{H\,4}. \tag{1.6}$$

In this manner the homogeneous offset B_{off} (which may vary over time but has to be invariant of the location) is cancelled out as it affects both Hall plates equally, this is shown by:

$$x_{Sig} = B_{amp}\,sin(\varphi+\pi) + B_{off} - (B_{amp}\,sin(\varphi) + B_{off}), \tag{1.7}$$

$$\tag{1.8}$$

when applying basic trigonometry,

$$sin(\varphi) = -sin(\varphi+\pi), \tag{1.9}$$

x_{Sig} simplifies to:

$$x_{Sig} = -2\,B_{amp}\,sin(\varphi). \tag{1.10}$$

y_{Sig} is calculated the same way by,

$$y_{Sig} = 2\,B_{amp}\,sin(\varphi+\frac{\pi}{2}). \tag{1.11}$$

The measured angle $\tilde{\varphi}$ can then be calculated with the ”arctan2” function given by,

$$\tilde{\varphi} = \text{arctan2}(x_{Sig}, y_{Sig}). \tag{1.12}$$

Compared to the regular ”arctan”, ”arctan2” can resolve angles in [0,360)°, whereas ”arctan” is only unique within (-90,+90)°.

The angle error $\Delta\varphi$ is then the difference of $\tilde{\varphi}$ and the true angle value φ given by,

$$\Delta\varphi = \tilde{\varphi} - \varphi. \tag{1.13}$$

As this gradiometric sensing principle needs information about the values of the magnetic field, out of magnetic angle sensing devices only Hall effect sensors are suitable for homogeneous stray field compensation.

1.5 Definition of accuracy errors

Equation (1.13) simply defined an angle error as the difference of some measured angle φ˜and the true angle φ which varies over the course of time. Application engineers are not interested in error curves within a certain angle domain (e.g. [0,360]°) but rather in an ”overall” error in this angle domain. Most commonly, two different definitions are used according to [23]: The Maximum Error ME-error being the absolute value of the largest peak in the error curve and the Average Error AE-error [24] being the mean of the highest and the lowest peak in the error curve. Both accuracy definitions are shown in Figure 1.4.

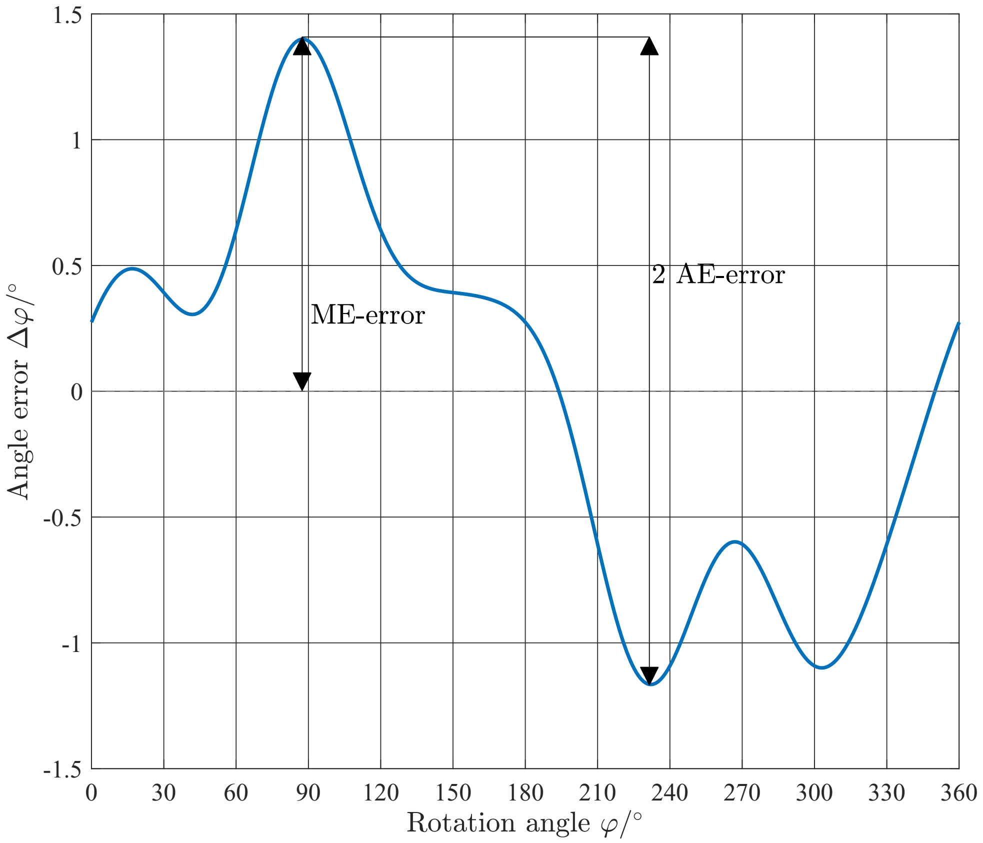

Figure 1.4: Most common quantities to describe accuracy errors by an exemplary angle error curve: Maximum Error ME-error = $max(|\Delta\varphi_{min}|,|\Delta\varphi_{max}|)$ and Average Error AE-error = $\frac{\Delta\varphi_{max}-\Delta\varphi_{min}}{2}$

AE-error is more sophisticated than ME-error as systematic (bias) angle errors are cancelled out for AE- but not for ME-errors. These systematic angle errors are cancelled out anyway during the calibration process of the sensor. In addition, AE-error is smaller than ME-error in general implying a higher accuracy of the sensor product.

Chapter 2

Background & State of the Art

2.1 The Hall effect and Hall effect devices

The Hall effect was discovered in 1879 by Edwin Hall, an American physicist working at John Hopkins University at that time [25].

Imagine a strip of metal (or a semi-conductor) with thickness d, which carries an electric current (which will be later on defined as bias current) I_B along its length. When a magnetic field (with flux density B) is applied simultaneously, electrons in this strip are deflected towards an edge of the strip. This produces a voltage gradient across the width of the strip. This voltage is the so-called Hall voltage V_H given by,

$$V_H = A_H \frac{I_B\, B}{d}, \tag{2.1}$$

where A_H is the Hall constant, which can be calculated by,

$$A_H = \frac{1}{n\, q}. \tag{2.2}$$

where n is the charge carrier density and q is the electric charge of a single charge carrier.

A schematic Hall plate is shown in Figure 2.1. If the direction of the magnetic field is reversed, the polarity of the induced voltage will also reverse [5].

The main advantage of this effect is it that this effect does not need a changing magnetic field to induce this voltage (compared to inductive sensors), so Hall sensors can also detect static magnetic fields [5].

The main disadvantages of Hall sensors are their relatively low sensitivity and a lower offset stability (due to the piezo effect) compared to magneto resistive sensors [19]. Under most conditions, the Hall Effect is not measurable in metals. Semi-conductors are much better due their lower charge carrier density [5].

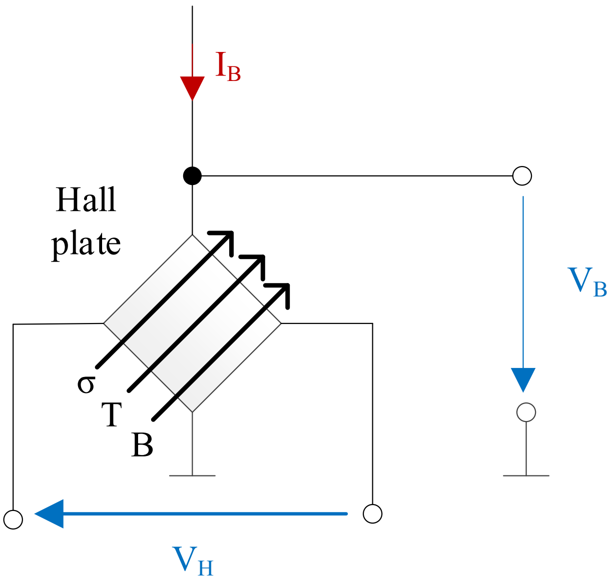

Figure 2.1: Schematic overview of a Hall plate. The Hall voltage V_H is affected by the bias circuit (bias current I_B and bias voltage U_B) as well as mechanical stress σ, temperature T and the magnitude of the magnetic field B.

Unfortunately, Hall sensors are not only sensitive to the magnetic field. They also detect changes of temperature T and mechanical stress σ. In addition also the ambient humidity has an effect on the sensitivity of the sensor as the plastic package swells in highly humid environment and effects the mechanical stress randomly in nature [26].

The bias voltage V_B plays quite an important role in Hall sensors. An angular Hall sensor needs to sense a changing magnetic field at different temperatures. With an increasing temperature the sensitivity of the sensor decreases. On the other hand, an increasing temperature will cause a higher bias current I_B. This increasing bias current increases the sensitivity of the Hall plate therefore cancelling out the negative effect to a certain stage and thus limiting the temperature effect.

2.1.1 Main characteristics of Hall sensors

According to [5] the main characteristics needed to model a Hall effect transducer are:

1. Sensitivity
2. Temperature coefficient of sensitivity
3. Ohmic offset
4. Temperature coefficient of ohmic offset

5. Linearity
6. Input and output resistances
7. Temperature coefficient of resistances
8. Electrical output noise

The individual characteristics are explained in detail below:

Sensitivity

The sensitivity (also referred to as gain) describes in general the ratio of some of output with respect to its input. A high sensitivity means that the output source is higher compared to a low sensitivity when the input value is the same for both sensors. A high sensitivity is generally a good thing. The sensitivity of a Hall sensor depends on the amount of current used to bias the sensor. There are two ways to describe the sensitivity of a Hall sensor: [5]

- Volts per unit field, per unit of bias current
- Volts per unit field, per unit of bias voltage

The sensitivity of silicon (Si) based Hall sensors is around 1 mV/mT for a 1 mA current. Indium antimonide (InSb) reaches a typical sensitivity of around 5 mV/mT and Indium arsenide (InAs) has a typical sensitivity of around 2 mV/mT [5]. Although InAs has a lower sensitivity than InSb, InAs has a lower temperature dependence and a wider working range (exceeding the temperature range from -40°C up to 150°C, which is needed for automotive applications) [19].

Temperature Coefficient of Sensitivity

In general, Hall effect sensors do have decent constant sensitivity (if they are powered by a constant current source), but the sensitivity does vary a little over temperature. For some cases, an assumed constant sensitivity is enough, but as soon as a higher stability is desired, the non constant sensitivity needs to be considered. [5]

Ohmic offset

The ohmic offset (or offset voltage) is a small voltage that can be measured from a Hall sensor, even when it is not excited by a magnetic field. The ohmic offset limits the ability to detect small magnetic field of steady state. There are multiple effects, which cause this offset voltage. Sources for this offset come from inhomogeneities of the sensor materials (e.g. doping density or surface conditions), alignment errors of the sensor contacts and the piezo-resistive effect, due to nature of semiconductors. [5] [6]

Temperature coefficient of ohmic offset

The offset voltage of a Hall sensor is not constant over temperature. In addition this effect tends to be of random nature making it unpredictable in general. This drift can vary across ”identical” devices. [5]

Linearity

Hall effect sensors are passive devices which cannot deliver unlimited output voltage. The physical limit is the input (or bias) voltage. In practice the limit of linearity is much lower due to saturation effects. When measuring small voltages compared to the bias voltage, a Hall sensor is very likely to be linear, whereas the linearity is lost as soon as the output voltage increases. When measuring very large magnetic fields, one may use a Hall sensor with a low sensitivity, because these sensors do not saturate so easily. [5]

Input and output resistances

Both the input and the output resistances have an influence on the shape of the bias circuitry and the front-end amplifier used to recover the transducer signal. The input resistance affects the design of the bias circuitry, while the output resistance affects the design of the amplifier used to detect the Hall voltage. [5]

Temperature coefficient of resistances

The temperature coefficients of the input and output resistance are very close to each other at least. In general cases they may be identical. According to [5] practical transducers the temperature coefficient of resistance can be quite high, often as much as $0.3\%/°C$. For typical automotive purposes the resistances can vary by up to 30%. [5]

Electrical output noise

Hall sensors show electrical noise at their outputs. Naturally, there can be many sources for a noisy output signal. The most common and fundamental noise source is the thermal noise caused by moving electrons, which is better known as *Johnson noise, Nyquist noise* or *Johnson–Nyquist noise*. The *Johnson noise* can be calculated by,

$$V_n = \sqrt{4\,k\,T\,R\,B_w}, \tag{2.3}$$

where ,

- k is the Boltzmann's constant (approximately $1.38\ 10^{-23}$ J K^{-1}),
- T is the absolute temperature in K,
- R is the resistance in Ω,
- B_w is the bandwith in Hz.

Another noise source can be $1/f$ noise (also known as Flicker noise), which can be found in many physical systems. $1/f$ noise can have various different types of underlying mechanisms. The amount of noise per unit bandwidth decreases with increasing frequencies explaining its naming. Different literatures [5, 6] either prioritize the Johnson or $1/f$ noise respectively.

Additional error sources

The error sources shown above are not the only ones that can affect the accuracy of Hall sensors. Some other error sources are assembly tolerances of the sensor and magnet (which will be covered in section 3.1 of this thesis) or imperfections and demagnetization effects of the permanent magnet the latter of which causing inevitable life-time drifts [20, 27] which cannot be compensated beforehand. But also external sources of deviation such has Electromagnetic Interferences (EMI) or stray fields of both homogeneous and inhomogeneous nature cause errors. All these non-compensable error sources perturb the measurement signal with uncertainties.

Chopper Stabilization

One possibility to reduce offset by removing errors caused by thermal and mechanical stresses is chopper stabilization. It separates these stresses from the magnetic field based signal in the frequency domain using a modulation-demodulation process [28]. Unfortunately, this causes a delay and adds electrical noise to the signal [29]. A newer and more advanced alternative to chopping is the spinning Hall technique. Such a sensor is given in [30]. New magnetic sensors even combine the spinning-current Hall probe and a chopped $3^r d$-order continuous-time $\Delta\Sigma$-converter to manage the balancing act of both low noise and good stability [31].

Integrated Hall Sensors

Integrated Hall sensors use more than a single Hall element inside an Integrated Circuit (IC). In most cases two or four (sometimes even more) Hall elements are placed within a common IC. Their positioning is not arbitrary as they should all experience the same packaging stress. Such a setup can eliminate the signal due to packaging stress [32]. To fulfil increasing requirements regarding Functional Safety of road vehicles (ISO 26262) [10] so called dual-channel integrated sensors are manufactured today. Two redundant measurement paths are placed on the same die. Each path operates independent of the other, but detect the same magnetic quantity [33, 34].

2.2 Signal processing

After the Hall effect device itself is now properly modelled it is time to look at the actual signal processing next. The aim of the signal processing path is it to transfer some analogue Hall output voltage V_H to some measured angle value $\varphi\tilde{}$. Recalling section 1.4.2 we already know that there are four Hall plates with four individual Hall voltages:

$$\boldsymbol{V_H} = \begin{bmatrix} V_{H\,1} \\ V_{H\,2} \\ V_{H\,3} \\ V_{H\,4} \end{bmatrix} . \tag{2.4}$$

The first step is it to calculate analog signal path voltages V_x and V_y of the respective voltage paths and digitalize them using a separate Analog Digital Converter (ADC) for each path. A schematic figure, which illustrates this, is shown in Figure 2.2.

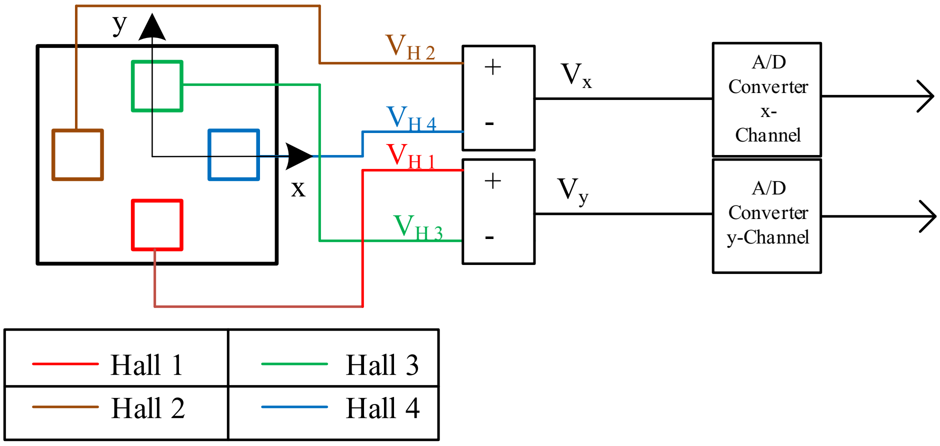

Figure 2.2: Schematic overview of the analog part of the signal processing path. Channel voltages V_x, V_y are calculated by using the differential voltage of two opposite Hall plates. These channel voltages are then digitalized.

2.2.1 Analog Digital Converters (ADC)

An Analog Digital Converter is a device that converts an analog time continous signal such as a voltage to a discrete digital signal (which is typically a two's complement binary number). The main parameters of an ADC are its resolution (in bits) and its maximum sample rate. Indicators for ADC performance are its bandwidth (which is mainly affected by the sample rate) and the signal-to-noise ratio (which is affected by its resolution, errors in accuracy and linearity as well as jitter and aliasing effects). For an ideal ADC the resolution Q in V/LSB (least significant bit) can be calculated quite easily by, [35]

$$Q = \frac{V_{FSR}}{2^N}, \tag{2.5}$$

where V_{FSR} is the full scale range of the voltage signal which is given by,

$$V_{FSR} = V_{max} - V_{min}, \tag{2.6}$$

where $[V_{min}, V_{max}]$ are the highest and lowest expected voltage peak in the signal.
N is the resolution of the ADC in bits. The limiting factor in resolution is usually the signal-to-noise ratio.

Sample-and-Hold

The conversion from a given voltage to a binary value cannot be performed instantly. That is why the voltage signal needs to be kept constant during the conversion. Most simply a circuit consisting of a simple capacitor controlled by an electronic switch can be used to store the signal. Such a circuit is called Sample-and-Hold circuit and is part of most ADC IC's.

Error sources

Like other electronic devices ADC's suffer from imperfections which cause errors to the output signal of the ADC. Some of the most prominent error sources of ADC's are listed below:

Quantization error

Converting analog values into digital ones is always perturbed with quantization error e_Q caused by the limiting resolution of the ADC (see Equation (2.5)). The quantization error cannot exceed the gap of one bit (but can be reduced to ±0.5 bit if the ideal transfer function is shifted to the center of each bit range [36]), or else the ADC would be converting the value to a neighbouring bit. The quantization error is defined as:

$$e_Q = V_d - V_a, \tag{2.7}$$

where V_d is the quantized (digital) voltage and V_a its analog input value. The quantization error is signal dependant and non-linear in general.

Aliasing

Aliasing (or discretization error) typically has two main sources. Either the fundamental *Nyquist–Shannon sampling theorem* was not adhered (The sampling rate has to be at least twice as high as the highest frequency f_{max} in the spectrum) or noise in the signal has a frequency which is higher than half of the sampling rate.
Aliasing can be avoided if the analog signal passes through a low pass filter before digitalization. After digitalization, aliasing cannot be avoided.

Offset Error

Like Hall effect devices, ADC's too are perturbed with offset error. This offset error is also defined as some output of the device when no input signal is present and the theoretical output signal should be zero. Such an offset error can be both positive or negative and may cause the ADC to saturate before the full scale range is reached [37]. Gain Error Consider the offset to be compensated. A gain error is a cause of a deviating slope against the ideal transfer function (while still being linear though). A positive gain error indicates a higher slope, whereas a negative gain error displays as a weaker slope of the transfer function.

The combination of the offset (e_O) and gain error (e_G) is called **Full-scale error** (e_{FS}). The Full-scale error is given by,

$$e_{FS} = e_O + e_G. \tag{2.8}$$

Non-Linearities

Unavoidable imperfections cause non-linearity errors which deviates the shape of the transfer function from the otherwise ideal linear transfer function. Deviations of the ideal step width are called Differential Non-Linearities (DNL) [38, 39]. The cumulation of these errors is defined as the Integral Non-Linearity (INL), which is the highest deviation of the real transfer function versus the ideal one [39, 40].

The effects of non-linearities cannot be removed by calibration [39].

2.2.2 Amplitude-, phase- and offset correction

Consider the signal processing path as given in Figure 2.2. Each ADC produces an digital output (x_{Sig}, y_{Sig}). Assume that the signals are ideal with the exceptions of offset-, gain errors and phase difference (also known as orthogonality error) being present. Such signals for both ADC's for a full rotation in $[0, 360]^\circ$ are shown in Figure 2.3.
Each signal (x-Signal x_{Sig} and y-Signal y_{Sig}) can be described with the following function given by,

$$x_{Sig} = A_x\, cos(\varphi) + O_x, \tag{2.9}$$

$$y_{Sig} = A_y\, cos(\varphi + \varphi_{ortho} + 90^\circ) + O_y, \tag{2.10}$$

where A_x and A_y denote the amplitude of each signal, φ_{ortho} denotes the phase shift between the two signals and O_x and O_y denote the offset of each signal.

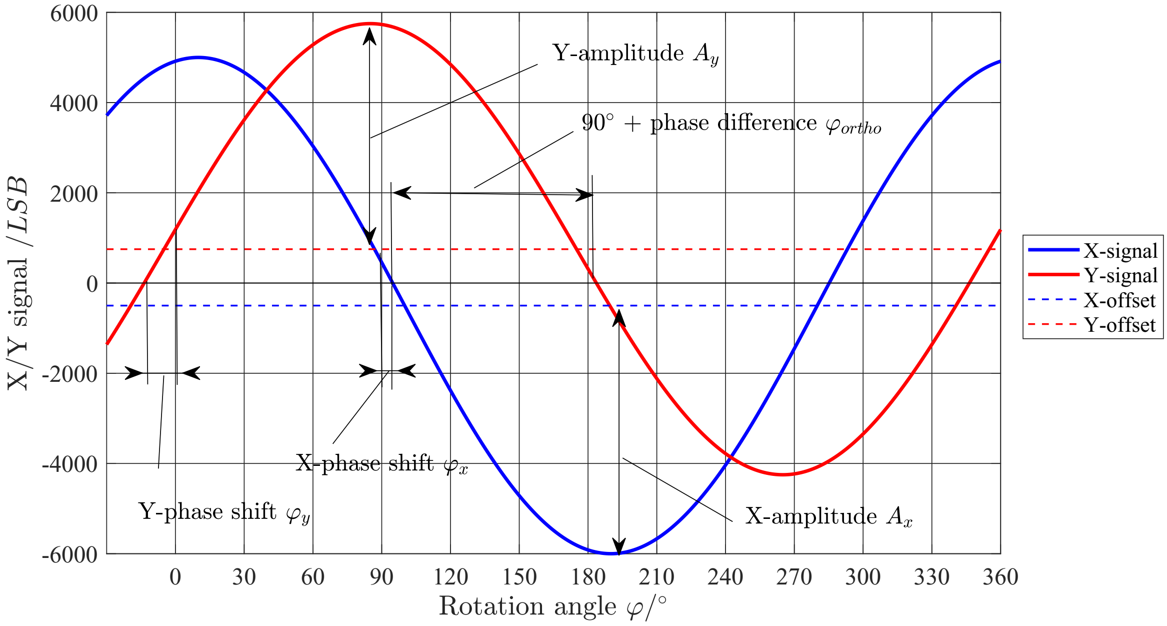

Figure 2.3: Definition of amplitude-, phase difference- and offset errors between the x-channel (blue) and y-channel (red). ADC's with sufficient resolution (the quantization error shall be neglected) are assumed.

This X-offset and Y-offset is not to be confused with the offset presented in Section 1.4.2. X- and Y- offsets here are caused by imperfections of the Hall device and the ADC, whereas the offset in Section 1.4.2 is merely caused by a homogenous stray field. This stray field is already cancelled out due to the gradiometric sensing principle presented in Section 1.4.2.

The calibration process explained here is based on the end-of-line calibrations documented in [41–43]. In this calibration process a full 360° rotation is recorded twice (once in clockwise (cw) and once in counter-clockwise (ccw) direction). Such an end-of-line calibration does not compensate long term drift effects [42]. Such effects are not covered in the scope of this thesis.

Determining calibration parameters

Amplitudes and offsets of both channels A_x, A_y,O_x and O_y can be calculated quite simply using only the peak values (X_{max}, X_{min}, Y_{max} and Y_{min}) of each channel. Their calculation is given by:

$$A_x = \frac{X_{max}-X_{min}}{2}, \tag{2.11}$$

$$A_y = \frac{Y_{max}-Y_{min}}{2}, \tag{2.12}$$

$$O_x = \frac{X_{max}+X_{min}}{2}, \tag{2.13}$$

$$O_y = \frac{Y_{max}+Y_{min}}{2}. \tag{2.14}$$

The phase difference φ_{ortho} (also known as orthogonality error [42]) can be calculated by determining the zero crossings of each signal:

$$\varphi_{ortho} = \varphi_{x,0} - \varphi_{y,0} - 90^\circ. \tag{2.15}$$

Alternatively, a Discrete Fourier Transformation (DFT) can be used for a more exact approach. The method is thoroughly explained in Section 5.1.2 of [42] and will not be cited here. For more informations to this approach refer to [42].

After these parameters are computed for both clockwise (cw) and counter-clockwise direction (ccw), the mean of the amplitudes $A_{x,m}$,$A_{y,m}$, offsets $O_{x,m}$, $O_{y,m}$ and phase difference $\varphi_{ortho,m}$ between both channels of both runs are calculated to obtain the symmetrical values:

$$A_{x,m} = \frac{A_{x,cw}+A_{x,ccw}}{2}, \tag{2.16}$$

$$A_{y,m} = \frac{A_{y,cw}+A_{y,ccw}}{2}, \tag{2.17}$$

$$O_{x,m} = \frac{O_{x,cw}+O_{x,ccw}}{2}, \tag{2.18}$$

$$O_{y,m} = \frac{O_{y,cw}+O_{y,ccw}}{2}, \tag{2.19}$$

$$\varphi_{ortho,m} = \frac{\varphi_{ortho,cw}+\varphi_{ortho,ccw}}{2}. \tag{2.20}$$

Most sensor devices also allow to compensate temperature dependant offsets. Sensor designers usually implement linear compensation (evaluating the offset at two temperatures) or quadratic temperature compensation (evaluating the offset at three different temperatures). The concept of linear temperature dependant offset compensation is illustrated in Figure 2.4.

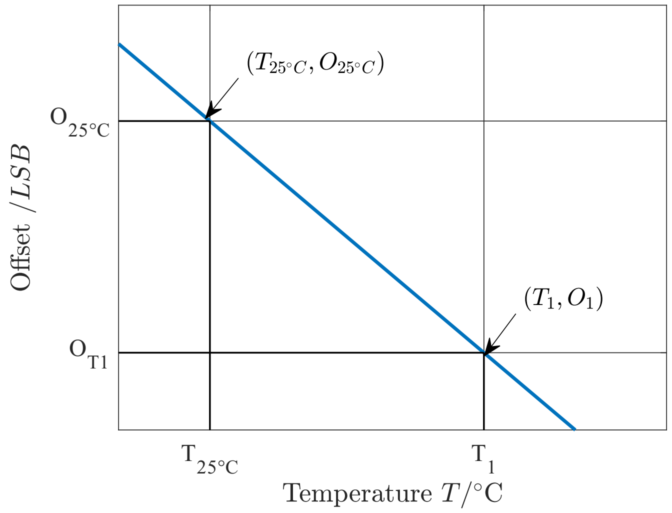

Figure 2.4: How compensation of temperature dependant offset is performed. In most cases it is sufficient to fit a straight line between two temperatures with known offsets. Usually room temperature (25°C) is chosen as one of the two points. This is not mandatory, though.

The calibration temperatures are chosen carefully in order two minimize this error for the full temperature range. In most cases room temperature is used as one of these two temperatures.
In order to compute the linear compensation two offsets (for both channels) at two different temperatures need to be known. The slope of the descending calibration line can be calculated by,

$$k_T = \frac{O_1 - O_{25°C}}{T_1 - T_{25°C}}, \tag{2.21}$$

for both channels ($k_{T\,x}$ and $k_{T\,y}$). Temperatures need to entered as digital values in the equation above. Offsets O_1 and $O_{25°C}$ are calculated for both channels by Equations (2.18-2.19). The calibration parameters are now fully determined and the calibrated measured angle $\varphi\tilde{}_{cal}$ can now be computed.

Angle Calibration

The actual calibration process, which is implemented in the microcontroller and calculated for each discrete signal value of the integrated sensor is shown in Figure 2.5. This multi-step process is explained below.

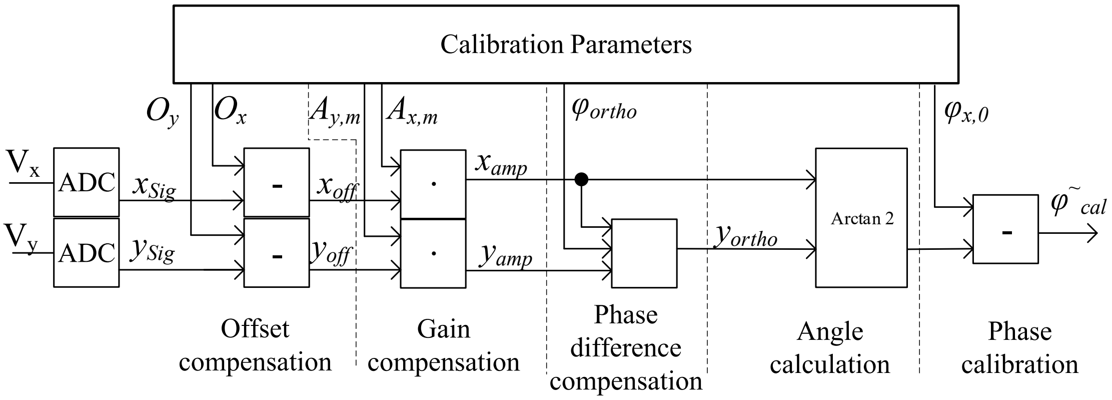

Figure 2.5: Schematic overview of the calibration process. Calibration consists of offset-, gain- and phase difference (orthogonality error) compensation. The measured angle is calculated after the compensations are performed. Typically a phase calibration is computed afterwards, if needed. The phase difference shift compensation covers the phase shift between the two signals. The final phase calibration calibrates the static offset of the angular output.

In the first step the offset values for both channels (O_x, O_y) are calculated using the temperature coefficients $k_{T\,x}$, $k_{T\,y}$ and the temperature (in digits) for both channels:

$$O_x = O_{x25^\circ C} + k_{T\,x}(T - T_{25^\circ}), \tag{2.22}$$

$$O_y = O_{y25^\circ C} + k_{T\,y}(T - T_{25^\circ}). \tag{2.23}$$

These offset values are then subtracted from the ADC output yielding a offset compensated signal given by,

$$x_{off} = x_{Sig} - O_x, \tag{2.24}$$

$$y_{off} = y_{Sig} - O_y. \tag{2.25}$$

In the next step, the amplitude of each signal is normalized to unit length. The gain compensation is performed by:

$$x_{amp} = \frac{x_{off}}{A_{x,m}}, \tag{2.26}$$

$$y_{amp} = \frac{y_{off}}{A_{y,m}}. \tag{2.27}$$

The error arising from the phase difference between the x and y signal is still present, one channel (mainly the y-channel) needs to be calibrated as given by,

$$y_{ortho} = \frac{y_{amp} - x_{amp}\,sin(-\varphi_{ortho,m})}{cos(-\varphi_{ortho,m})}. \tag{2.28}$$

Finally the $arctan2$ function can be used to calculate an unique angle value in $\varphi = [0,360)^\circ$. The phase shift of the x-Signal $\varphi_{x,0}$ needs to be subtracted from this angle and the final calibrated angle $\varphi\tilde{}_{cal}$ is obtained by:

$$\varphi\tilde{}_{cal} = arctan2\left(\frac{y_{ortho}}{x_{amp}}\right) - \varphi_{x,0}. \tag{2.29}$$

2.3 Kalman filtering

2.3.1 Need for prediction of future states

One might see the world and especially time as continuous system (Touching the field quantum mechanics is clearly not intended in this thesis), but when working with sensors one knows that he needs to switch to the discrete world owing it to the quantization effects when digitalizing physical units to bits and bytes. Of course one can choose small increments, but nonetheless these are discrete values. But when increasing the sample size, the amount of storage needed, increases of course. But also the components for digitalizing and computing are not ideal and need some physical time to conduct their computations (compare section 2.2.1).

Real Hall sensors typically have sampling rates between 5 Hz and 40 kHz [44–46]. This corresponds to sample intervals of 0.2 s to 25 μs. The computation time needed to convert the Hall sensor output to a digitalized angle value lies between 10 and 100 μs. This creates a latency between the signal acquisition and the further processing of the result. Such a latency causes a delayed sensor output producing systematic errors. To name an example, the efficiency of electrically commutated motors (ECM) (also known as brushless DC motors (BLDC)) is is strongly affected by these delays. Such safety critical sensors need evaluation cycles as close to real time as possible, meaning that some sort of prediction or forecast for upcoming values may need to be programmed.

The first option which one might come up to solve this problem is applying a Kalman Filter. Of course one needs to think about the applicability and limitations of the Filter method beforehand.

2.3.2 Describing Kalman's filter algorithm

The Kalman Filter (KF) is a type of mathematical algorithm, named after one of its main developers, Rudolf E. Kálmán [47].

It can be used as an estimator for the linear quadratic problem. The future dynamic system which is impaired with *white noise* can be estimated using state space techniques, making it possible to use the Kalman Filter either as an estimator, filter or even as a smoother [48].

The Kalman Filter produces estimates of future unknown target variables. This estimation tends to be more precise than a simple estimation based on measurements only. The most known application of Kalman filtering is the navigation of aircrafts, spacecrafts, missiles, ships and many more [49].

In general a Kalman Filter compares a mathematical model to a measurement of the real life system and creates a prediction for the next state of the system. Both the mathematical model and the measurements suffer from imperfections. The system model as well as the measurements are multiplied with a weighting factor indicating the weighting of the model (often denoted as β) and the measurement (often denoted as α). A high α indicates a strong accuracy of the measurements, whereas a high β indicates a high accuracy of the model.

In the past years, many extensions or adaptions to the ”classic” Kalman Filter were developed, the most known are the ”Extended Kalman Filter” (EKF) (which is a non-linear version of the KF) and the ”Unscented Kalman Filter” (used for highly non-linear predict- and update functions) (UKF) [50]

The following listing procedure names the main steps of Kalman filtering: [51]

1. Initialization and first prediction
2. Measurement
3. State Update
4. Prediction
5. Repeating step 2-4 (new iteration)

Figure 2.6 shows a simplified schema of the Kalman Filter algorithm. It is an example for an recursive algorithm.

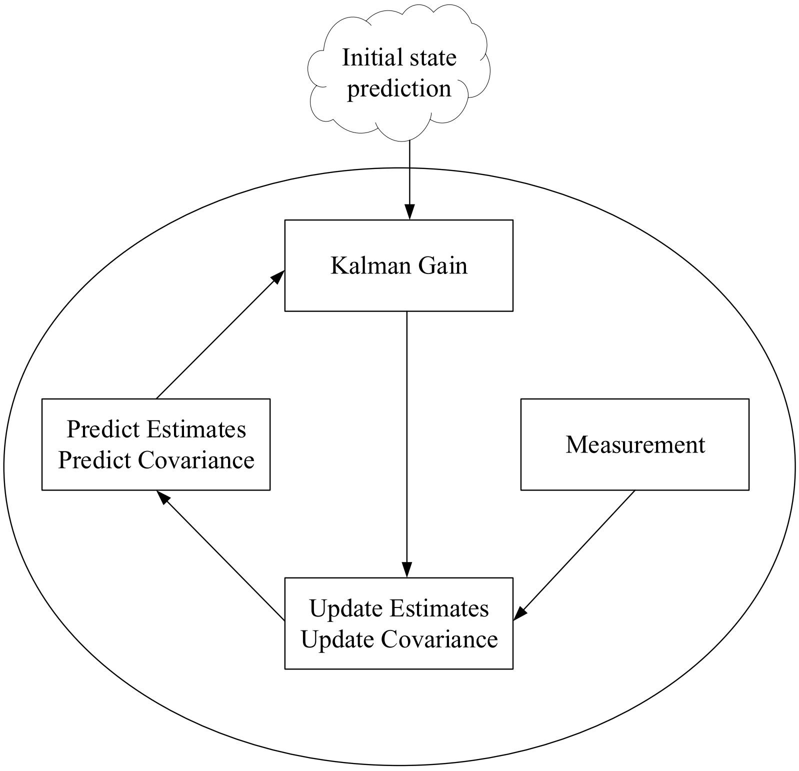

Figure 2.6: Schematic algorithm of a recursive Kalman Filter. This algorithm consists of an initialization and a recursive loop, in which the Kalman Gain, the estimates of the current state and the predictions for the upcoming iteration are computed.

A huge advantage of the Kalman Filter is its ability to be implemented as a recursive algorithm, meaning that besides of the underlying algorithm only the data of the previous cycle needs to be stored for the computation of the next cycle. This keeps the overall dynamic storage demand to a minimum, making it able to be implemented within an Integrated Sensor application.

Ad 1: The initialization is conducted at the beginning and it needs a guess of the system state and the state uncertainty. [51]

Ad 2: Like the initialization, the measurement step also delivers a system state and a uncertainty of the measurement. [51]

Ad 3: This step is responsible for the current state estimation of the system. Also the Kalman Gain and the outputs of the filter (System State Estimate and Previous State Estimate Uncertainty) are calculated [51].

Ad 4: In the final step the state and the uncertainty are extrapolated for the next system state and estimated using the provided model [51].

The derivation of the Kalman equations are outside the scope of this thesis and can be found in the original publication presented by Rudolf E. Kálmán [47] or various online tutorials and publications [48,51,52].
In this chapter the five main equations of Kalman Filters are taken as is. It is explained how these equations can be used to implement a filter. The five main questions of the Kalman Filter are (other names for the equations are written in brackets: [51]) :

1. Kalman Gain Equation (Weight equation)
2. State Update Equation (Filtering Equation)
3. Covariance Update Equation (Corrector equation)
4. Dynamic Model Equation (Predictor-, Transition-, Prediction Equation, or State Space Model)
5. Predictor Covariance Equation

We shall now take a closer look at the underlying equations. At the beginning, we distinguish between true values $\boldsymbol{x}$, estimates $\hat{x}$, predictions $\hat{x}'$ and measured values $\boldsymbol{z}$, all of those are functions over time, or in the discrete sense functions over some index k. All of the vectors above contain the value as well as all considered derivatives of it, e.g.:

$$\hat{\mathbf{x}}' = \begin{bmatrix} \hat{x}' \\ \dot{\hat{x}}' \\ \ddot{\hat{x}}' \\ \vdots \end{bmatrix} . \tag{2.30}$$

As already explained above the Kalman Gain $\mathbf{K}$ is a measure for the confidence in the model or the equations, a high $\mathbf{K}$ indicates us a weak model, whereas a low $\mathbf{K}$ shows us, that the Kalman Filter relies stronger on the model equations rather than the measurements. Using the following equation, the Kalman Gain can be computed easily by,

$$\mathbf{K} = \frac{\mathbf{P}'\mathbf{H}^T}{\mathbf{H}\mathbf{P}'\mathbf{H}^T + \mathbf{R}}. \tag{2.31}$$

In this equation $\mathbf{P}'$ is the predicted covariance matrix computed in the previous step (during the initialization an assumption on this matrix needs to be made)
$\mathbf{H}$ is the so-called observation matrix. $\mathbf{H}$ indicates which quantities of the vector $\boldsymbol{x}$ can be observed (or measured) in our system. The shape of $\mathbf{H}$ must be identical to $\boldsymbol{x}^T$:

$$\mathbf{H} = \begin{bmatrix} H & \dot{H} & \ddot{H} & \dots \end{bmatrix}. \tag{2.32}$$

If a quantity (e.g. position) and its considered derivatives (e.g. speed and acceleration) are observable, the correct entry is 1 and if not, the correct entry is 0.
Let's take a look at this little example: Consider a mathematical model for a car driving with almost constant speed along a straight road. This model needs information about the position and the speed of the vehicle ($\mathbf{H}$ is then of size $(1x2)$). Only the position is observed (measure) over time, yielding an Observation matrix $\mathbf{H}$ given by,

$$\mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{2.33}$$

And finally the matrix $\mathbf{R}$ is the covariance of the measurement noise. For the example above $\mathbf{R}$ yields to a scalar, as there is only one measured quantity, and $\mathbf{R}$ therefore simplifies to the variance of the measurement noise σ^2 as there is no variable it can correlate to.

Author Note: The standard uncertainty σ is not to be confused with the mechanical stress, which is also denoted by σ. I did not want to introduce new notations here and apologize for the inconvenience caused.

With this Kalman Gain, the updated estimate $\hat{x}$ for the current step can be calculated by applying the following equation. To calculate $\hat{x}$ we simply take our prediction of the estimate $\hat{x}$ acquired from the previous iteration (For the 1^{st} iteration an assumption for $\hat{x}$ needs to be made in the initialization step) and add the weighted residual of the measurement signal $\boldsymbol{z}$ and the prior estimate. When $\mathbf{K}$ is very small or even zero, the estimate for the current state becomes the prediction made in the previous state, yielding an accurate model. The state update equation is given by,

$$\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}' + \mathbf{K}(\boldsymbol{z} - \mathbf{H}\hat{\boldsymbol{x}}'). \tag{2.34}$$

The updated covariance matrix $\mathbf{P}$ can be calculated by the following equation. $\mathbf{I}$ is the Identity matrix of the same size as the product of $\mathbf{KH}$. Again, when dealing with a low or even zero Kalman Gain $\mathbf{K}$, the estimate for the covariance is close to (or even equal) to the prediction made in the state before:

$$\mathbf{P} = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}'. \tag{2.35}$$

With the updated estimates for the current step and the covariance matrix the predictions for the next step can be calculated using a matrix $\mathbf{F}$ which links future and current step.

To be more precise **F** contains the underlying equations based on some physical principle. It is called the Transition matrix. For instance: Let's assume that the change rate of the angle is constant for high sample rates, then **F** would look like this (dT denotes the time between two samples):

$$\mathbf{F} = \begin{bmatrix} 1 & dT \\ 0 & 1 \end{bmatrix}. \tag{2.36}$$

The equations for the predicted estimate is formed quite simply by,

$$\hat{\boldsymbol{x}}' = \mathbf{F}\hat{\boldsymbol{x}}. \tag{2.37}$$

Its shape is quite obvious when looking at the row-wise formulation. In the following set of equations the product of $\mathbf{F}\hat{\boldsymbol{x}}$ has been computed:

$$\begin{aligned} \hat{x}' &= \hat{x} + dT\dot{\hat{x}}, & (2.38) \\ \dot{\hat{x}}' &= \dot{\hat{x}}. & (2.39) \end{aligned}$$

Obviously, this set of equation does simply describe a motion of constant speed. The transition matrix **F** needs to be adapted to the given physical behaviour of the real system to work accurately.

The last step of each iteration is it to compute the predicted covariance matrix $\mathbf{P}'$:

$$\mathbf{P}' = \mathbf{F}\mathbf{P}\mathbf{F}^{\mathbf{T}} + \mathbf{Q}. \tag{2.40}$$

The matrix **Q** contains information about the process noise of the system is added to our updated covariance matrix. **Q** has a high influence on the smoothing and stability of the filter. Very small entries in **Q** result in high confidence of the modelled system, returning a smooth curve over time. On the other hand, such a **Q** would not be able to directly follow changes from the idealized model in the real system, even neglecting measurements in some extreme cases.

To sum up: A high **Q** allows the model to question the perfection of the model equations and returns some of the noise obtained by the measurements, whereas a small **Q** reduces noise but is slower in detecting changes in the real system.

Notes:

After all this excitement, the limitations and prerequisition of Kalman Filters shall not be forgotten to be stated here:

1. The classic Kalman Filter can only be properly used for linear-quadratic problems.
2. The measured sources need to be perturbed by Gaussian noise.
3. The filter only works as good as its model was programmed. If the model is not sufficient enough to describe the actual process, the filter will fail.

2.3.3 Simplifying the Kalman algorithm

The set of equations presented above presents the following conclusion:
The set of Kalman equations can be split in two parts, Equations (2.31), (2.35) and (2.40) can be computed separately. No information from the measurements is needed. Additionally, both the Covariance matrix and the Kalman Gain matrix converge after a decent amount of iterations, allowing to pre-define this matrices for a certain application. These equations do not need to be in the actual iteration cycles if the process noise matrix and measurement noise matrix are time-invariant. This reduces the actual filter work significantly as only Equations 2.34 and 2.37 need to be solved.

Let's take a look at the example above (recalling the car travelling with constant speed) and split the matrix equations into separate line-wise equations. Equation 2.34 then splits up into the following two equations:

$$\hat{x} = \hat{x}' + K_x \, x_{res}, \tag{2.41}$$

$$\hat{\dot{x}} = \hat{\dot{x}}' + K_{\dot{x}} \, x_{res}, \tag{2.42}$$

where K_x and $K_{\dot{x}}$ are the entries of the Kalman Gain matrix **K** (which is now assumed constant over time) given by,

$$\mathbf{K} = \begin{bmatrix} K_x \\ K_{\dot{x}} \end{bmatrix} \tag{2.43}$$

and x_{res} is the residual of the measurement result from the current state between the prediction of the previous state to the current one:

$$x_{res} = x^{\sim} - \hat{x}'. \tag{2.44}$$

Equation 2.37 was already adapted to the given example, let's recall Equations (2.38,2.39):

$$\hat{x}' = \hat{x} + dT\hat{\dot{x}}, \tag{2.45}$$

$$\hat{\dot{x}}' = \hat{\dot{x}}. \tag{2.46}$$

The actual Kalman Filter algorithm de-solves into a system of four simple equations (Equation 2.46 generates no new information and does not need to be calculated) which need to be iterated over time. This set of equations has only three multiplications, three additions and one subtraction per iteration cycle.

These simplifications now enables the Kalman Filter to be used within an integrated sensor. The computational power of the microcontrollers in use is sufficient enough to compute the set of rather simple equations. This concludes the proposal for a method to predict future state of a safety relevant sensor in automotive applications with Kalman filtering.
A first estimate of the computation time per Kalman Filter iteration yields about 5 μs, which is sufficient enough for sensors with typical sample rates specified in section 2.3.1.

Note:

The example of the car travelling with constant speed can be adapted to model a car travelling with constant acceleration, the remaining set of Kalman Filter equations is presented without derivation:

$$\begin{aligned} \hat{x} &= \hat{x}' + K_x\, x_{res}, && (2.47)\\ \hat{\dot{x}} &= \hat{\dot{x}}' + K_{\dot{x}}\, x_{res}, && (2.48)\\ \hat{\ddot{x}} &= \hat{\ddot{x}}' + K_{\ddot{x}}\, x_{res}, && (2.49) \end{aligned}$$

for the state update equations and,

$$\begin{aligned} \hat{x}' &= \hat{x} + dT\hat{\dot{x}} + dT^2/2\,\hat{\ddot{x}}, && (2.50)\\ \hat{\dot{x}}' &= \hat{\dot{x}} + dT\hat{\ddot{x}}, && (2.51)\\ \hat{\ddot{x}}' &= \hat{\ddot{x}}, && (2.52) \end{aligned}$$

for the state prediction equations. In this set of equations above, K_x, $K_{\dot{x}}$, $K_{\ddot{x}}$ (all entries of the Kalman Gain matrix) as well as dT and $dT^2/2$ are all constants.

2.4 Uncertainty modelling

2.4.1 What is uncertainty?

The dictionary of the Cambridge University defines uncertainty as *a situation in which something is not known, or something that is not known or certain* [53]. Any measurement result is fraught with systematic ("known") and random ("not-known") errors. The definition above implies that systematic errors are not uncertain, because they can be quantified, only random errors remain *uncertain*.

How do mechanical engineers deal with uncertainties?

Mechanical engineers usually need to have hard limits to reject or accept a batch of deliverables. Let us assume that a shaft with nominal length of 200 mm with a tolerance of ± 0.5 mm is required. Then a mechanical engineer would choose a notation given by,

$$x = (200 \pm 0.5)\text{mm}. \qquad (2.53)$$

Any shaft with a length exceeding 200.5 mm or deceeding 199.5 mm will be rejected. The production process is perturbed with random error sources resulting in an error distribution which could be Gaussian. Such a distribution indicates that the vast majority of shafts will have a length close to the nominal value but very few shafts can have much larger (or smaller). So the manufacturer would like to keep his uncertainty as low as possible. This has a huge effect on production costs. Let us also consider different batch sizes. A smaller batch has a lower probability of containing a rejected part than a batch with a much higher size. Even with this in mind, one needs to define a certain amount of parts within a batch which is allowed to exceed the given limits. In the automotive industry this limit is very low (→ section 1.3) combined with the typical batch sizes exceeding 10^6 parts per batch. In such vast batch counts rare outliers suddenly are not that unlikely any more, making it impossible to define hard limits. That is why one needs to think of the probability of such a tolerance range, e.g.: 99.95% of the parts lie within $x = (200 \pm 0.5)$

mm. This is a much accurate definition than Equation 2.53 alone. For a long time there was no standardized methodology to define uncertainties properly. For this reason the ”Guide to the Expression of Uncertainty in Measurement” (GUM) was created.

2.4.2 Guide to the expression of Uncertainty in Measurement (GUM)

The *Bureau International des Poids et Mesures* (BIPM) is an international organization, through which member states act together on matters related to measurement science and measurement standards aiming to be centre for scientific and technical collaboration between member states and being the coordinator of the world wide measurement system [54]. It is part of the *Joint Committee for Guides in Metrology* (JCGM) organization, which has released the GUM [55–59] and the *International vocabulary of metrology-basic and general concepts and associated terms* (VIM) [60]. The GUM, VIM and all their supplementary documents are available at [61] for free. The GUM does not give detailed instructions to solve a specific task but rather gives general rules for evaluating and expressing uncertainty in measurement.

In contrast to the definition of uncertainty above, the GUM defines uncertainty (of measurement) as [55]:

Parameter, associated with the result of a measurement, that characterizes the dispersion of the values that could reasonably be attributed to the measurand

GUM proposes the following notation for defining uncertain measurement results:

$$U = 10.04(25)\mathrm{V}. \tag{2.54}$$

This is obviously the measurement result of a voltage U in Volts. A proper definition needs the quantity of the measurand (U), the value of the measurand 10.04, its standard uncertainty (25) and its unit V. The standard uncertainty refers to the last digits of the corresponding result. This is a much proper definition of uncertainty than equation (2.53). A standard uncertainty of two digits implies that the least significant digit of the measurement result has to be arbitrary.
Please beware that GUM also allows a definition looking exactly like Equation (2.53), whereby GUM defines the number after ± as the (combined) standard uncertainty and not a confidence interval. For obvious reasons the latter definition is not preferred by GUM. Other definitions can be found in section 7.2.2 of [55].

Propagation and summarizing of uncertainty evaluation

JCGM 104:2009 [58] summarizes three methods for the propagation of uncertainty evaluation:

1. The GUM uncertainty framework

2. Analytic methods

3. Monte-Carlo simulations

GUM uncertainty framework

The GUM uncertainty framework [55] uses the law of uncertainty propagation and characterizes the output Z by a normal- or t-distribution. The GUM approach fails when [58]:

- measurement functions are non linear,
- asymmetric input variables occur and if output variables are not normally- or t-distributed,
- input variables do not have similar contribution to the combined uncertainty.

Evaluating standard uncertainty

JCGM 100:2008 [55] names two methods to evaluate standard uncertainties. Type A uses statistical methods based on results from experiments such as computing the standard deviation from a set of experimental data, whereas Type B uses a-priori knowledge of the input quantity such as data from previous measurements, personal experience or specifications given by the manufacturer. Type B is not more inaccurate than Type A in general, especially with a certain experience or a rather small sample size for Type A evaluations.

Determining combined standard uncertainty

The computation of the combined standard uncertainty $u_C(y)$ is presented in section 5 of JCGM 100:2008 [55]. A 1^{st} order Taylor approximation of the measurand $Y = f(X_1, X_2, ..., X_n)$ yields an approximation of the standard uncertainty. This approach is reasonably accurate if the transfer function f is reasonably linear, if this is not the case, higher order terms of the Taylor series approximation also need to be considered.
The combined standard uncertainty $u_C(y)$ of $Y = f(X_1, X_2, ..., X_n)$ where Y is the measurand and y is its estimate can be calculated by (Eq. 10 in [55]):

$$u_C(y) = \sqrt{\sum_{i=1}^{N} \left(\frac{\partial f}{\partial x_i}\right)^2 u^2(x_i)}. \tag{2.55}$$

The combined standard uncertainty can also be calculated for correlated variables $x_1, x_2, ..., x_n$ (Eq. 13 and 15 in [55]):

$$u_C(y) = \sqrt{\sum_{i=1}^{N} \left(\frac{\partial f}{\partial x_i}\right)^2 u^2(x_i) + 2\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{\partial f}{\partial x_i}\frac{\partial f}{\partial x_j} u(x_i)u(x_j)r(x_i, x_j)}, \tag{2.56}$$

where $r(x_i, x_j)$ is the correlation coefficient of x_i, x_j which can be calculated by,

$$r(x_i, x_j) = \frac{u(x_i, x_j)}{u(x_i)u(x_j)}, \tag{2.57}$$

where $u(x_i, x_j)$ denotes the estimated covariance of x_i, x_j.
Equations (2.55,2.56) are an expression of the *law of propagation of uncertainty* (as defined in section 3.1 of [55]).

Analytic Methods

Analytic methods yield an algebraic form for the probability distribution. They do not produce any approximations and can only be used for simple cases with very few input variables. Analytic methods were not used in the scope of this thesis, as the applications were too complex to be modelled analytically.

Monte Carlo method

The Monte Carlo method is explained in JCGM 101:2008 [56]. The distribution of the output Z can be determined approximately by taking random picks according to the probability distributions of the input variables and evaluating the model at the resulting values Z. A Monte Carlo simulation can be used to determine if an application is suitable for the calculation by the GUM framework. Monte Carlo simulation is more versatile and can handle non-linearities where the GUM approach would fail, it also outperforms GUM with increasing numbers of uncertain variables as the complexity of linearisation increases with the number of variables. On the other hand, Monte Carlo simulation requires many samples to deliver stable results, due to its non deterministic nature performing quite poor at the tails of the distribution, not allowing direct application to reliability investigations.

Every computation conducted with the GUM approach in the scope of this thesis has been cross-checked with the Monte Carlo method.

2.4.3 Central limit theorem

It states that in a case of a sequence of n independent, identically distributed random variables (r.v.) $X_1 ... X_n$ with mean μ and variance σ^2 and $\overline{X_n}$ being the arithmetic mean of $X_1 ... X_n$, then Z_n tends to the standard normal distribution ($\mu = 0$, $\sigma = 1$) for $n \to \infty$:

$$Z_n = \frac{X_1 + \cdots + X_n - n\mu}{\sigma\sqrt{n}} = \frac{\overline{X_n} - \mu}{\sigma/\sqrt{n}}. \tag{2.58}$$

For a large n, the sum $S_n = X_1 + X_2 + \cdots + X_n$ has approximately a Gaussian distribution regardless of the distribution of each X_i $(i = 1, 2, \ldots, n)$ [62]. The closer the distributions of each individual X_i are to a Normal distribution, the less X_i are required to yield a Normal distribution for Z_n. But even the convolution of very few extreme non-normal distributions such as a rectangular distribution with equal widths is approximately normal [55]. This is one of the reason for the vast popularity of Gaussian distributions.
Beware that the Central limit theorem is not applicable for the square of the sum of several Gaussian distributed random variables. Such a square of the sum yields to a distribution with much longer tails [63].

2.4.4 Software for uncertainty propagation

A vast number of both commercial and free to use software to perform uncertainty propagation is available for the most common programming environments such as Java, R, Python, Julia, MATLAB and even Microsoft Excel. Most of them can handle basic mathematical operations very well. More advanced functionality such as complex number calculation or the capability to handle correlations is only covered by a very few of them. Such advanced software is given in [64] and [65].

2.4.5 An introduction to ”An Uncertainty Toolbox” (AUT)

An Uncertainty toolbox (AUT) is a MATLAB based toolbox developed by the University of Klagenfurt [66].It uses the capabilities of object-oriented programming in MATLAB allowing storage and operation of data as well as interactions between different objects [67]. It has been investigated in previous publications [68–71] and theses [72]. This toolbox is yet in its developmental stage and will be available at [1] once released. The toolbox is able to handle the uncertainty propagation of the GUM framework [55] with correlations as well as complex numbers. Additionally a supplementary toolbox was developed to handle Monte Carlo simulations as well.

The semantics for using the GUM framework based approach is very easy. An uncertain variable can be created with a single line of code:

```
x=unc(mu,sigma,'name')
```

Listing 2.1: Defining an uncertain variable with AUT, where: `mu` represents the expected value of the random variable, `sigma` represents its standard uncertainty and `name` is a string used for variable identification

`x` does not necessarily has to be a single random variable, it can also be an array of any dimension `[m,n]`, by simply defining `mu` and `sigma` as arrays of the same type. The expected value of `x` can be extracted from the object by `gmv(x)` or in ”dot” notation by `x.value`. Similarly the standard uncertainty of `x` can be obtained by `gmu(x)` or by `x.std`.

MATLAB allows users to overload basic operators such as `+,−,/,*,^,..` and functions such as `min,max,sum,sqrt,diag,trace,sin,cos,tan,...` In this way the same operators and functions that the user has become familiar with can be used in the same way for `unc` variables too. A more detailed overview can be found in the documentation of the toolbox. As already explained above, the toolbox can also handle correlations quite well as illustrated in the following example:

```
x1=unc(10,1,'x1');
x2=unc(10,1,'x2');
x3=x1;
x1-x2
        >> ans= 0.0(1.4)
x1-x3
        >> ans= 0(0)
```

Listing 2.2: Some example code to illustrate the principle of correlation

In the first step, two independent random variables `x1,x2` are set. As `x1` and `x2` are not correlated, the expected value of `x1−x2` yields `0` and the standard uncertainty equals `sqrt(x1^2+x2^2)`. As `x1` fully correlates with itself, the standard uncertainty of `x1−x3` equals `0`.

Correlations between two random variables can also be set manually:

```
x1=unc(10,1,'x1');
x2=unc(10,1,'x2');
x3=unc(10,1,'x3');
x1-x2
        >>ans=0.0(1.4)
set_correl(x1,x2,1);
```

```
x1-x2
      >>ans=0(0)
set_correl(x1,x3,0.5);
x1-x3
      >> ans= 0.0(1.0)
```

Listing 2.3: Setting correlations between random variables manually

The command `set_correl(variable1,variable2,corrCoeff)` correlates `variable1` with `variable2` by the correlation coefficient `corrCoeff`.

The semantics for the Monte Carlo simulation based toolbox is actually very similar, new random variables with Gaussian distributions can be defined by:

```
x=unc_t(mu,sigma)
```

Listing 2.4: Defining an uncertain variable with AUT for Monte Carlo simulations, where: `mu` represents the expected value of the random variable and `sigma` represents its standard uncertainty

$10,000$ Monte Carlo samples are taken per default. This can be changed by setting the global variable `MCSAMPLES` to the desired value in the beginning of the script code. An alternative and convenient way to switch in-between GUM and Monte Carlo toolboxes is to insert the following line at the beginning of the script:

```
unc= @ unc_t
```

Listing 2.5: Calling unc_t by overriding unc

Further examples can be found in [68, 72] and the documentation of AUT.

2.5 The Tukey-Lambda distribution and its application

The Tukey-Lambda distribution (often only called Lambda distribution) is a family of symmetric distributions proposed by John Tukey in 1960 [73] and generalized by Ramberg and Schmeiser [74] in 1972 to also include asymmetric distributions.
The distribution can be defined numerically with the following three parameters:

1. shape parameter λ
2. location parameter μ (”mean”)
3. scale parameter σ (”standard deviation”)

Most commonly, the percent point function (PPF) is used to describe the standardized Lambda distribution (omitting μ and σ). A PPF is the inverse of the cumulative distribution function (CDF). Given a cumulative input probability p the corresponding ”x”-value is computed for a given λ.

$$G(p,\lambda) = \frac{p^{\lambda} - (1-p)^{\lambda}}{\lambda}, \tag{2.59}$$

with,

$$\lambda > -1.$$

The shape of the CDF is given for four different λ in Figure 2.7. The given λ are not chosen arbitrarily. $\lambda = -1$ resembles a good approximation of a Cauchy Distribution, $\lambda = 0.14$ looks very similar to the CDF of the standard normal distribution, $\lambda = 0.5$ resembles a "U-shaped" distribution and $\lambda = 1$ looks exactly like a uniform distribution in $[-1, 1]$.
That is why there is no unique probability density function to describe all possible values of λ. The Lambda Distribution is therefore not used for statistical modelling but rather to approximate another symmetric distributions or to determine the distribution of a given set of symmetrically distributed random variables.

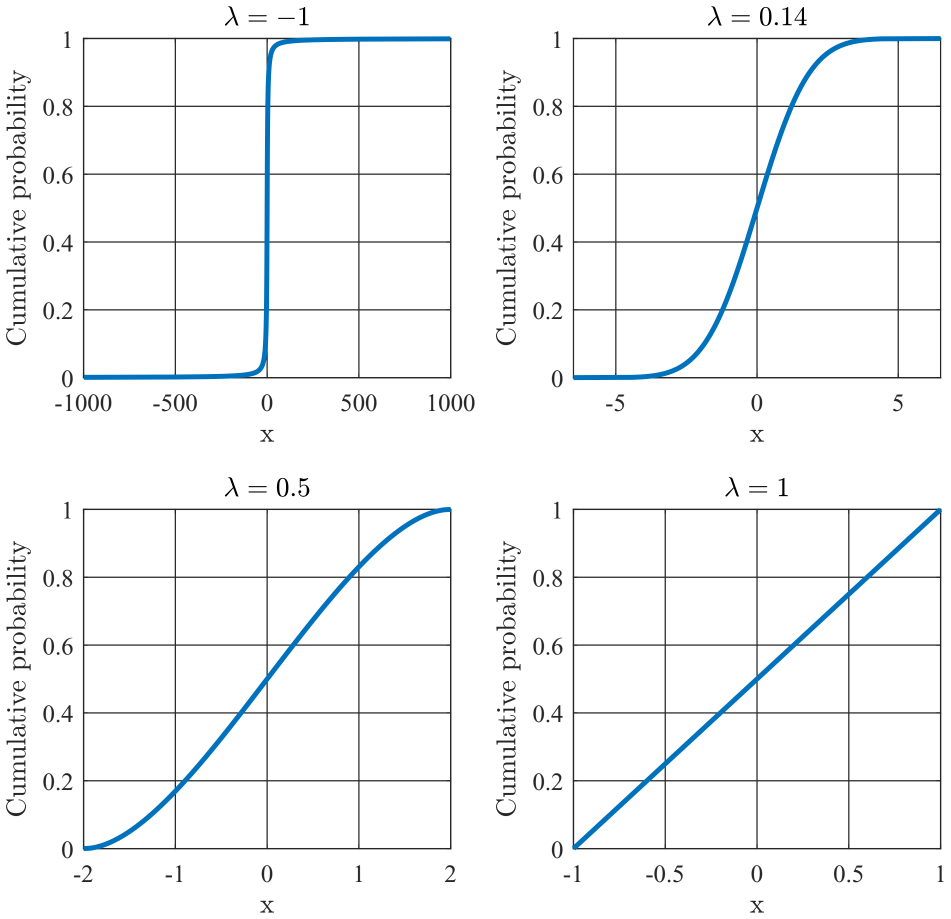

Figure 2.7: Cumulative Distribution Function (CDF) of the Lambda distribution for $\lambda = [-1, 0.14, 0.5, 1]$. **Upper Left Plot:** λ =-1 is a good approximation of a Cauchy Distribution. **Upper Right Plot:** λ =0.14 is a good approximation of a Gaussian distribution. **Lower Left Plot:** λ =0.5 resembles a "U"-shaped distribution. **Lower Right Plot:** λ =1 is an exact rebuild of a uniform distribution.

The Lambda distribution is an exact rebuild of the

1. Logistic distribution for $\lambda = 0$

2. Uniform distribution in $[-1, 1]$ for $\lambda = 1$ (Figure 2.7)

The following distributions can be approximated using the Lambda distribution (which is also shown in Figure 2.7):

1. Cauchy distribution for $\lambda \approx -1$
2. Normal distribution for $\lambda \approx 0.14$
3. U-shaped distribution for $\lambda \approx 0.5$

Probability Plot Correlation Coefficients (PPCC) plots can be used to identify the distribution of a given set of symmetrically distributed random variables. The correlation coefficient of the given Lambda distribution and the set of random variables is plotted against the shape parameter λ. An example of such an PPCC is given Figure 2.8. The correlation coefficient of $100,000$ Gaussian distributed random variables versus the Lambda distribution for $\lambda = [-1...1]$ is shown. The maximum of this function lies at 0.14 and therefore implies that a Gaussian distribution is suitable for the given random variables.

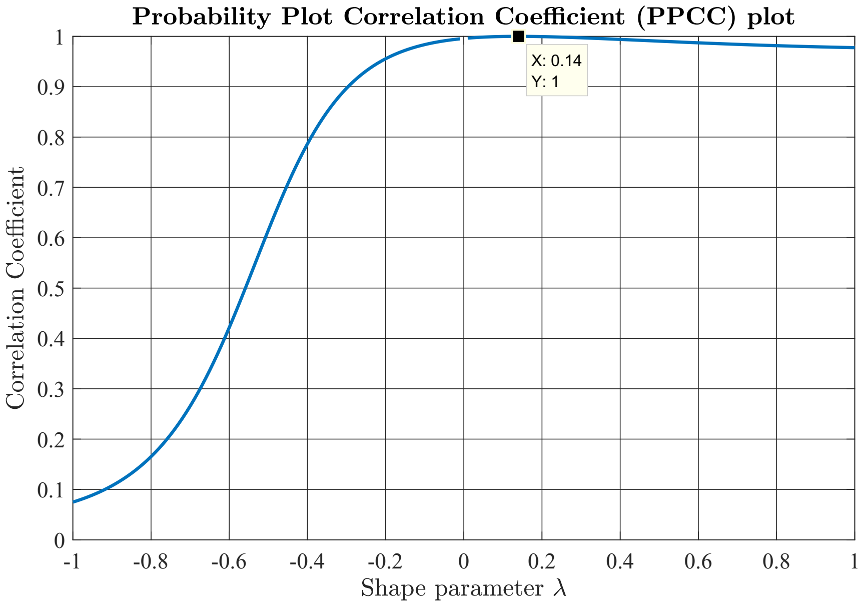

Figure 2.8: Probability Plot Correlation Coefficient (PPCC) plots for set of $100,000$ Gaussian distributed random variables. The λ value at the peak of this curve is a strong indication of the statistical distribution of the underlying random variable.

The implementation using MATLAB is quite simple. In the 1^{st} step the Tukey-Lambda distribution is set up in MATLAB:

```
function gp=tukey(p,lambda)
if lambda ~=0
gp=(p.^lambda-(1-p).^lambda)./lambda;
else
gp=log10(p/1-p);
end
end
```

Listing 2.6: Code to compute the percent point function (ppf) (Equation (2.59)) of a Tukey-Lambda distribution

The calculation of the correlation coefficient between the given distribution and the Lambda distributions with λ in $[-1,1]$ is shown by the following MATLAB code snippet:

```
% Create a set of random variables
x=randn(100000,1);
% Sort this set in increasing order
y=sort(x);
% Set up a linspace vector of probabilities
b=linspace(1/(2*length(x)),1-1/(2*length(x)),length(x));
% Pre-assign storage for the Correlation Coefficient matrix
C=zeros(2,2,201);
% Compute the Correlation Coefficient matrix
for i=1:201
lambda=(i-101)/100;
v=tukey(b,lambda);
C(:,:,i)=corrcoef(v,y');
end
% Pre-assign storage to correlation coefficient vector
corrCoef=zeros(201,1);
% Collect the the correlation coefficients
for i=1:201
corrCoef(i)=C(1,2,i);
end
```

Listing 2.7: Code to compute to correlation coefficients for a given distribution against a set of Tukey-Lambda distributions. The function `corrcoef` is used to compute the correlation matrix between the given set of measurements and the Tukey-Lambda distribtuions for different values of the shape parameter λ.

Be aware that the given distributions need to be symmetric. This can be pre-checked by using e.g. histograms. The Tukey Lambda distribution is not implemented in most of the commercial software. The PPCC plot can also be calculated online by using the tool presented available at [75], which is based on the theory presented in [76].

Other well known tests in statistics are Kolmogorov-Smirnov test, Lilliefors test, Shapiro-Wilk test and many more. The advantage of the PPCC plot of Tukey-Lambda over other test algorithms lies in its versatility to test against multiple distributions at once.

Chapter 3

Modelling Signal Processing & Uncertainty

3.1 Calculation of magnetic fields

3.1.1 3D field solution

The calculation of a three dimensional field solution of the magnetic field created by a permanent magnet may become a non-trivial task, with increasing complexity of the magnet geometry. Calculating magnetic fields from scratch lead to solving Maxwell's Equations. For simple geometries, such as cylinders or bars, many solutions of analytical, numerical or polynomial approximated nature have been presented in numerous publications. Additionally, Finite Element Analysis also provides a way to evaluate complex geometries too. For the given task an accurate, fast and MATLAB compatible approach was searched to model the field created by a diametrically polarized disc magnet. Both numerical solutions [77] and analytical solutions [78] were considered. All considered approaches delivered results with satisfying accuracy. Naturally, the numerical solution prompts results faster as analytical solutions such as [78], as they rely on integral solving. In the MATLAB environment, these integrals had to be solved numerically undermining the concept of analytical modelling to a certain degree. Ultimately, the numerical solution of [77] was chosen.
During the stage of integration of the Uncertainty Toolbox, also 3D polynomial-approximated solutions of [77] were analysed to further improve the speed of the computations at the cost of accuracy. Such 3D polynomial approximations have also been considered by [79] as tri-cubic polynomial approximations. These approximated solutions deliver decent results, when the region of interest (ROI) is kept low. With an increasing ROI the results increase in error. This can be reduced by either cutting the complete ROI into multiple segments and applying case differentiations or alternatively coding a piece-wise polynomial approximation in three dimensions. Although polynomial approximations of the magnetic field were considered, they were discarded as maximum accuracy which was aimed to be achieved.

There are of course many software tools (both freeware and commercial) allowing to perform numerical computations. Analytical solvers are also available, allowing to do fast computations without the need to rely on time consuming FEA models. Such a tool is MagPyLib [80], a freeware tool written in Python.

3.1.2 3D solution for a stray field robust Hall sensor with 360° detection

With the approaches presented in section 3.1.1 the magnetic field created by a diametrically polarized disc magnet (magnet properties are specified in Table 3.1) can be calculated in any point in space outside the magnet.

Table 3.1: Magnet properties.

Magnet diameter	D	6 mm
Magnet height	H	2.5 mm
Remanence	B_r	1100 mT
Material	–	Samarium-Cobalt (SmCo)

This section aims to present the mathematical relation needed to calculate the magnetic field at each Hall plate location of the sensor system. The sensor system consists of four Hall plates in total, which are located on a circle with reading radius (*RR*) around the center of a die (Equation 3.1 consider also Figures 1.3 and 3.1). Their position on the die with respect to the die center is best described in vector notation in Cartesian coordinates:

$$\boldsymbol{h_1} = \begin{bmatrix} 0 \\ -RR \\ 0 \end{bmatrix} \quad \boldsymbol{h_2} = \begin{bmatrix} -RR \\ 0 \\ 0 \end{bmatrix} \qquad \boldsymbol{h_3} = \begin{bmatrix} 0 \\ +RR \\ 0 \end{bmatrix} \quad \boldsymbol{h_4} = \begin{bmatrix} +RR \\ 0 \\ 0 \end{bmatrix} \tag{3.1}$$

where $\boldsymbol{h_1}$ defines the position of Hall 1 with respect to the die center and $\boldsymbol{h_2}$ to $\boldsymbol{h_4}$ define the position of the consecutive Hall elements. Typical values of *RR* lie between $5 - 25\%$ range of the magnet diameter D. For the given magnet in Table 3.1 typical values for lie between $RR \approx 0.3 - 1.5$ mm. Choosing small *RR* allows to build smaller sensors at the cost of signal strength, as the magnetic field is very small straight below the magnet's center. A large *RR* delivers much higher signal strength at the cost of larger sensor area needed. Unless stated otherwise, a *RR* of 1 mm was chosen in all computations as this proved to be an optimal compromise

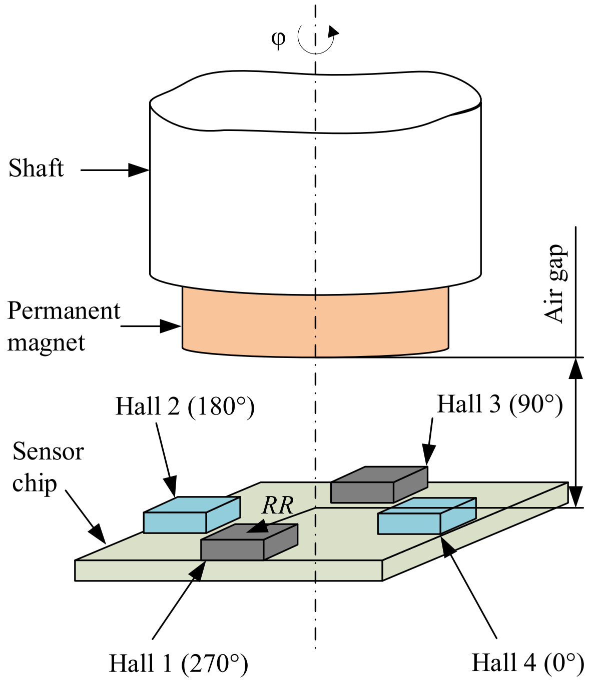

Figure 3.1: Schematic overview of the angle sensor system. An end-of shaft solution (compare Figure 1.2) has been chosen. Four Hall elements are placed on a circle at a reading-radius *RR* from the shaft center. This figure has also been published in [63].

The die, which carries the Hall plates, may have an arbitrary orientation within its package. Both the sensor and the magnet are afflicted with assembly tolerances, which are of random nature, resulting in arbitrary trajectories of each Hall plate. A misplacement of the magnet can be caused by imperfect magnet positioning while mounting the magnet on the shaft (which is commonly done be glueing). An inhomogeneous glue film thickness may even cause a tilt of the magnet with respect to shaft's plane surface.

A possible solution for modelling this sensor system is using coordinate transformations to switch in between coordinate systems (CS). A total of three CS were chosen to describe the position and orientation of all components: Magnet CS, Sensor CS and Shaft CS. A schematic overview of the model is shown in Figure 3.2.

Major parts of this section were also published in [63].

This system has twelve degrees of freedom (DoF) in total consisting of the sensor displacement $\mathbf{d_S} = [x_S, y_S, z_S]^T$, magnet misplacement $\mathbf{d_M} = [x_M, y_M, z_M]^T$ in Cartesian coordinates and the orientations of the sensor $[\alpha_S, \beta_S, \gamma_S]$ and magnet $[\alpha_M, \beta_M, \gamma_M]$ with respect to the reference (=shaft) coordinate system. All DoF are listed in Table 3.2. Eight of these DoF can be used to describe assembly tolerances (γ_M and γ_S are not considered as assembly tolerances as they have linear effects on the angle error and are simply cancelled out during phase shift calibration. The effects of z_S and z_M are usually covered separately).

Table 3.2: Overview of Degrees of Freedom (DoF). There are a total of twelve degrees of freedom.

Sensor DoF	**Magnet DoF**
Misplacement ($\mathbf{d_S}$)	Misplacement ($\mathbf{d_M}$)
x_S y_S z_S	x_M y_M z_M
Orientation (w.r.t. shaft CS)	Orientation (w.r.t. shaft CS)
α_S β_S γ_S	α_M β_M γ_M

In proper vector notation $\mathbf{d_S}$ and $\mathbf{d_M}$ look like:

$$\mathbf{d_S} = \begin{bmatrix} x_S \\ y_S \\ z_S \end{bmatrix}, \quad \mathbf{d_M} = \begin{bmatrix} x_M \\ y_M \\ z_M \end{bmatrix}. \tag{3.2}$$

Besides all these parameters also the ”air gap” AG is a highly important parameter in sensor applications. AG is the perpendicular distance between the top surface of the sensor package and the bottom surface of the magnet. It is given by,

$$AG = z_M + z_S - H/2. \tag{3.3}$$

Typical values for AG lie between 0.5 to 4 mm, usually depending on external conditions.

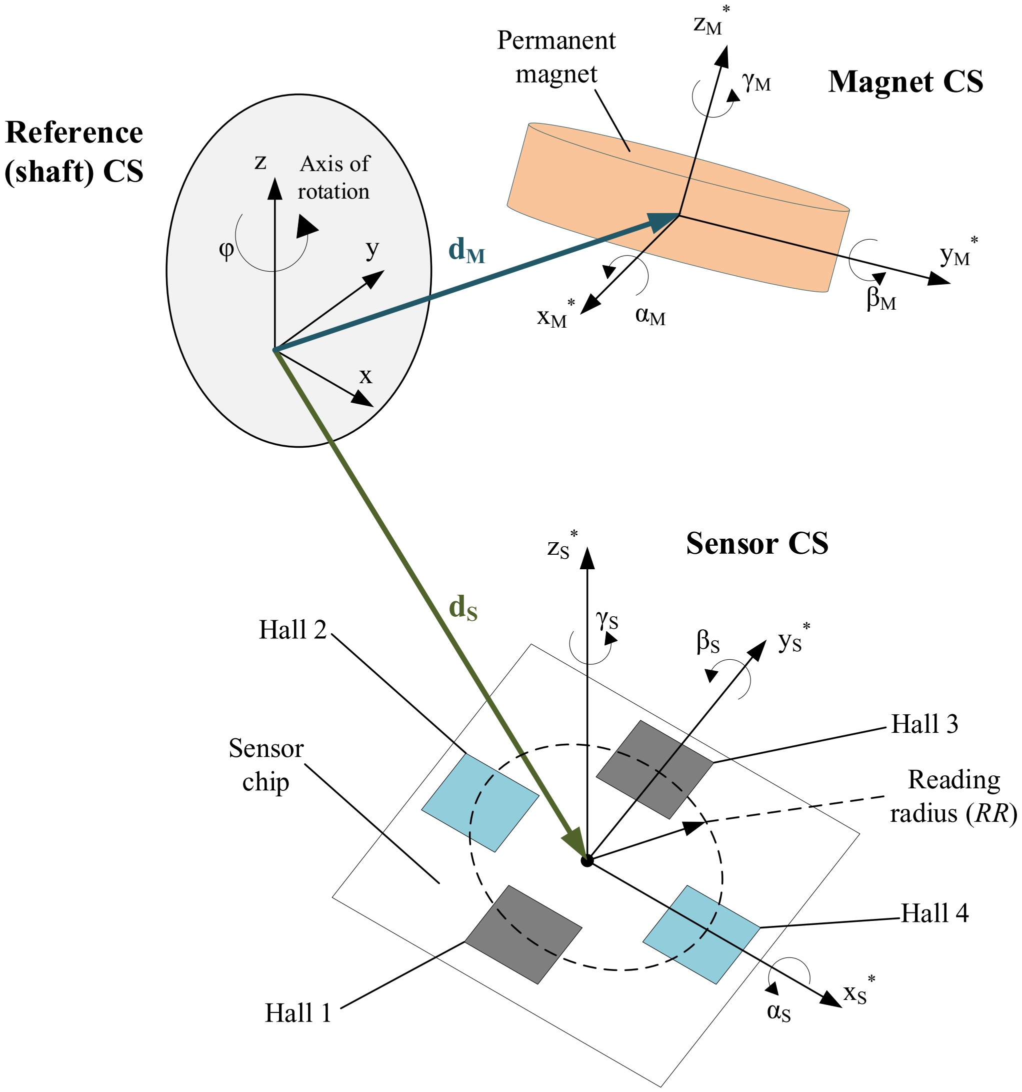

Figure 3.2: Schematic overview of the different coordinate systems (CS). The relative CS of the magnet is defined by a displacement vector $\mathbf{d_M} = [x_M, y_M, z_M]^T$ and a set of Euler angles $[\alpha_M, \beta_M, \gamma_M]$. The sensor CS is defined similarly: $\mathbf{d_S} = [x_S, y_S, z_S]^T$ and $[\alpha_S, \beta_S, \gamma_S]$. A set of multiple coordinate transformations are needed to calculate the position and orientation of each Hall plate with respect to the magnet center. This figure has also been published in [63].

How to switch in-between different coordinate systems? One prominent solution are transformation matrices $\mathbf{T}(\mathbf{n}, \alpha)$ such as:

$$\mathbf{x}^* = \mathbf{T}(\mathbf{n}, \alpha)\mathbf{x}. \tag{3.4}$$

Such a coordinate transformation allows not only to transform a vector $\mathbf{x}$ to meet the orientation of another CS $\mathbf{x}^*$, but also to perform rotations of the vector by an angle α

around a certain axis $\mathbf{n}$ with $\mathbf{n} = [n_1, n_2, n_3]^T$. $\mathbf{T}(\mathbf{n}, \alpha)$ can be found in literature [81, 82] and is cited here:

$$\mathbf{T}(\mathbf{n},\alpha) = \begin{bmatrix} n_1^2(1-cos(\alpha))+cos(\alpha) & n_1n_2(1-cos(\alpha))-n_3sin(\alpha) & n_1n_3(1-cos(\alpha))+n_2sin(\alpha) \\ n_2n_1(1-cos(\alpha))+n_3sin(\alpha) & n_2^2(1-cos(\alpha))+cos(\alpha) & n_2n_3(1-cos(\alpha))-n_1sin(\alpha) \\ n_3n_1(1-cos(\alpha))-n_2sin(\alpha) & n_3n_2(1-cos(\alpha))+n_1sin(\alpha) & n_3^2(1-cos(\alpha))+cos(\alpha) \end{bmatrix}. \tag{3.5}$$

This vast matrix can be greatly simplified when Cartesian axes are used as rotation axes. To rotate by a certain angle α around $\mathbf{n}_x = [1, 0, 0]^T$ $\mathbf{T}(\mathbf{n}, \alpha)$ becomes:

$$\mathbf{T}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha) & -sin(\alpha) \\ 0 & sin(\alpha) & cos(\alpha) \end{bmatrix}. \tag{3.6}$$

The rotation matrices for rotations around the y- ($\mathbf{n}_y = [0, 1, 0]^T$) and z- ($\mathbf{n}_z = [0, 0, 1]^T$) axes by angles of β and γ respectively are derived in the same way and yield:

$$\mathbf{T}_y(\beta) = \begin{bmatrix} cos(\beta) & 0 & sin(\beta) \\ 0 & 1 & 0 \\ -sin(\beta) & 0 & cos(\beta) \end{bmatrix}, \tag{3.7}$$

for rotating around the y- axis and

$$\mathbf{T}_z(\gamma) = \begin{bmatrix} cos(\gamma) & -sin(\gamma) & 0 \\ sin(\gamma) & cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.8}$$

for rotations around the z- axis.

All three rotation matrices can be grouped to a single line yielding,

$$\mathbf{x}^* = (\mathbf{T}_x(\alpha)\mathbf{T}_y(\beta)\mathbf{T}_z(\gamma))\mathbf{x}, \tag{3.9}$$

allowing to rotate against all three Cartesian axes by the that exact order.

Be aware that matrix multiplications are not generally commutative [83]. When performing reverse rotations the order of the transformation matrices need to be mirrored.

Transforming coordinates of Hall plates with respect to Magnet CS

The process of transforming the coordinates of the Hall plates to match the orientation of the magnet is a multi-step process. Let us begin with taking the relative position of each Hall element w.r.t. to the sensor center (Equation (3.1) and rotate them to meet the orientation of the reference CS. Recalling Equation (3.9) the new coordinates $\mathbf{h_i}^*$ can be calculated by,

$$\mathbf{h_i}^* = (\mathbf{T}_x(\alpha_S)\mathbf{T}_y(\beta_S)\mathbf{T}_z(\gamma_S))\mathbf{h_i}, \tag{3.10}$$

where,

$$i = \{1, 2, 3, 4\}, \tag{3.11}$$

indexes the corresponding Hall plate.

All Hall plates have the same orientation as the shaft CS, by adding the sensor displacement vector $\mathbf{d_S}$ the absolute position of the Hall plates can be calculated by,

$$\mathbf{h_{abs,i}} = \mathbf{d_S} + \mathbf{h_i}^*. \tag{3.12}$$

The next step is to perform a full rotation around the axis of revolution z. In practice, the sensor remains static as the magnet is mounted to the rotating shaft. This kind of inverse kinematics is in fact easier to compute here. $\mathbf{h_{abs,i}}$ now becomes an array of position vectors $\mathbf{H_i}$ in $[0,360]^\circ$ described by,

$$\mathbf{H_i} = \mathbf{T}_z([0,360]^\circ)\mathbf{h_{abs,i}}. \tag{3.13}$$

$\mathbf{H_i}$ is now rotated again to meet the magnet CS orientation in a similar way as above,

$$\mathbf{H_i}^* = (\mathbf{T}_x(-\alpha_M)\mathbf{T}_y(-\beta_M)\mathbf{T}_z(-\gamma_M))\mathbf{H_i}. \tag{3.14}$$

Like before, the offset vector $\mathbf{d_M}$ needs to be added to this vector array do yield the final coordinates $\mathbf{H_m}$, given by its equation,

$$\mathbf{H_{m,i}} = \mathbf{d_M} + \mathbf{H_i}^*. \tag{3.15}$$

Calculating the magnetic field vector

After feeding the numerical field solution [77] with the array of coordinates $\mathbf{H_{m,i}}$ its solution yields,

$$\mathbf{B_i} = \begin{bmatrix} B_{x,i} \\ B_{y,i} \\ B_{z,i} \end{bmatrix}, \tag{3.16}$$

which is an array of magnetic fields for a full rotation for each Hall plate.

Transforming the magnetic field vector with respect to the sensor CS

$\mathbf{B_i}$ is an array of magnetic field vectors. In a multi-step process $\mathbf{B_i}$ needs to be rotated back to meet the sensor's orientation. The first step is to rotate this array to meet the reference CS using transformations matrices given by,

$$\mathbf{B_i}^* = (\mathbf{T}_z(\gamma_M)\mathbf{T}_y(\beta_M)\mathbf{T}_x(\alpha_M))\mathbf{B_i}. \tag{3.17}$$

Note that the order of the transformation matrices has now been flipped to perform a valid re-transformation. Also recall that inverse kinematics have been applied before, where the sensor has been rotated instead of the magnet. That is why the field solution so far is just a static field solution, as the orientation of magnetic field vector changes when the magnet is rotated. To compensate this, the magnet rotation is simulated by,

$$\mathbf{B_{rot,i}} = \mathbf{T}_z([0,-360]^\circ)\mathbf{B_i}^*. \tag{3.18}$$

The computation is almost complete. In the last step, $\mathbf{B_{rot,i}}$ needs to be tilted one last time to meet the orientation of the sensor:

$$\mathbf{B_{sensor,i}} = (\mathbf{T}_x(\alpha_S)\mathbf{T}_y(\beta_S)\mathbf{T}_z(\gamma_S))\mathbf{B_{rot,i}}. \tag{3.19}$$

$\mathbf{B_{s,i}}$ contains all three components of the magnetic field vector as an array in $[0,360]^{\circ}$. Each Hall element only detects the z-component of this field vector array. The x- and y-component were nevertheless needed for computing the z-field, and therefore must not be neglected beforehand. They are not needed for the further computations. But this model could also allow computing the field of magneto resistive sensor or Vertical Hall sensor, which do actually need information about x- and y-component of the field vector.

The z-component of $\mathbf{B_{s,i}}$ (denoted by $B_{s,zi}$) could be directly used to calculate the angular error arising from a certain set of assembly tolerances. The measured angle $\varphi^{\tilde{}}$ can be calculated by (also consider section 1.4.2) determing the magnetic field for each channel given by,

$$B_{x-Ch} = B_{s,z2} - B_{s,z4}, \tag{3.20}$$

for the x-channel and by,

$$B_{y-Ch} = B_{s,z1} - B_{s,z3}, \tag{3.21}$$

for the y-channel. The measured angle $\varphi^{\tilde{}}$ is then calculated by,

$$\varphi^{\tilde{}} = \arctan2(B_{x-Ch}, B_{y-Ch}). \tag{3.22}$$

The resulting angle error solely caused by assembly tolerances is then given by,

$$\Delta\varphi = \varphi^{\tilde{}} - \varphi. \tag{3.23}$$

For the further signal processing path only the z-component of the magnetic field array will be needed. For that reason $B_{s,z1}$ to $B_{s,z4}$ is grouped to a 2D array given by,

$$\mathbf{B_z} = \begin{bmatrix} B_{s,z1} \\ B_{s,z2} \\ B_{s,z3} \\ B_{s,z4} \end{bmatrix}. \tag{3.24}$$

Some restrictions arising from the calculation

The methods presented above do hold some restrictions:

1. The numerical field solution [77] assumes an ideal magnet with a fixed and uniform polarization.
2. The model assumes infinitesimal small Hall plates. The actual magnetic field detected by the Hall plates may vary slightly.

3.1.3 Results

In this section, the results of two exemplary runs are shown for a full shaft rotation $\varphi = [0,360]^{\circ}$, which is split into samples of 1° width, using the algorithm described in section 3.1.2. The assembly parameters used can be found in Table 3.3. The results are shown in Figures 3.3-3.4. The upper sub-figure shows a plot of the z portion of the magnetic field $B_{s,z}$, which is detected by each Hall plate, the center sub-figure shows the theoretical magnetic fields of both channels (compare Equations (3.20-3.21)) and the bottom subfigure shows a plot of the angle error $\Delta\varphi$ solely related to the assembly tolerances of the model (compare Equation (3.22)).

Table 3.3: Overview of Degrees of Freedom (DoF) values which were used in the computations presented in Figures 3.3-3.4. Instead of defining z_S, z_M AG was defined according to Equation (3.3). Magnet properties according to Table 3.1. Additionally AE error (compare Figure 1.4) is shown for both runs.

Run	RR	AG	x_S	y_S	α_S	β_S	γ_S	x_M	y_M	α_M	β_M	γ_M	AE error
—	mm	mm	mm	mm	°	°	°	mm	mm	°	°	°	°
Run 1	1	2	0	0	0	0	0	0	0	0	0	0	≈ 0
Run 2	1	2	0.1	0.1	1	1	0	0.1	0.1	1	1	0	0.181

Run 1 represents an ideal set of assembly parameters, without the presence of tolerances. Regardless of the reading radius RR and air gap AG values chosen the model should yield an angle error $\Delta\varphi \approx 0$ for $\varphi = [0,360]^\circ$. The results are presented in Figure 3.3.

The curves for $B_{s,z}$ yield perfect sinusoidal curves with a phase shift of 90° between each consecutive Hall element. The differential signals for the x- and y-channel are perfectly sinusoidal with doubled amplitude. As expected $\Delta\varphi$ is approximately zero for the whole rotation cycle. The noisy error distribution is just a combination of the numerical accuracy of the numerical 3D field solution used and some numerical inaccuracies of the `atan2` function in Matlab.

The results of Run 2 are shown in a similar way in Figure 3.4. Assembly tolerances cause distortions in amplitude and phase to the perfect sinusoidal behaviour of each $B_{s,z}$ curve shown in the upper plot. This is also detected by the channel voltages depicted in the central plot. An angle error can be detected. This angle error is not only affected by the magnitude of the assembly tolerances alone but is also affected by the combination magnet dimensions (D, H) and choice of RR and AG. Some of these effects have been analysed and published in [63]. A more detailed investigation would go far beyond the scope of this thesis.

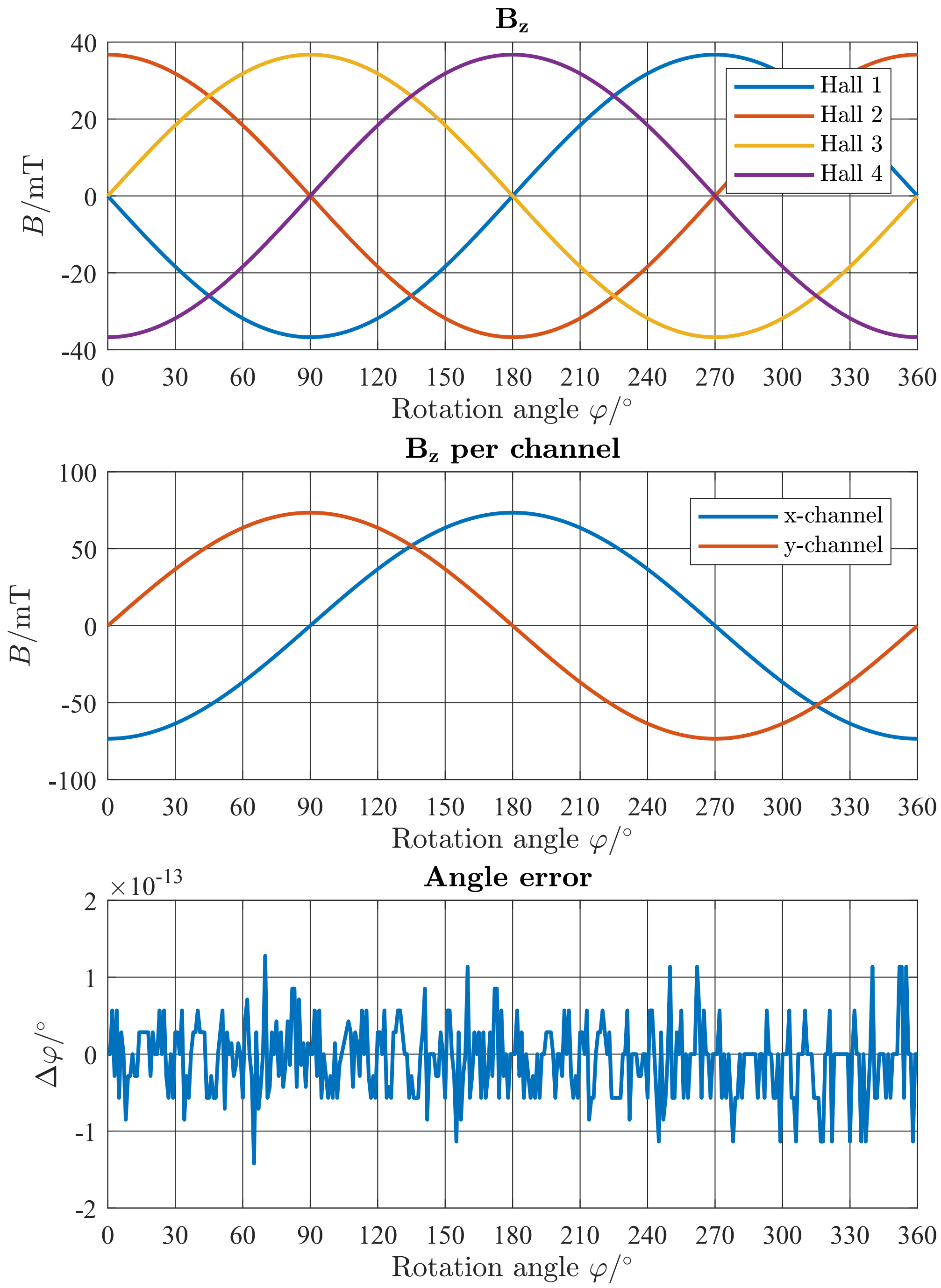

Figure 3.3: Results of the magnetic field calculation of a full shaft rotation $\varphi = [0, 360]°$ for Run 1 (assembly tolerances according to 3.3). **Top Plot**: $B_{s,z}$ of each in Hall plate. **Center Plot**: Hypothetical channel magnetic fields for x-channel and y-channel. **Bottom Plot**: Angle error $\Delta\varphi$ resulting from the given set of assembly tolerances (compare Run 1 of Table 3.3).

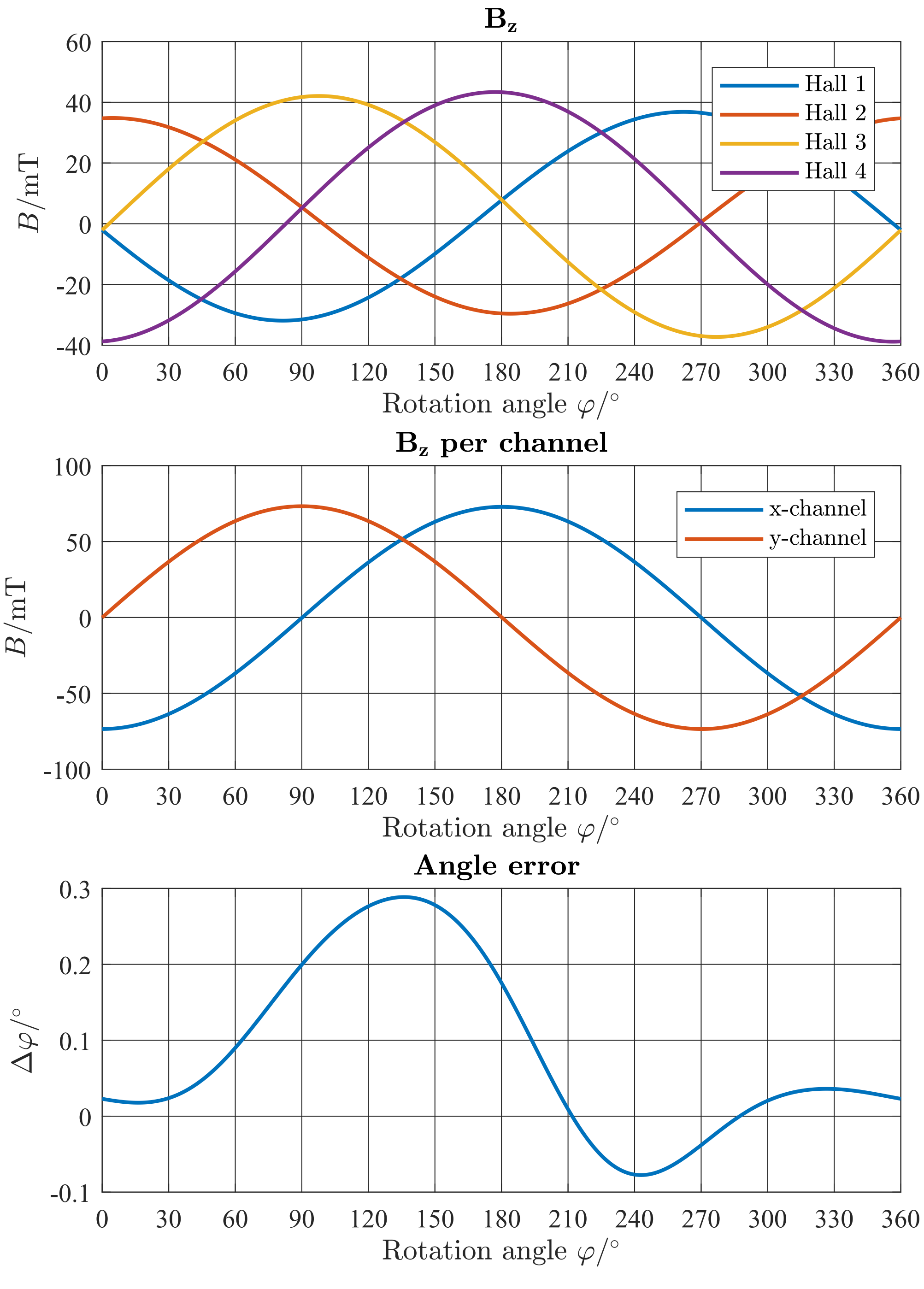

Figure 3.4: Results of the magnetic field calculation of a full shaft rotation $\varphi = [0, 360]°$ for Run 2 (assembly tolerances according to 3.3). **Top Plot**: $B_{s,z}$ of each in Hall plate. **Center Plot**: Hypothetical channel magnetic fields for x-channel and y-channel. **Bottom Plot**: Angle error $\Delta\varphi$ resulting from the given set of assembly tolerances (compare Run 1 of Table 3.3).

3.1.4 Model parameters for uncertainty analysis

The next step is to perform an uncertainty analysis of the MATLAB script. All assembly tolerances are now defined as ”uncertain” variables. The rest of the script may remain unchanged except for some restrictions which are discussed in the next section. An exemplary variable definition in MATLAB is given below in Listing 3.1. Table 3.4 specifies all statistical position and angle tolerances, which were used in the statistical analysis in section 3.1.5. Eccentricities and tilts were Gaussian distributed with standard deviations of $\sigma = 0.1$ mm and $\sigma = 1°$, respectively (Gaussian distributed random variables are denoted by $\mathcal{N}(\mu, \sigma^2)$ and uniformly distributed random variables are denoted by $\mathcal{U}([min, max])$ in Table 3.4):

Table 3.4: Overview of assembly tolerances (eccentricities and tilts) for sensor and magnet. These are reasonable values for today’s mass production processes. This set of assembly tolerances was also used in [63] and is based on [23]

Sensor tolerances	**Magnet tolerances**
$x_S = \mathcal{N}(0, 0.01)$ mm	$x_M = \mathcal{N}(0, 0.01)$ mm
$y_S = \mathcal{N}(0, 0.01)$ mm	$y_M = \mathcal{N}(0, 0.01)$ mm
$z_S = \mathcal{N}(0, 0)$ mm	$z_M = \mathcal{N}(0, 0)$ mm
$\delta_S = \mathcal{N}(0, 1)°$	$\delta_M = \mathcal{N}(0, 1)°$
$\eta_S = \mathcal{U}(0, 360)°$	$\eta_M = \mathcal{U}(0, 360)°$
$\alpha_S = \delta_S\, cos(\eta_S)$	$\alpha_M = \delta_M\, cos(\eta_M)$
$\beta_S = \delta_S\, sin(\eta_S)$	$\beta_M = \delta_M\, sin(\eta_M)$
$\gamma_S = \mathcal{N}(0, 0)°$	$\gamma_M = \mathcal{N}(0, 0)°$

Assuring an equal probability for the sensor and the magnet to be tilted around arbitrary axes in the xy plane of the sensor (magnet) requires uniformly distributed auxiliary angles η_S, η_M were introduced. The tilt angles of sensor and magnet are δ_S, δ_M. It holds $\alpha_M = \delta_M\, cos(\eta_M)$, $\beta_M = \delta_M\, sin(\eta_M)$, $\alpha_S = \delta_S\, cos(\eta_S)$, $\beta_S = \delta_S\, sin(\eta_S)$. This is also illustrated in Figure 3.5. This approach has also been used in [63].

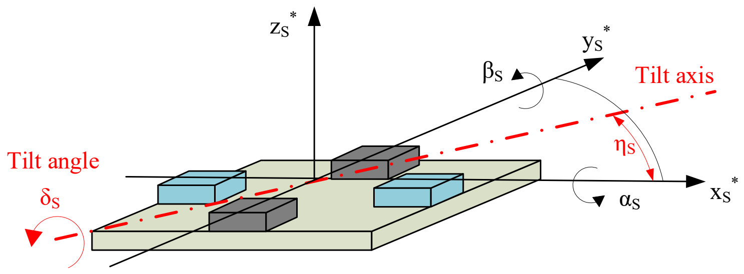

Figure 3.5: How to assure an equal probability of the orientation in the xy plane of the sensors’ tilt axis. A uniformly distributed auxiliary angle η_S is introduced as the angle between the sensor x-axis x_S^* and the tilt axis. The tilt angle δ_S is then the actual tilt angle of the sensor. The same procedure has been performed on the magnet (compare Table 3.4).

```
xS=unc(0,0.0001,'xS')
```

Listing 3.1: Defining the x- component of sensor displacement with respect to the shaft center x_S as an uncertain variable with mean 0 and standard uncertainty of 0.0001 m. Note: units of physical quantities are defined as standard SI units

3.1.5 Magnetic field calculation with AUT

As mentioned above, the MATLAB code of the magnetic field calculation was adapted to allow calculations with the unc and unc_t environment. This adaption is conducted rather fast as only the uncertain input variables need to be redefined as shown in section 2.4.5 and Listing 3.1. Precautions have to be taken, when built-in MATLAB functions are used in the script such as zeros() or cart2pol(), as Matlab is not capable of processing other than default objects such as double, strings or integers. In contrast to basic mathematical operations (which may be overloaded), MATLAB does not allow overriding such functions. A possible remedy is to redefine this function under a different name e.g. UNCzeros or UNCcart2pol.

The magnetic field calculation was conducted for the GUM approach (unc) and validated against Monte Carlo simulations (unc_t). The magnet properties can be found in Table 3.1. The set of assembly tolerances chosen are defined in Table 3.4.

The GUM approach

The results of the GUM based approach (unc) are presented in Figure 3.6 in the same manner as in section 3.1.3. A full rotation of the shaft is split up into 64 equally spaced angular increments. The dashed lines in each plot indicate the $\pm 3\sigma$ limits of the uncertain results. The sinusoidal waves of $B_{s,z}$ show a rather high but mere constant uncertainty of 20 mT (3 σ). In contrast to an amplitude of approximately 37 mT this uncertainty is rather high. This uncertainties seem to cancel each other perfectly when observing the differential signals B_{x-Ch} and B_{y-Ch} in the central plot. The angle error $\Delta\varphi$ shown in the bottom plot is practically zero. The first order terms of the Taylor series expansion seem to cancel each other out perfectly leaving no angle error. Knowing that the contribution of higher order terms of the Taylor series expansion usually delivers much smaller contributions, one might be tempted to say that assembly tolerances of gradient field based angular sensors do not produce any significant angle errors. This is obviously not the case [23, 63]. A Monte Carlo simulation using AUT (unc_t) should clarify this dilemma.

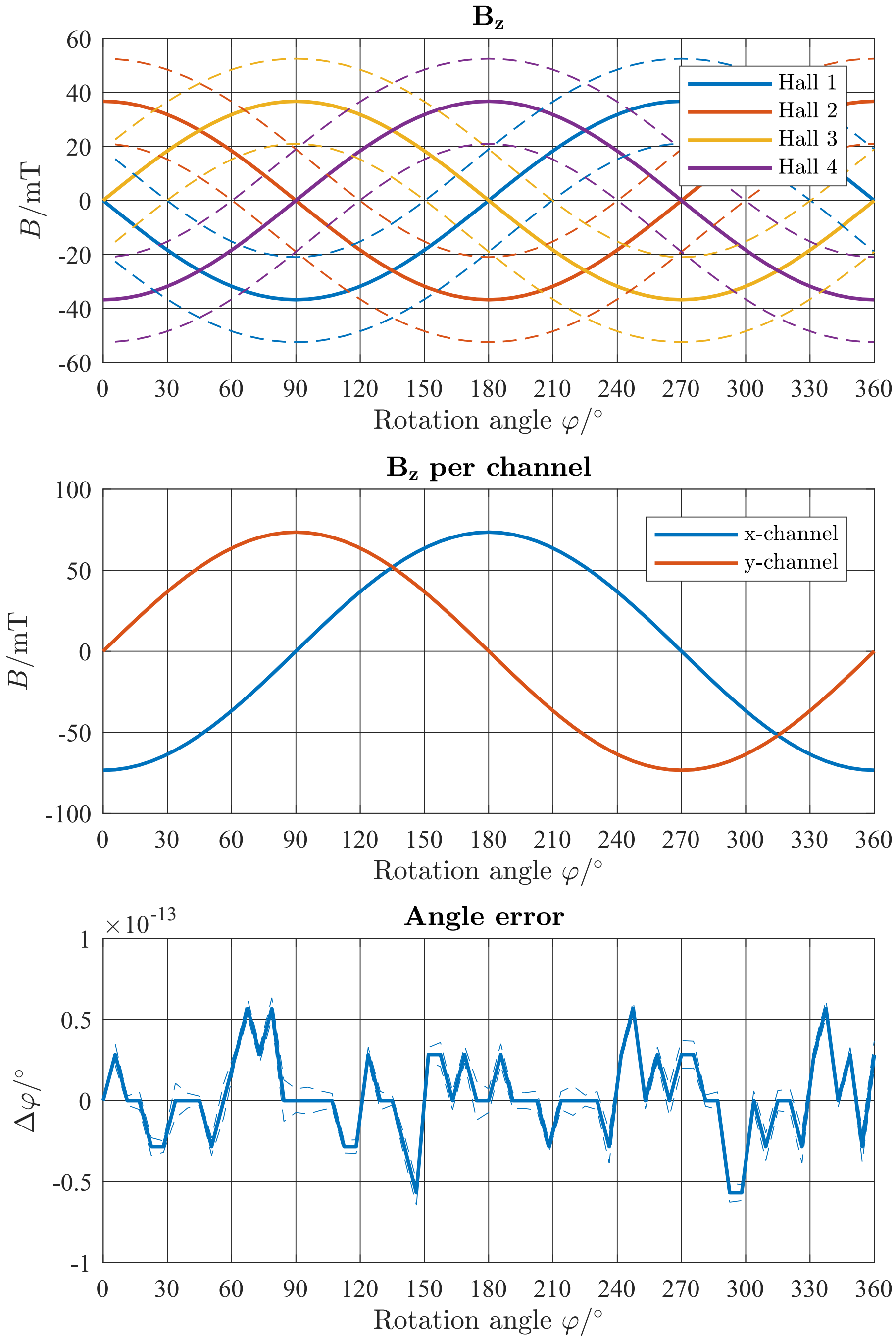

Figure 3.6: Magnetic field result of the GUM based (`unc`) approach. **Top Plot**: $B_{s,z}$ of each in Hall plate. **Center Plot**: Hypothetical channel magnetic fields for x-channel (B_{x-Ch}) and y-channel (B_{y-Ch}). **Bottom Plot**: Angle error $\Delta\varphi$ as a result of the given set of uncertain assembly tolerances. The dashed lines in each plot show the $\pm 3\sigma$ limits of the uncertain result. Although the magnetic field detected by each Hall plate is afflicted with a rather large portion of uncertainty, they seem to cancel each other out, when looking at the uncertainty of the hypothetical magnetic field of each channel. According to GUM no angle errors are detected. The computation time was $\approx 25,000$ seconds.

Monte Carlo simulation

As a next step, a Monte Carlo simulation using the Monte Carlo based approach of AUT has been investigated. The result of the Monte Carlo simulation (`unc_t`) for $10,000$ Monte Carlo runs is shown in Figure 3.7. A full rotation of 360° is again split up in 64 equally spaced angle increments. Like above, the dashed lines indicate the $\pm 3\sigma$ limits of the uncertain results. Like before, the uncertainty of each magnetic field portion is rather high, but its effects cancel (almost) each other out, the uncertainties of B_{x-Ch} and B_{y-Ch} are very small and non detectable in the graph. This is due to the cancellation of first order terms. The second and higher order error terms however remain, yielding an angle error, which is clearly non neglectable. This is due to the additive effects of multiple assembly tolerances being present simultaneously [23, 63].

The standard deviation of $\Delta\varphi$ should be a straight horizontal in theory, due to the statistical uncertainty in finite Monte Carlo simulations a small ripple remains.

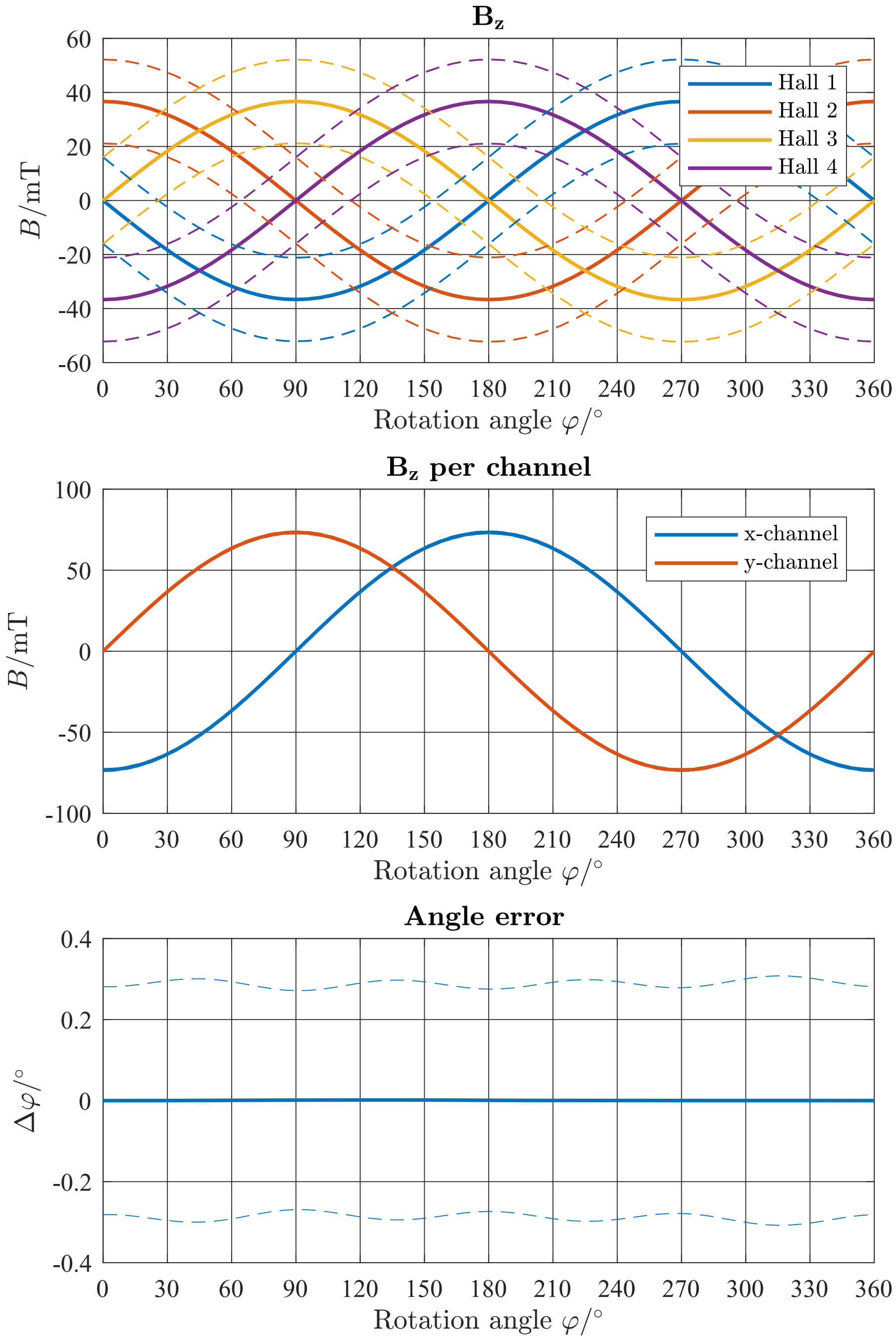

Figure 3.7: Magnetic field results of the Monte Carlo (`unc_t`) approach. **Top Plot**: $B_{sensor,z}$ of each in Hall plate. **Center Plot**: Hypothetical channel magnetic fields for x-channel (B_{x-Ch}) and y-channel (B_{y-Ch}). **Bottom Plot**: Angle error $\Delta\varphi$ resulting from the given set of assembly tolerances. The dashed lines in each plot show the $\pm 3\sigma$ limits of the uncertain result. 10,000 Monte Carlo samples taken. Although the magnetic field detected by each Hall plate is afflicted with a rather large portion of uncertainty, they seem to cancel each other out, when looking at the uncertainty of the hypothetical magnetic field of each channel. A resulting angle error is yet present, which has not been detected with the GUM approach. The computation time was $\approx 1{,}000$ seconds.

Résumé

As a conclusion of the result presented above, the GUM approach (`unc`) is not applicable of calculating the uncertain angle error of a gradient field based angle sensor in its current state. This is owed to the fact that the toolbox only covers first order term of the Taylor series expansion, which is sufficient enough to calculate uncertainties in general. Unfortunately, the modelling of Gradient Field based angle sensor is one of these few exceptions. The second and higher order terms of individual assembly tolerances produce small errors, but the combination of multiple assembly tolerances produce much higher angle errors that are non-neglectable anymore.

Modifying the source code of AUT to additionally compute of some next higher order terms of the Taylor series expansion could resolve this issue. This would lead to much longer computations. Using GUM based `unc` variables is already more than 25 times slower than pushing Monte Carlo simulations with $10,000$ runs, although one might expect the exact opposite behaviour. The complexity of the linearization process increases with the number of correlated uncertain variables. All this makes the `unc` approach (at least in its current state) not applicable for this task.

3.2 Hall effect device and datapath modelling

3.2.1 Model of the datapath

The whole signal processing path is shown as a schematic in Figure 3.8. This rather complex signal processing path can be split up in multiple parts:

1. Defining input values
2. Biasing
3. Hall voltage calculation
4. Channel voltage calculation
5. Digitalization
6. Applying end-of-line calibration parameters (compare section 2.2.2)
7. Stress compensation
8. Temperature compensation
9. Angle calculation

The compensation of temperature and stress dependencies do have their own signal processing path. These paths are called auxiliary paths” whereas the Hall signal path will be further paraphrased as ”main path”.

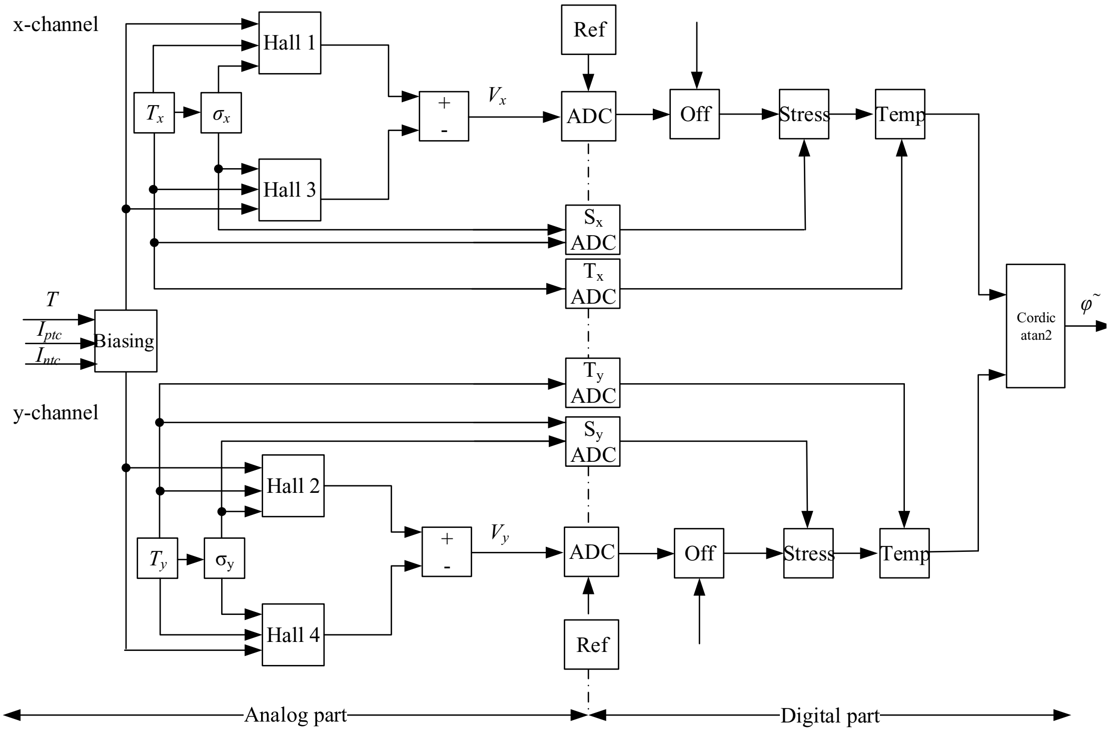

Figure 3.8: Schematic overview of the angle sensor system. Off: Offset-, phase- and amplitude compensation; Stress: Stress compensation, Temp: Temperature compensation, T: Temperature, σ: mechanical stress

Defining input values

The most important input parameter for this model is the magnetic field at each Hall plates' location given in matrix notation by,

$$\mathbf{B} = \begin{bmatrix} B_{z1} \\ B_{z2} \\ B_{z3} \\ B_{z4} \end{bmatrix} . \tag{3.25}$$

Despite the small size of Hall effect devices, an inhomogeneous temperature distribution may need to be considered, as small drifts of temperature between the Hall elements may cause additional errors. Especially incautious sensor design may lead to additional inhomogeneities of the temperature field as the circuitry of the sensor produces heat as well. Therefore the temperature is also defined in matrix notation given by,

$$\mathbf{T} = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} . \tag{3.26}$$

In some cases simplification to the temperature field can be made. Channel temperatures $T_x = \frac{T_1+T_3}{2}$, $T_y = \frac{T_2+T_4}{2}$ and the overall average temperature $T_m = \frac{T_1+T_2+T_3+T_4}{4}$ can be used. Another important input value is the current related sensitivity of the Hall plate s_I in V/(A T). It is mainly a function of the material (compare section 2.1) but is also affected by stress and temperature. Additionally imperfections of the Hall effect devices also need to be considered such as the Hall offset voltage V_o (which is also influenced by ambient temperature). Typical values for V_o are $10-1000\ \mu$V.

The input values of individual elements of the circuitry will not be highlighted in the scope of this thesis.

Temperature dependencies

All electric components show a temperature dependency. In most cases this temperature dependency can be modelled with sufficiently accuracy using a linear model. The following example shows the calculation of the resistance R_T of a resistor as a function of temperature T:

$$R_T = R_{RT}\ (1+\alpha_R\ (T-T_{RT})), \tag{3.27}$$

where R_{RT} is the resistance at room temperature T_{RT} (20° or 25°), α_R is the temperature coefficient of that resistor. In cases, when a linear model is not sufficient enough a second order polynomial can be alternatively chosen to model the temperature behaviour of the resistor. A quadratic temperature coefficient β_R can be introduced by,

$$R_T = R_{RT}\ (1+\alpha_R\ (T-T_{RT})+\beta_R\ (T-T_{RT})^2). \tag{3.28}$$

More complex models of temperature dependencies are needed in rare cases.

Stress dependencies

Hall effect IC's are sensitive to mechanical stress. This mechanical stress is mainly caused by the manufacturing and packaging process [84]. This stress itself depends on the ambient temperature. There are also other sources for stress drifts which are difficult to quantify. Under humid ambient conditions the plastic package can soak up water yielding such unruly stress dependencies [85]. Moisture dependant stress changes are not covered in this thesis.

Additionally, mechanical stress is not a scalar like temperature leading to much more complex compensation algorithms. Mechanical stress is described by an 3×3 tensor consisting of six different stress terms ($\sigma_{xy}=\sigma_{yx}$, $\sigma_{xz}=\sigma_{zx}$, $\sigma_{yz}=\sigma_{zy}$ due to the symmetry of this tensor). The first index of each individual stress component defines the orientation of the surface and the second index describes the orientation of the force. A typical stress tensor is given by,

$$\mathbf{S} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix}. \tag{3.29}$$

The shape of the package (much smaller thickness in comparison to its length and width) allows it to be treated as a laminate, reducing the number of individual stress components down to three (**S** is now a 2×2 square matrix). Recent research of [86] analysing the stress pattern within a package has found out that the (geometric) sum of the normal stresses $\sigma_{xx}+\sigma_{yy}$ is nearly constant at the package center, with the remaining shear stress component σ_{xy} vanishing, yielding a homogeneous stress state. This assumption is not valid close to the package edges. Additionally the (geometric) combination of $\sigma_{xx}+\sigma_{yy}$ is invariant to in-plane rotations. This allows an algebraic addition of $\sigma_{xx}+\sigma_{yy}$. The algebraic sum of $\sigma_{xx}+\sigma_{yy}$ is therefore a scalar.

The influence of mechanical stress generated by the sensor-mould-compound was modelled using a linear stress model, such as shown in Figure 3.9. The internal stress (pressure) increases with decreasing temperature and reaches zero above 150°C.

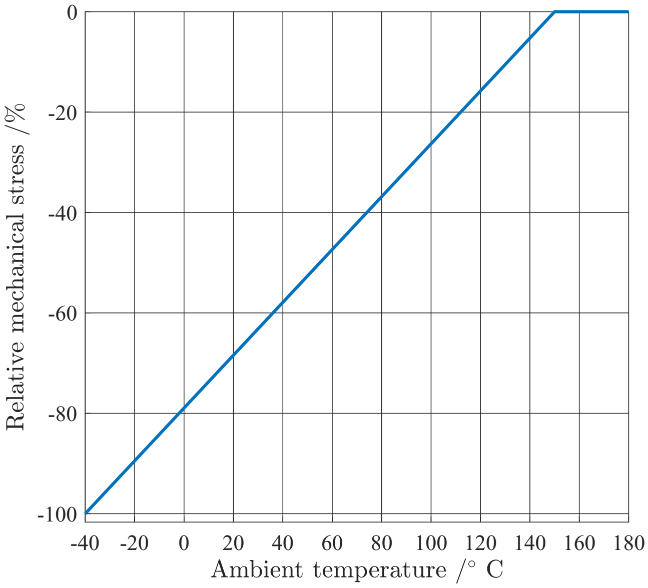

Figure 3.9: Mechanical stress vs. temperature within sensor package. When optimal layout of the sensing elements is chosen, such a linear model is sufficient enough for stress modelling

Biasing

Biasing is needed for a constant (DC) voltage supply of the integrated circuit. In sensor design this bias voltage has to be kept as constant as possible over temperature. As mentioned in section 2.1 there are two options for biasing. Voltage- and current mode biasing, whereby current mode biasing is the preferred solution due to higher temperature stability [5].

Once the bias current I_B is obtained, the bias voltage given by,

$$V_B(T) = R_B(T)I_B(T), \tag{3.30}$$

can be calculated when the equivalent resistance of the bias circuitry R_B is known.

There are multiple options in bias circuit design. There may be a single bias circuit for each Hall plate, or most commonly a biasing circuit for each ”channel”. In some cases each Hall plate has its own biasing circuit. A single bias circuit reduces statistical errors as the differential signal cancels out the uncertainties of the bias circuit. On the other hand, a single bias source requires more power as it powers all Hall plates.
The Matlab model uses an overall bias circuit for all Hall plates. This can nevertheless be adapted to multiple biasing circuits with little coding effort.

Hall voltage calculation

In the model, the Hall output voltage V_H is computed in a multi-step process, adding the influencing quantities in order. In the first step the temperature dependant Hall voltage V_{HT} is computed using the magnetic flux density at each Hall element B_z, the current-related sensitivity s_I and the bias current I_B. V_H is given by,

$$V_{HT} = B_z\, s_I(T)\, I_B. \tag{3.31}$$

The stress dependency is added in the next step. In most cases a quadratic approximation of the stress influence is sufficient using stress coefficient for the linear $s_{\sigma L}$ and quadratic $s_{\sigma Q}$ portion. Unfortunately these stress coefficients are not temperature invariant and need to have an approximation of their behaviour under temperature as well. The equation for the stress dependant Hall voltage $V_{H\sigma}$ is given by,

$$V_{H\sigma} = V_{HT}(1 + s_{\sigma L}(T)\,\sigma + s_{\sigma Q}(T)\,\sigma^2). \tag{3.32}$$

The Hall plates are also affected by Back-Bias (*BB*) effects caused by the common-mode voltage V_{CM}. Again a quadratic approximation using linear ($s_{BB\,L}$) and quadratic ($s_{BB\,Q}$) coefficients is quite sufficient to model this effect. The equation for the back-biased Hall voltage $V_{H\,BB}$ is given by,

$$V_{H\,BB} = V_{H\sigma}(1 + s_{BB\,L}\, V_{CM} + s_{BB\,Q}\, V_{CM}^2). \tag{3.33}$$

Next, the offset voltage of each Hall element is added to the model of the voltage signal to obtain the offset dependant Hall voltage V_f given by,

$$V_f = V_{H\,BB} + o_{HC} + o_{HL}\,(T - T_{RT}) + o_{HQ}\,(T - T_{RT})^2. \tag{3.34}$$

Again, a temperature dependency has to be considered. As before a quadratic model comprising of a constant o_{HC}, linear o_{HL} and quadratic o_{HQ} term is modelled.

The computation of the Hall output voltage is almost completed. In the final step, thermal noise V_T and an eventual amplitude mismatch V_{mis} is modelled. In order to cancel out correlation effects, the noise needs to be individually defined for each Hall plate. To reduce the overhead in notational footnotes, this ”final” Hall output voltage is denoted by V_H given by,

$$V_H = V_f + V_T + V_{mis}. \tag{3.35}$$

The amplitude mismatch V_{mis} is only added to a single Hall plate. This is sufficient enough to model an amplitude mismatch between the Hall voltages.

The algorithm described above is performed for each Hall plate yielding an array of Hall voltages $\mathbf{V}_H$ given by,

$$\mathbf{V}_H = \begin{bmatrix} V_{H1} \\ V_{H2} \\ V_{H3} \\ V_{H4} \end{bmatrix}. \tag{3.36}$$

Channel voltage calculation

After the Hall output voltages for each Hall element are obtained, the voltage of each channel can be calculated by,

$$V_X = V_{H2} - V_{H4}, \tag{3.37}$$

for the x-channel and by,

$$V_Y = V_{H1} - V_{H3}, \tag{3.38}$$

for the y-channel.
V_x and V_y may be grouped to an array for convenience purposes such as:

$$\mathbf{V} = \begin{bmatrix} V_X \\ V_Y \end{bmatrix}. \tag{3.39}$$

The analog part of the sensor model is completed. The channel voltages are now digitalized in the next step.

Digitalization

The digitalization is performed using Analog-Digital Converters (compare section 2.2.1). Considering Equation (2.5) (Resolution Q of a ADC) the ideal ADC output can be calculated by,

$$\begin{bmatrix} H_{ADC,x} \\ H_{ADC,y} \end{bmatrix} = \frac{1}{Q} \begin{bmatrix} V_X \\ V_Y \end{bmatrix}. \tag{3.40}$$

Error sources of each ADC (compare section 2.2.1) can be added in a similar way as for Hall voltages. The necessity of adding error contribution is dependent of their influence to the result.

In most cases 12 to 16 bit ADC's provide sufficient accuracy for these applications. The output is usually provided in $2's$ complement to keep the information about the sign, which is later on useful for the angle calculation.

Offset and phase shift compensation

A possible solution for offset and phase shift compensation has been explained in section 2.2.2.

Stress compensation

Unfortunately, no analogue signal is obtainable which is solely affected by the mechanical stress, as the circuit is always perturbed with thermal influences. One possible solution is to detect the difference between a stress and temperature dependant signal to a temperature dependant signal only. A schematic of this algorithm is shown in Figure 3.10.

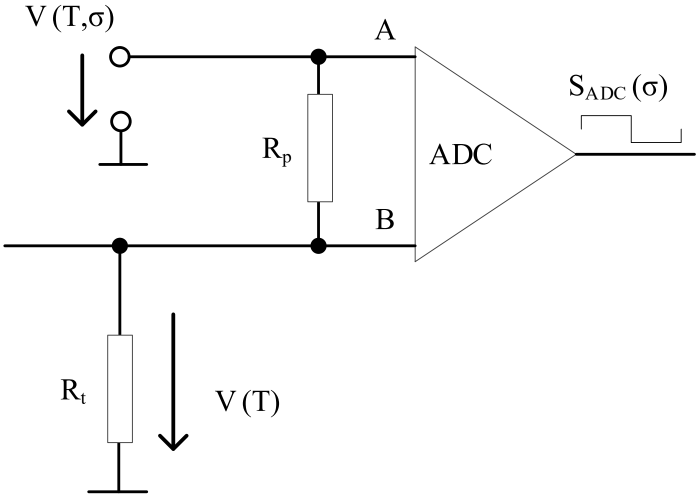

Figure 3.10: Digital stress-signal generation, where the main part of temperature dependence is compensated by using a temperature dependent signal on input B. S_{ADC} is the digitalized stress dependent output. R_p and R_t are dropping resistors in each input path.

Like biasing, S_{ADC} can either be computed per channel (using an average path temperature in the model) or a single stress ADC for both channels combined.

Temperature compensation

Under increasing temperature the output values of the main path ADC (H_{ADC}) decreases. Compensating this behaviour can be accomplished using a temperature related signal as for stress compensation (compare to paragraph above). Such a temperature related signal can be obtained from the Biasing path. Similar to Stress Compensation this signal is digitalized and its output values (T_{ADC}) are then used to reduce the temperature dependency of H_{ADC}.

Angle calculation

In real life application the angle calculation is performed using a CORDIC (Coordinate Rotation Digital Computer) algorithm. This CORDIC algorithm enables the calculation of "arctan2", an modification of the "arctan" algorithm allowing unique resolution in $[0,360)^\circ$ [87]. (The classic "arctan" is only unique in $(-90,+90)^\circ$.) To achieve this "arctan2" has two inputs (*sin* and *cos* portion). The combination of the portion's sign can be used to determine the quadrant, the angle lies in.

MATLAB provides a function for calculating the "arctan2", making the calculation of the measured angle $\varphi\tilde{}$quite easily [88] by,

$$\varphi\tilde{} = \text{atan2}(H_{ADC,y}, H_{ADC,x}). \tag{3.41}$$

By obtaining $\varphi\tilde{}$the actual signal processing model is completed.

3.2.2 Results

Validation against simulation results

The results of the signal processing model in MATLAB were validated against simulation results in SPICE for an equivalent chip without noise or disturbances at a hypothetical magnetic field amplitude of 1 mT. Sensor designers demanded a max. deviation of no more than 5% as an acceptance criterium of this model. An overview of the validation results are shown in Table 3.5. The deviations of Bias voltage V_H, bias current I_B, current sensitivity s_I, Hall voltage V_H and the uncalibrated ADC output H_{ADC} between simulation results and MATLAB model were analysed at three different temperatures: cold (-40°C), room temperature (RT $+25$°C) and hot ($+175$°C). The acceptance criterium was met for all temperature levels.

Table 3.5: Validation of results of MATLAB signal processing model versus simulation results. The relative deviation of the signal amplitude of bias voltage, bias current, current sensitivity, Hall voltage and ADC output is given at three temperatures: ”cold” (-40°C), ”room temperature (25°C) and hot (175°C)

Temperature	Bias voltage	Bias current	Current sensitivity	Hall voltage	ADC output
°C	%	%	%	%	%
−40	−0.30	−0.20	−0.12	−0.52	−0.35
+25	1.02	0.90	0.09	0.85	1.10
+175	1.87	1.80	0.53	1.80	1.96

Hall and ADC behaviour versus temperature

Influence of temperature to non-calibrated results of the signal processing path is illustrated in Figure 3.11.Signal strength of the Hall voltage V_H can decrease by as much as 50%. Temperature compensation can significantly improve the performance of the ADC values H_{cal}over temperature. Choosing the calibration parameters more carefully may even increase the performance of the calibrated ADC outputs to deviations ranging down to single LSB’s.

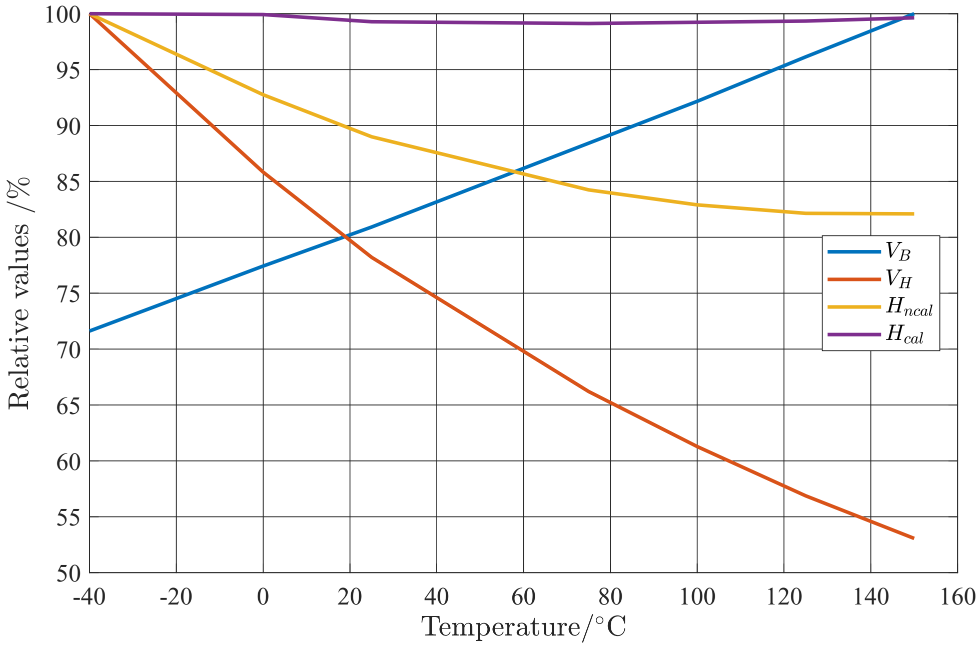

Figure 3.11: Behaviour of Hall circuits (bias voltage V_B and Hall voltage V_H) and non-calibrated ADC output (H_{ncal}) versus temperature. Additionally a temperature compensated ADC output curve (H_{cal}) was plotted to illustrate the capabilities of temperature compensation

Results of the signal processing path

The signal processing model was applied to results of the magnetic field calculation of section 3.1.3 (the assembly tolerances for the magnetic field calculation were chosen according to Run 2 of Table 3.3). Using a single biased circuit causes full correlation in the biasing of each Hall plate, eliminating the error caused by uncertainties.

The non-calibrated results are presented in Figure 3.12. No deterministic disturbances and offsets were added in the signal processing path. Due to this fact the angle error $\Delta\varphi$ is identical to Figure 3.4.

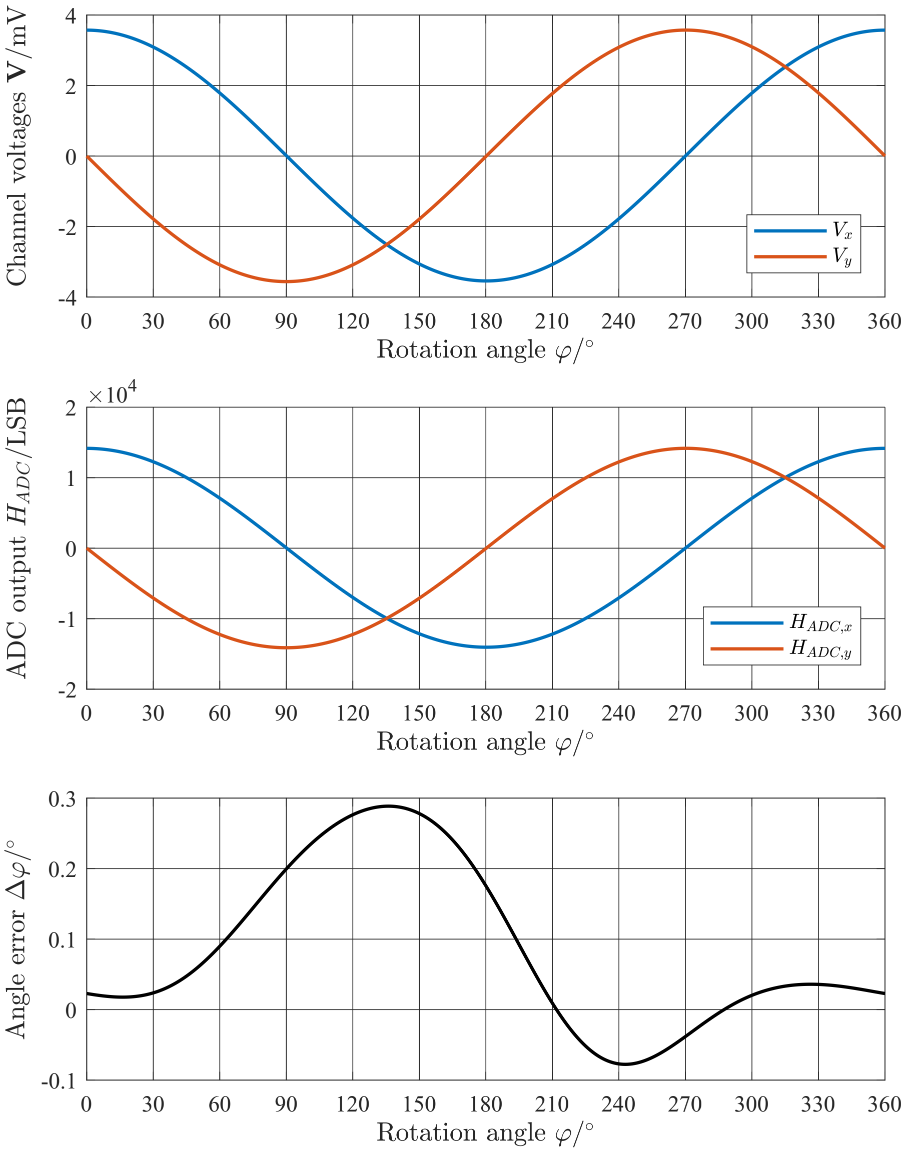

Figure 3.12: Results of the signal processing path with magnetic fields perturbed by assembly tolerances according to Run 2 of Table 3.3 **Top Plot:** Hall channel voltages V_x, V_y **Center Plot:** Uncalibrated ADC output $H_{ADC,x}$, $H_{ADC,y}$ and **Bottom Plot:** Angle error $\Delta\varphi$ for a full shaft rotation $\varphi = [0, 360]^\circ$

A calibration has not been performed on this set of data. In the following section, the influence of statistical uncertainties of the signal processing path will be illustrated using the GUM approach (`unc`) and the Monte Carlo method (`unc_t`) as validation for GUM.

3.2.3 Datapath modelling with AUT

All electrical components (resistances, transistors, Hall plates) are perturbed with statistical uncertainties. In the current state they are fully correlated for each Hall plates' signal path. This causes cancellation effects in the differential result. In fact the only uncertainty detected in the angle error result is the thermal noise.

GUM approach

The results of the signal processing path using the GUM based uncertainty propagation is shown in Figure 3.13.

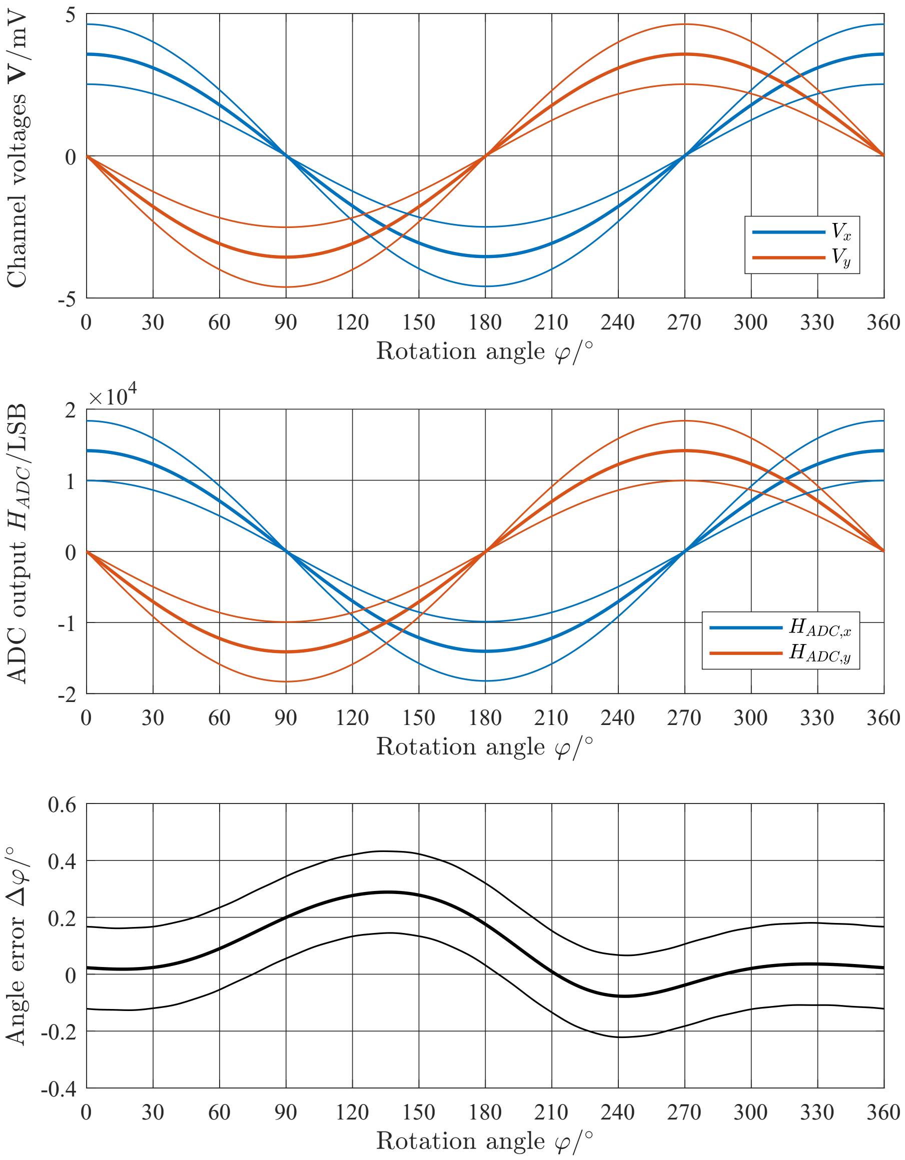

Figure 3.13: Results of the signal processing path modelled with GUM uncertainty modelling. Calculation of magnetic fields was perturbed by assembly tolerances according to Run 2 of Table 3.3 **Top Plot:** Hall channel voltages V_x, V_y **Center Plot:** Uncalibrated ADC output $H_{ADC,x}$, $H_{ADC,y}$ and **Bottom Plot:** Angle error $\Delta\varphi$ for a full shaft rotation $\varphi = [0,360]°$. The thin lines show the $\pm 3\sigma$ limits of the uncertain result. Computation time was 400 s.

Monte Carlo method

The identical signal processing path was used to verify the applicability of the GUM based AUT toolbox on signal process modelling. The results of the Monte Carlo based (`unc_t`) approach are shown in Figure 3.14.

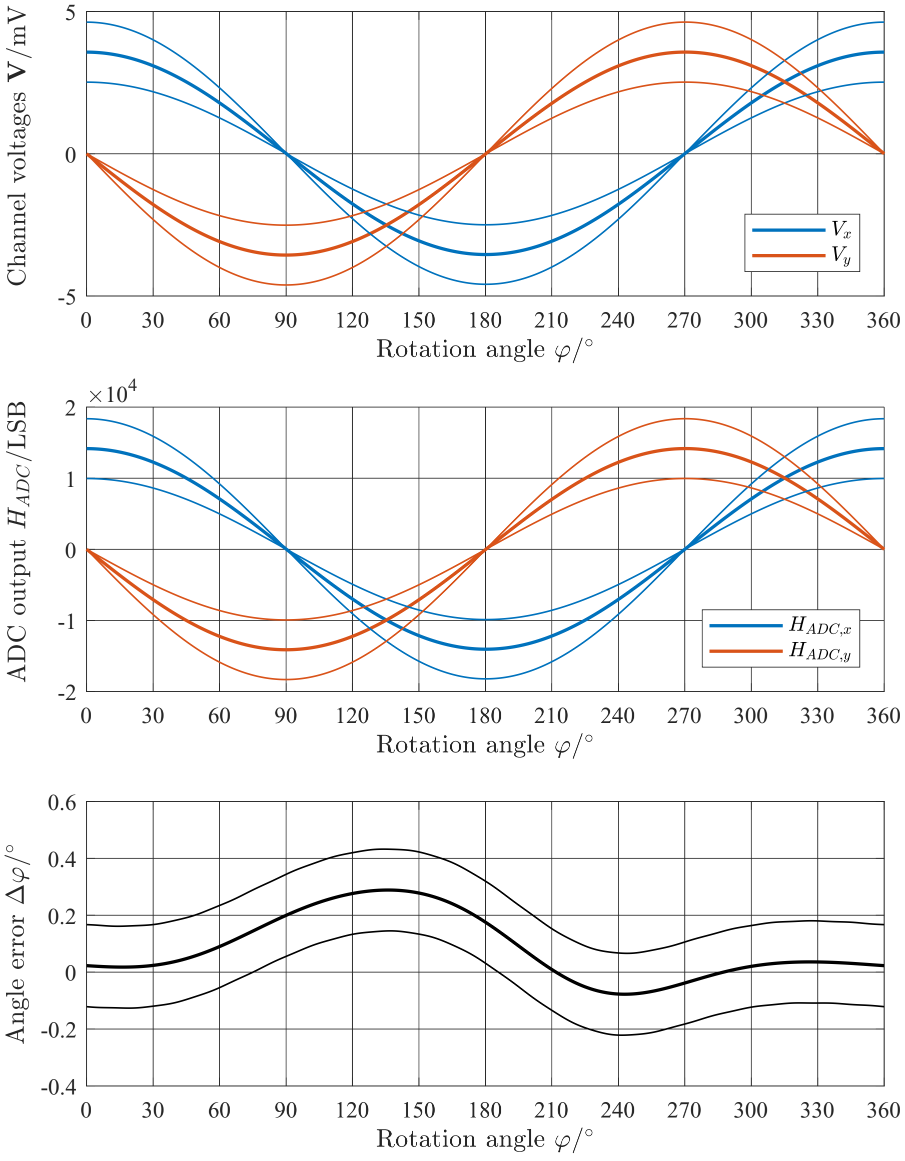

Figure 3.14: Results of the signal processing path modelled with Monte Carlo simulations. Calculation of magnetic fields was perturbed by assembly tolerances according to Run 2 of Table 3.3 **Top Plot:** Hall channel voltages V_x, V_y **Center Plot:** Uncalibrated ADC output $H_{ADC,x}$, $H_{ADC,y}$ and **Bottom Plot:** Angle error $\Delta\varphi$ for a full shaft rotation $\varphi = [0,360]°$. The thin lines show the $\pm 3\sigma$ limits of the uncertain result. Computation time was 50 s.

A comparison of the results of both approaches presented in Figures 3.13- 3.14 shows no discrepancy in the results for Hall channel voltages, ADC outputs (uncalibrated) and in the angle error result $\Delta\varphi$. No indication for inapplicability of the GUM approach (`unc`) was found during the investigations made.

3.2.4 Calibration

In addition the calibration algorithm was run to illustrate that the systematic errors are eliminated after calibration. The calibrated curve of the angle error curve (shown in Figures 3.13 and 3.14) is shown Figure 3.15. The statistical portion of the remaining angle error $\Delta\varphi_{cal}$ is $\sigma \approx 0.05°$.

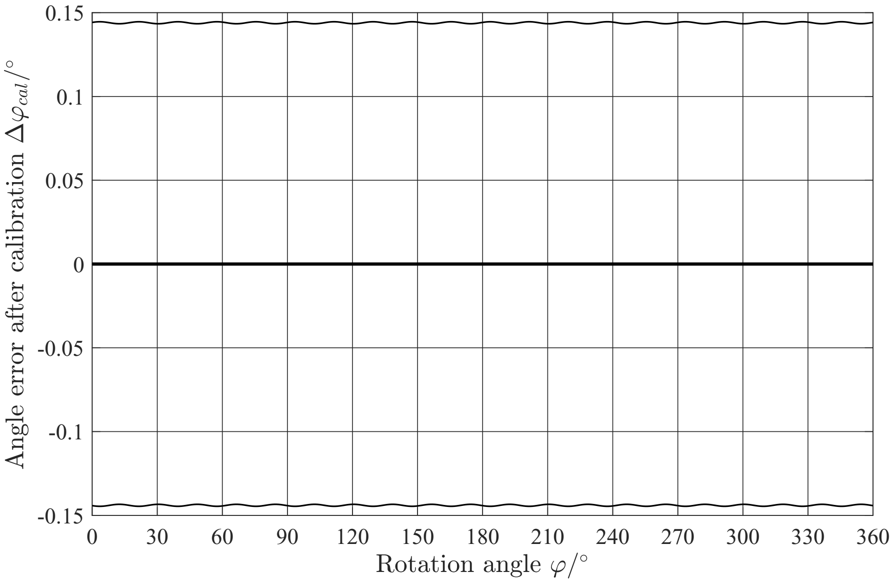

Figure 3.15: Calibrated angle error curve $\Delta\varphi_{cal}$ of the error curves shown in Figures 3.13 and 3.14. The systematic portion of the angle error is cancelled out, with the random portion of the remaining angle error. This angle error has a standard uncertainty of $\sigma \approx 0.048°$. The thin lines show the $\pm 3\sigma$ limits of $\Delta\varphi_{cal}$

3.2.5 Further remarks

This thesis aims to develop a fair model of the analogue and digital circuits of an integrated gradient field based magnetic field sensor and performing uncertainty propagation to the model. With respect to statistics this model is not yet finished as it can be modelled far more in detail such as considering more components and defining correlations between elements of the signal processing path as well. Such an analysis is time consuming and is outside the scope of this thesis.

Cross-checks of interim results by more complex models (compare Table 3.5) have shown, that even such an simple model yields decent results. An increase in complexity will likely improve the results of the model further but by much smaller means. But the effort in model creation, which changes from sensor device to sensor device will harm the versatility of this model on the other hand.

As a final remark: This model is applicable for early design phases and uncertainty propagation through the signal path. It answers questions such as: In what range will my expected angle errors lie in? Which input parameter affects the uncertainty of the angle error result most? Which parameter has little or no effect on the angle error result?
This model does not replace extensive simulation models when looking at precise sensor designs.

3.3 Kalman filtering

3.3.1 Filter design

A schematic overview of the Kalman Filter design is shown in Figure 3.16. The filter algorithm can be divided into two steps: The first of which being the initialization process, where the Kalman Gain is computed. The Kalman Gain itself does not depend on the individual measurements allowing its computation beforehand. The second step is the actual filter algorithm. This algorithm consists of the acquisition of the measurand $\varphi\tilde{}$and using the residual of the subtraction with the previously predicted estimate. This residual is then used to produce an estimate of the current state (angular position φ' and angular speed $\dot{\varphi}'$), as the measurement signal is still perturbed with noise. The Kalman Gain matrix **K** defines the weighting factors for the trustworthiness of the measurement and the model equations. Using state transition equations, a prediction of the current state is made for the upcoming one. Much care lies in the decision for the best fitting state space representation. This algorithm assumes a constant change of angular velocity over time yielding a simple and thus fast computable algorithm. The derivations of the formulas in use were made in section 2.3.3.

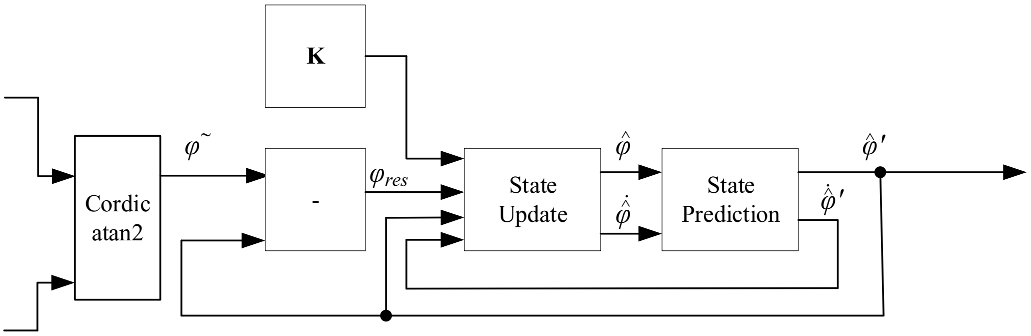

Figure 3.16: Schematic overview of the Kalman Filter algorithm. The Kalman Gain matrix **K** is computed during initialization and stored onboard. The recurring Kalman Filter algorithm only needs to perform the State Update and State Prediction. This reduces computation time down to $3-5$ μs.

Initialization

The initialization step is itself a loop, which computes the Kalman Gain matrix **K** until it converges. This loop can be computed during engine start-up. In this way no computing power of the microchip is needed during driving. Let's recall equation (2.31) for the Kalman Gain matrix **K** as given by,

$$\mathbf{K} = \begin{bmatrix} K_{\varphi} \\ K_{\dot{\varphi}} \end{bmatrix} = \frac{\mathbf{P}'\mathbf{H}^T}{\mathbf{H}\mathbf{P}'\mathbf{H}^T + \mathbf{R}}. \tag{3.42}$$

with **H** being the measurement matrix showing the observables. Only the current angular position is observed in this angle sensor system, yielding a **H** of:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{3.43}$$

The prediction of the covariance matrix $\mathbf{P}'$ (here a 2×2 matrix) has to be assumed for initial iteration. It is then calculated by Equation (3.46).
Finally, **R** is the covariance matrix of the measurement noise. This matrix is actually just a scalar showing the variance of the remaining angle error after calibration, as it is again the only observable in this system. It is given by,

$$\mathbf{R} = var(\Delta\varphi). \tag{3.44}$$

The estimate of the covariance matrix **P** is calculated using the prediction from the previous iteration and the weighting factors of **K**. **P** is given by,

$$\mathbf{P} = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}'. \tag{3.45}$$

The predicted covariance for the upcoming iteration is then calculated by applying the state transition matrix **F** to the current estimate **P** and adding the process noise matrix **Q** given by,

$$\mathbf{P}' = \mathbf{F}\mathbf{P}\mathbf{F}^{\mathbf{T}} + \mathbf{Q}. \tag{3.46}$$

with **F** being the state transition matrix (also defined in Equation (2.36)) given by,

$$\mathbf{F} = \begin{bmatrix} 1 & dT \\ 0 & 1 \end{bmatrix}. \tag{3.47}$$

where dT denotes the sample time.

Information on the trustworthiness of the state space equations is stored in the process noise covariance matrix **Q**. Increasing values in this process noise matrix (decreasing confidence in the state transition equations) causes an increase of the Kalman Gain matrix, which indicates that the measurements receive higher weighting. Assuming that the angular speed $\dot{\varphi}$ may change over time, **Q** might look like this:

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 \\ 0 & \dot{\varphi} \end{bmatrix}. \tag{3.48}$$

Unfortunately, the microcontrollers in use do not perform matrix operations. All equations in matrix notation need to be re-formulated in row-wise notation. A MATLAB code snippet for the row wise Kalman Filter initialization is provided in Appendix B.1.

Loop

Once the initializing is completed, the actual filtering process can begin. Only the entries of the Kalman Gain matrix $\mathbf{K} = [K_{\varphi}, K_{\dot{\varphi}}]^T$ are needed. The derivations for the State Update Equations and State Prediction Equations have already been made in section 2.3.3. The State Update Equations (consider Equations (2.41,2.42)) are

$$\begin{aligned} \hat{\varphi} &= \hat{\varphi}' + K_{\varphi}\, \varphi_{res}, & (3.49)\\ \hat{\dot{\varphi}} &= \hat{\dot{\varphi}}' + K_{\dot{\varphi}}\, \varphi_{res}, & (3.50) \end{aligned}$$

with φ_{res} being the residual of the measurement and the prediction of the model (consider Equation (2.44)), which is given by,

$$\varphi_{res} = \varphi^{\tilde{}} - \hat{\varphi}'. \qquad (3.51)$$

Note: The residual angle φ_{res} is not to be confused with the angle error $\Delta\varphi$ ($\Delta\varphi = \varphi^{\tilde{}} - \varphi$).

In analogy to Equations (2.45,2.46) the State Prediction are to be as

$$\begin{aligned} \hat{\varphi}' &= \hat{\varphi} + dT\,\hat{\dot{\varphi}}, & (3.52)\\ \hat{\dot{\varphi}}' &= \hat{\dot{\varphi}}. & (3.53) \end{aligned}$$

Note: A Matlab Code snippet of this Kalman Filter loop is provided in Appendix B.2.

Filter tuning

The initialization algorithm allows some fine tuning to the filter's properties. The measurement noise matrix **R** cannot be modified as it defines the variance of the measurement noise. The best way to tune the filter is to adjust the entries of the process noise variance matrix **Q**. Higher entry values in the process noise matrix will cause the filter to put more faith on the measurements. This means that the filter will primarily work as a predictor. Lower valued entries in **Q** put a higher weight on the model equations. The filter will not be able to follow rapid changes of the measurement signal and will primarily work as a smoother. A better set up of initial values has little effect on the filter results in general, as a proper tuned filter will quickly catch up to the true angle value φ.

3.3.2 Filter example application: Angle sensing for motor commutation

In this section the principle of Kalman Filtering will be illustrated using a possible application in automotives: The commutation of the motor for electric power steering (EPS). EPS receives sensor data (motor speed, current angle position) from the motor [89].

Some technical data of this application are:

1. max. acceleration rate $\ddot{\varphi}_{max} = 2.4 \times 10^6\ ^\circ/\mathrm{s}^2$
2. max. rotation speed $n_{max} = 10{,}000$ rpm (or $\dot{\varphi}_{max} = 60{,}000^\circ/\mathrm{s}$)
3. sample interval time $dT = 25\ \mu\mathrm{s}$

4. standard uncertainty of noisy signal $\sigma_{noise} = 0.05^\circ$ (after all systematic errors are calibrated, consider also Chapter 4 of this thesis)

The value for $\ddot{\varphi}_{max}$ looks astonishingly high at first sight, but when this acceleration rate is converted to the sample time domain, $\ddot{\varphi}_{max}$ remains only $0.0015^\circ/dT^2$ which is equal to an increase of 7.5×10^{-4} ° per sample dT. In comparison to σ_{noise} the acceleration suddenly looks irrelevant.

Using this technological data, a simple speed ramp was modelled in Matlab. This ramp is shown in Figure 3.18.

When looking at sampling intervals as small as $dT = 25\ \mu$s, the time needed for data processing and transmission reaches the same order of magnitude as dT and cannot be neglected any more. This causes a delay by as much as 50 μs (compare Figure 3.17), which is roughly the equivalent of a delay of two sample intervals.

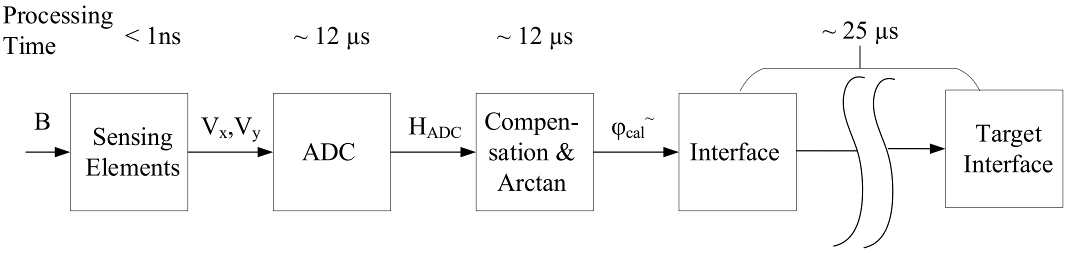

Figure 3.17: Time consumption for signal processing and transmission. The calculation of the ADC output takes approximately 25 μs. Signal transmission to the target interface may take up to 25 μs too. This yields to a total delay of 50 μs.

This delay causes angle errors proportional to the rotation speed of the shaft (compare Figure 3.19).

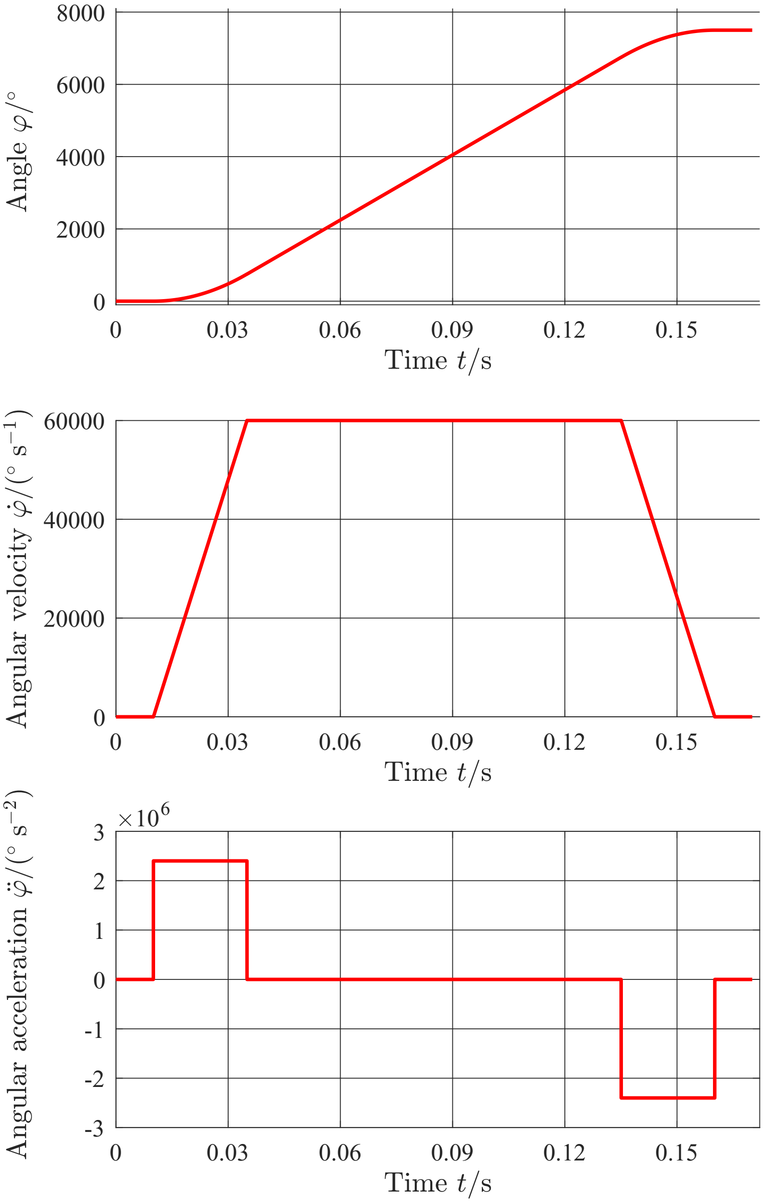

Figure 3.18: Exemplary speed ramp for Kalman Filter testing. Technological data is in accordance with the data presented above. **Top Plot:** Rotation angle φ during this exemplary ramp. **Center Plot:** Course of the angular velocity $\dot{\varphi}$. **Bottom Plot:** Course of the angular acceleration $\ddot{\varphi}$.

This simple ramp shown above is made up of five different states. An idle state is followed by a rapid acceleration by $\ddot{\varphi}_{max}$ until $\dot{\varphi}_{max}$ is reached. The speed $\dot{\varphi}_{max}$ is kept constant for the following 100 ms. The motor is then de-accelerated by $-\ddot{\varphi}_{max}$ to idle state, in which it remains for another 10 ms.

The process noise variance chosen, allows an increase (or decrease) of the angular velocity by $\dot{\varphi}_{max}$ for each sample. It is given in matrix notation by,

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 \\ 0 & (dT\ \dot{\varphi}_{max})^2 \end{bmatrix}. \tag{3.54}$$

3.3.3 Results

The results of this Kalman Filter (angular position, angular velocity and angular acceleration) example are shown in Figures 3.19 to 3.21. The filter manages to predict highly accurate results as long as the physical behaviour of the actual system is equal to the model algorithm.

Angle error caused by the delay of the signal processing path are compensated by the predictor of the Kalman filter in the states of constant velocity. During both states of constant acceleration, a static lag error of 18 sample intervals is produced, as the state model equations are not capable of detecting quadratic changes of the angular position.

Furthermore, the Kalman Filter also smooths the noisy data quite well (compare Figure 3.19). After a state of constant velocity is reached, the filter is adjusting back close to the true angle values very fast.

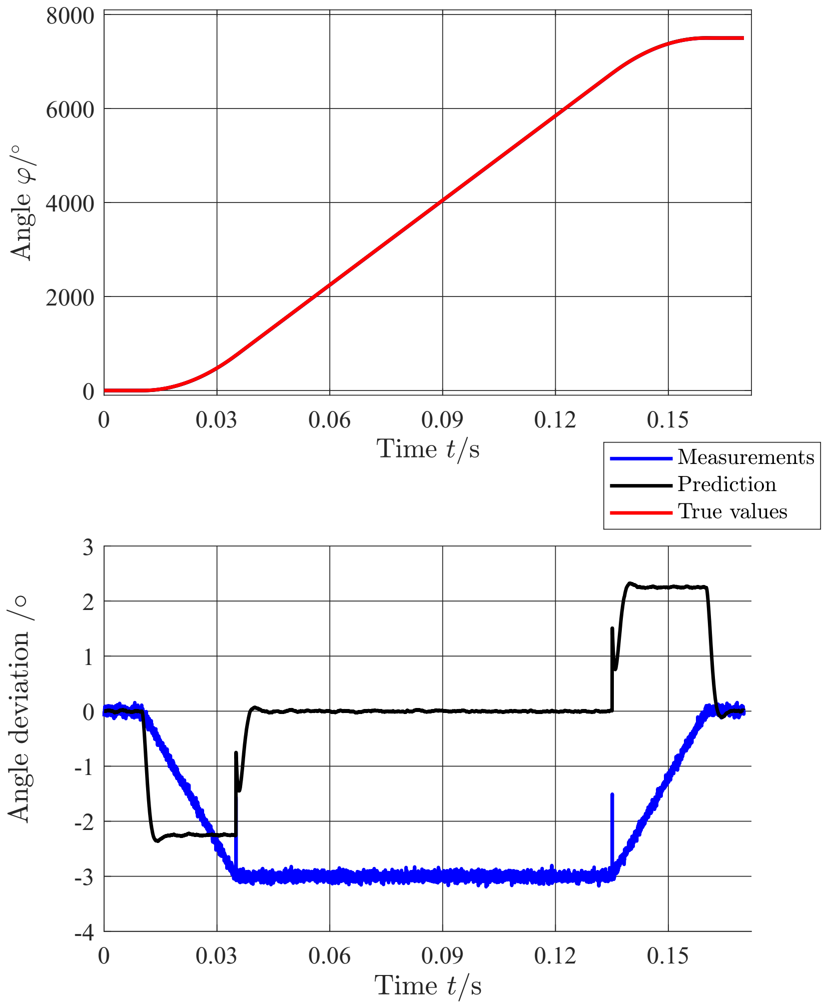

Figure 3.19: Angular position results of the Kalman Filter example: ”Motor commutation”. **Upper Plot**: Absolute angular position values of the true values, measurements and Kalman Filter predictions. **Lower Plot**: The deviation of the angular position with respect to the true position for the measurements and Kalman Filter prediction, respectively.

Figure 3.20 shows the true values and the Kalman Filter predictions of the angular speed. The rapid change in angular acceleration causes a static lag. This lag is cancelled out once the acceleration is again zero.

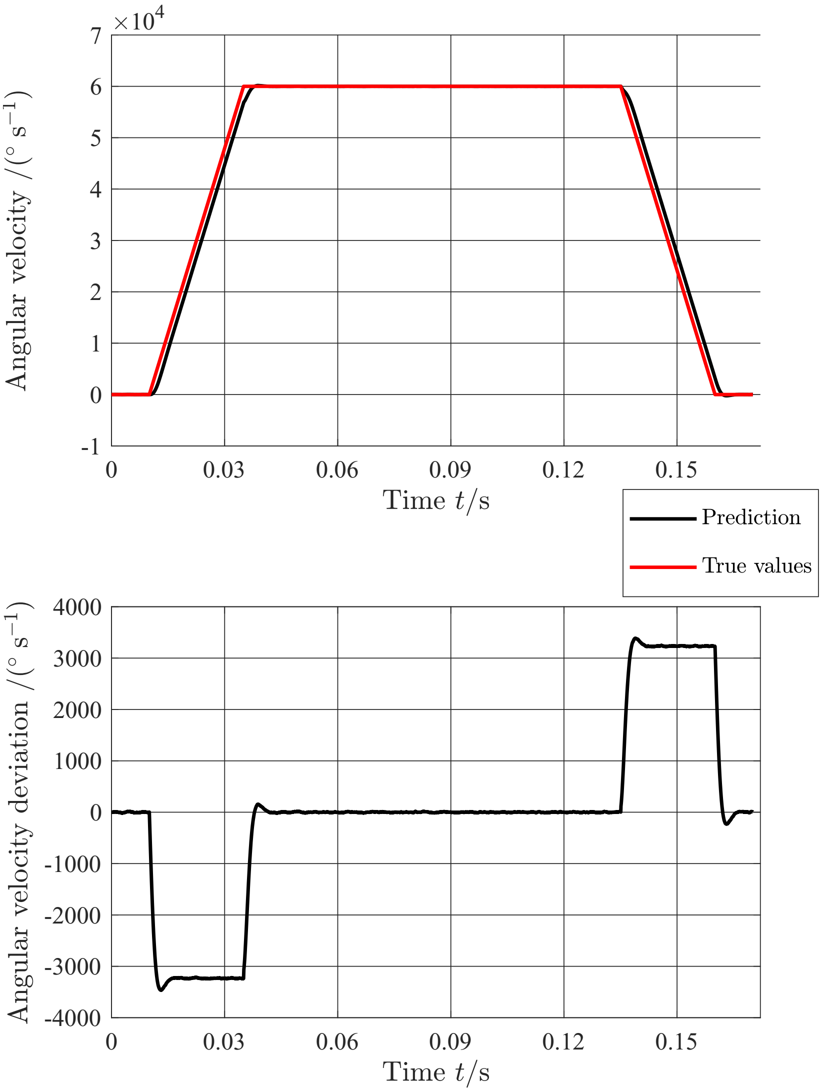

Figure 3.20: Angular velocity results of the Kalman Filter example: ”Motor commutation”. **Upper Plot**: Angular velocity values of the true values and Kalman Filter predictions. **Lower Plot**: The deviation of the angular velocity with respect to the true velocity for the Kalman Filter prediction.

Figure 3.21 shows the true values of the angular acceleration and an approximation of the predicted angular acceleration of the Kalman Filter, as the prediction of the acceleration is not calculated by the filter algorithm. The approximation of the predicted angular acceleration is calculated by,

$$\ddot{\varphi}'(i) = \frac{\dot{\varphi}'(i+1) - \dot{\varphi}'(i)}{dT}. \tag{3.55}$$

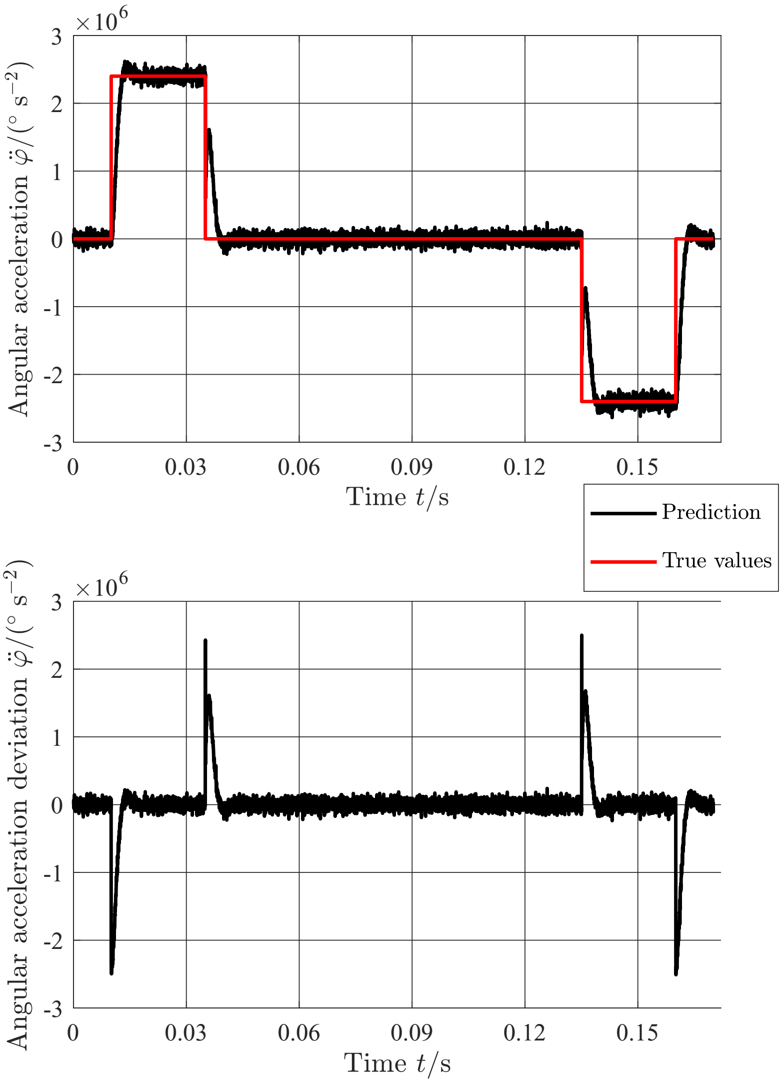

Figure 3.21: Angular acceleration results of the Kalman Filter example: ”Motor commutation”. **Upper Plot**: Angular acceleration values of the true values and estimations of Kalman filter predictions. The angular acceleration of the filter is obtained by calculating the increase in velocity between two velocity predictions and not calculated by the filter algorithm. **Lower Plot**: The deviation of the angular acceleration with respect to the true acceleration for the calculated Kalman Filter prediction.

A sudden change of the angular acceleration rate can not be handled by the filter producing the constant angular deviation in Figure 3.19. Nevertheless the filter is able to assume the accurate angular acceleration rate in after a lag period of 18 samples. By reverting the angular acceleration rate this deviation is cancelled out.

By adapting the entries of the process noise covariance matrix **Q** the filter could be adapted to better process these changes at the cost of losing the ability to smooth the noisy time series data. In extreme cases the noise level of the filter result can even be higher than the original noise level.

3.3.4 Kalman filtering with AUT

In this section, the results of the applicability analysis of the Uncertainty Toolbox for Kalman Filter applications is analysed. Additionally, a Monte Carlo simulation was conducted as a means of verification.

The MATLAB code of the example code only needed to be altered slightly for uncertainty analysis. Gaussian distributed measurement noise can be added to the true angle signal by adding an uncertain variable with zero mean and standard uncertainty *R*:

```
% true angle value with time delay of two sample intervals
phi_delay
% standard uncertainty of the measurement noise
R
% measurement
z
%
for i=1:nrPoints
      z(i)=phi_delay(i)+unc(0,R);
end
```

Listing 3.2: How to add a Gaussian distributed measurement noise to an angle signal. A Monte Carlo simulation can be performed by exchanging `unc` by `unc_t`.

GUM approach

Unfortunately there are no results to present here. Due to the nature of the Kalman Filter algorithm being recursive, the uncertainty of every previous measurement has an effect on the uncertainty of the estimate at some time stamp t_S. Due to the linearisation effects, information of the uncertainty of each previous measurement is stored in the uncertainty property of the current measurement point. The uncertainty contribution of each previous measurement decreases obviously as time continues, but the model computes them nevertheless. This causes a rapid increase in computation time after just a comparatively small sample size.

One such run was started, though, but aborted after a computation time of 2 1/2 days.

A possible remedy would be to alter the portion of the uncertainty contribution calculation of an uncertain variable to keep only a fixed number of error contribution (e.g. keep the uncertainty contribution of ten previous measurements) or to define a limit under which the error contribution of the previous measurement will be deleted.

Monte Carlo simulation

Fortunately, the Monte Carlo simulation of the Kalman Filter example did produce results with reasonably computation time of approximately 50 s. The results for the angular position's prediction are shown in Figure 3.22. Additionally the velocity prediction is shown in Figure 3.23. Looking at the bottom plot of Figure 3.22 no huge changes to the corresponding Figure 3.19 are observable. In fact, they look almost perfectly alike. The noise level is now indicated by the thin lines representing the $\pm 3\sigma$ interval around the mean value. The uncertainty of the noise decreases by over a magnitude (the thin lines are too close to the thick mean value to be detected).

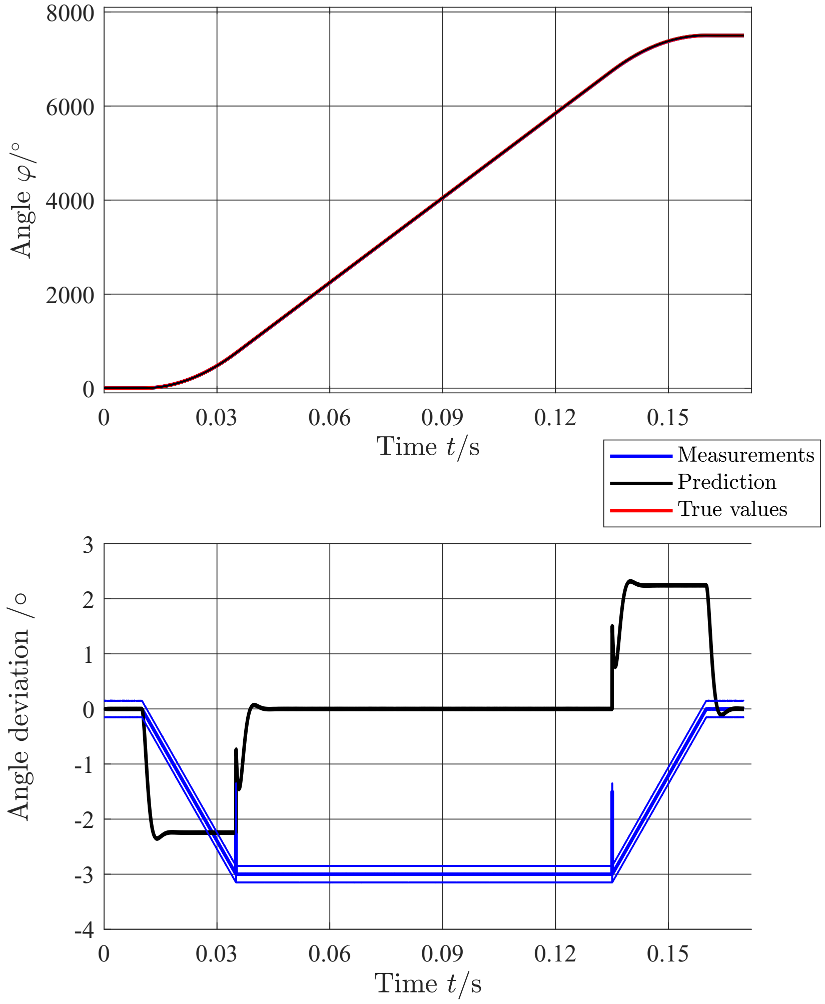

Figure 3.22: Angular position results of the Kalman Filter example: ”Motor commutation” with Monte Carlo simulation. **Upper Plot**: Absolute angular position values of the true values, measurements and Kalman Filter predictions. **Lower Plot**: The deviation of the angular position with respect to the true position for the measurements and Kalman filter prediction, respectively. The thin lines show the $\pm 3\sigma$ limits of the respective quantity. The uncertainty of the Kalman Filter prediction has decreased by one order of magnitude. 10,000 Monte Carlo samples taken.

Comparing the bottom plots of the corresponding angular velocity prediction (Figures 3.20 and 3.23, also no obvious alteration is observed. In fact both angular velocity deviation curves look alike. Like above, the thin lines should indicate the $\pm 3\sigma$ limits, but the uncertainty of the angular velocity deviation is too small to be observed.

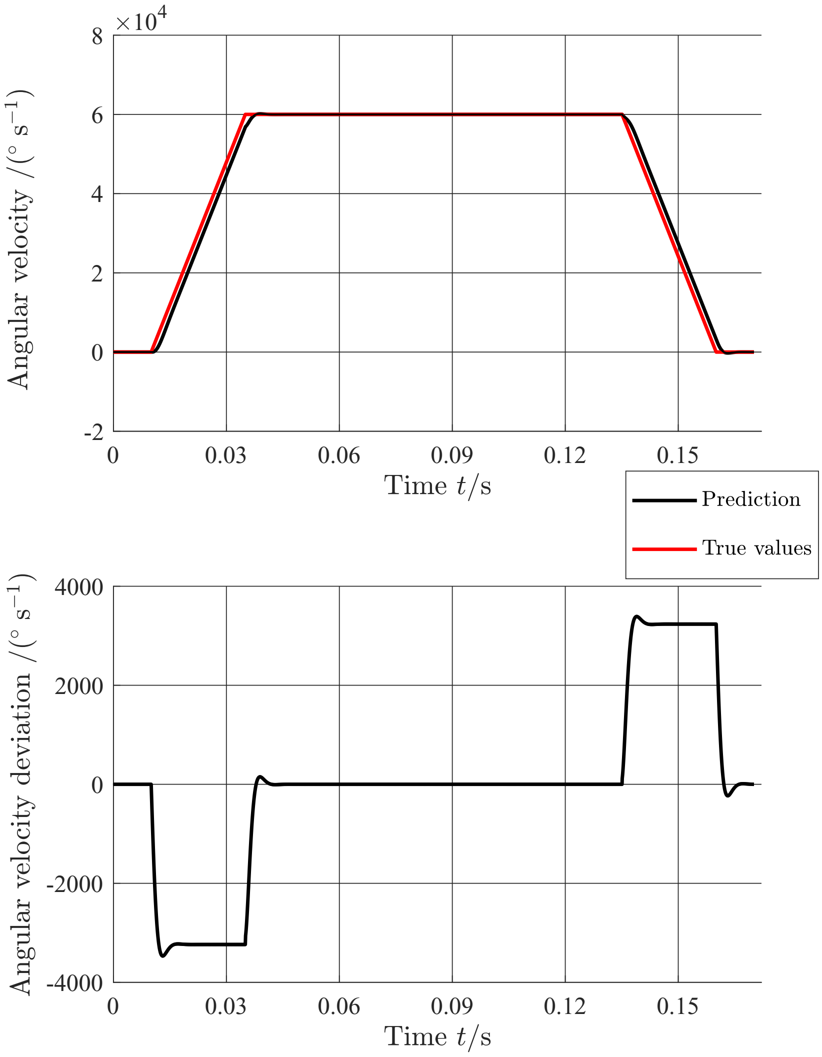

Figure 3.23: Angular velocity results of the Kalman Filter example: ”Motor commutation” with Monte Carlo simulation. **Upper Plot**: Angular velocity values of the true values and Kalman Filter predictions. **Lower Plot**: The deviation of the angular velocity with respect to the true velocity for the Kalman Filter prediction.The thin lines show the $\pm 3\sigma$ limits of the respective quantity. The uncertainty of the result is by far smaller than its mean value. 10,000 Monte Carlo samples taken.

Resumé

A Kalman Filter is a good method for both predicting and smoothing data. Angle errors caused by delays signal processing and transmission may reach values of up to $\Delta\varphi = 3°$. When the actual behaviour of the real system does not longer follow the coded behaviour a deviation is observable. This deviation vanishes as soon as this state is left. The uncertainty toolbox is not able to deploy results within reasonable computation time, but the Monte Carlo approach did perform well, and showed that the data is smoothed for 10,000 observed runs.

Chapter 4

Experimental Verification

In this chapter some measurement data, that has been gathered from test-chips are presented and evaluated to validate that the assumption made about propagation of Gaussian distributed angle errors is in fact valid. Furthermore the applicability of Kalman Filters relies on Gaussian distributed measurement data. Two different types of measurements were analysed. Section 4.1 looks at noise measurement results at various temperatures , Section 4.2 focusses on how to obtain measurement noise from angle measurements with a high portion of systematic errors.

4.1 Noise measurement

Noise measurements have been conducted on a test chip at four different temperatures in the range of $-40°$C up to $+150°$C. The sensor was kept in idle mode (no systematic change of output) . 990 samples were recorded for each run. The measurement result is given in digital values. Due to the step size of the digitalization, Gaussian behaviour of noise cannot be detected for this data sets. Nevertheless the data set allows to search for systematic behaviour in the noise measurement.
To show that this recorded noise is free of any systematic portion, the auto- covariance function was calculated for each run. This function computes the covariance between the noisy data and a delayed replica of that same data. If no systematic (periodic) noise contribution is present, the auto-covariance function will compute the variance at zero lag. In theory, the function should return zero co-variance at all other lags, or within some confidence bound for a finite series of data.

The results of the auto-covariance function are shown in Figure 4.1. The red line indicates the 95% confidence interval of indicating no auto-covariance for the given sample size.

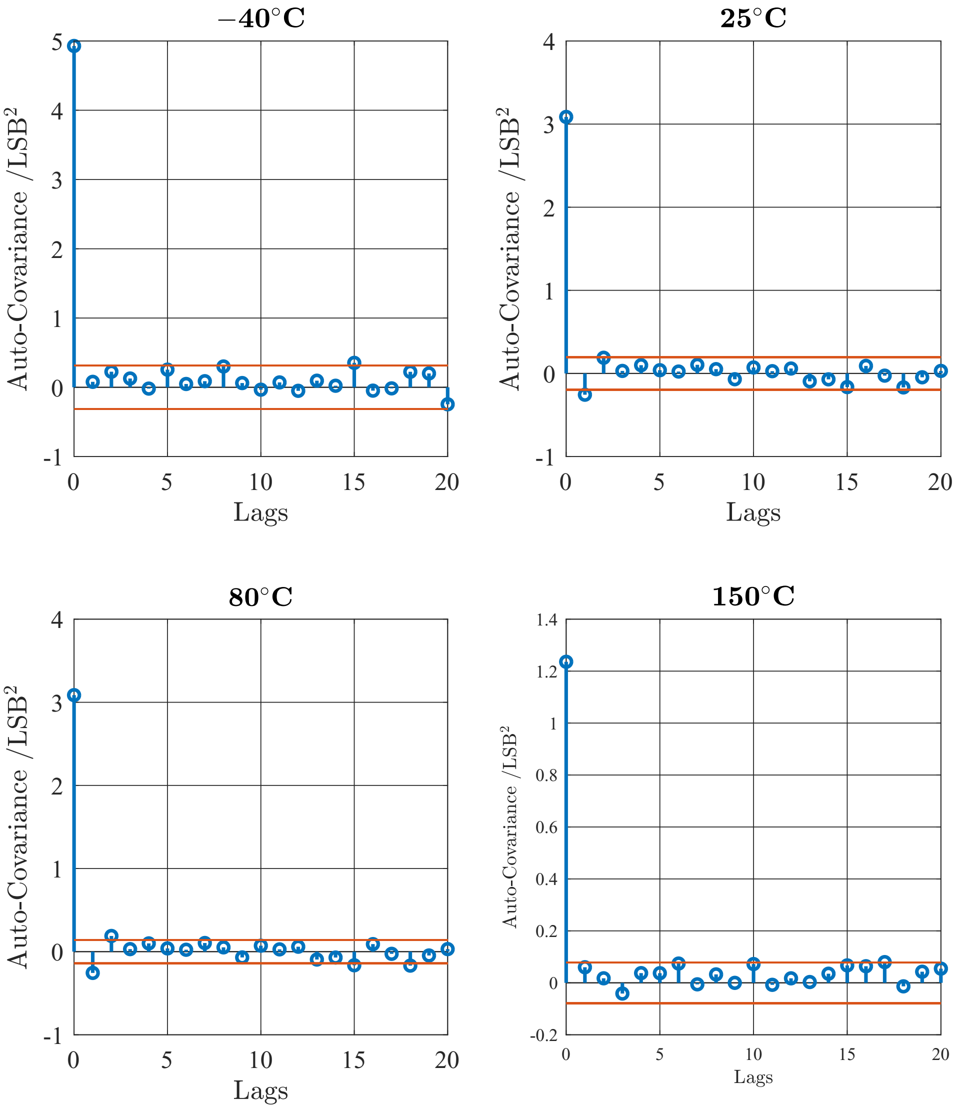

Figure 4.1: Auto-Covariance plots [90] of noise measurement runs of a test chip at −40°C, 25°C, 80°C and 150°C. The red lines indicate the 95% confidence interval bounds (equal to $\pm 2\sigma$).

4.2 Angle error measurements

For Experimental verification of the angle error results, a test chip was mounted on a test rig (as shown in Figures 4.2 and 4.3). This rig consisted of a shaft and a diametrically polarized disc magnet fixed to the shaft end. Multiple measurement runs of a full 360° rotation, with 136 samples recorded in each of the runs, were performed. Between each run, the sensor displacement $\mathbf{d_S} = [x_S, y_S, z_S]^T$ was altered using a precise coordinate table. All other Degrees of Freedom according to Table 3.2 remained unaltered but unknown in all runs. The test chip was not calibrated and all runs were conducted at room temperature ($RT = 23$°C).

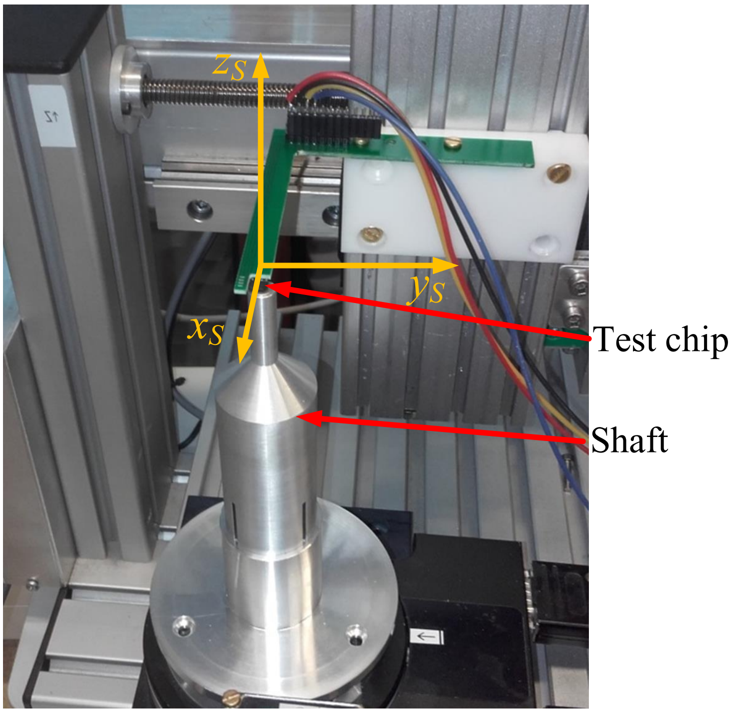

Figure 4.2: Overview of the test rig used for experimental verification. The sensor position ([x_S,y_S,z_S], compare Table 3.2) was altered for each measurement run using a highly precise coordinate table. For each run a full 360° rotation was recorded. Courtesy of Infineon Austria AG

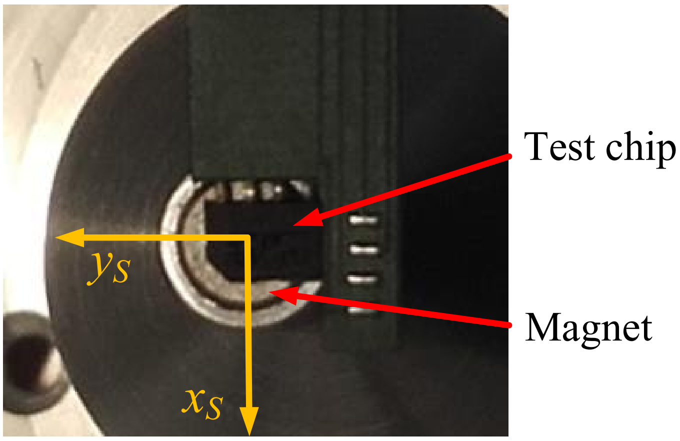

Figure 4.3: Top view of the measurement setup as shown in Figure 4.2. A diametrically magnetized disc magnet was mounted to the shaft end. The shaft performed a full 360° rotation, whereas the sensor remained fixed. Courtesy of Infineon Austria AG

Six representing runs (A to F; compare Table 4.1) were picked out of a set of measurement runs at different air gaps and different sensor positions x_S, y_S aiming to indicate the influence of sensor positioning tolerances to the angle error result. The reading radius *RR* was 1 mm. All runs were conducted at room temperature and only the input parameters given in Table 4.1 were changed. Despite this other assembly tolerances such as sensor tilts, magnet tilts and magnet misplacements were present but not changed for each run. The raw data for all runs are shown in Figure 4.4.

Table 4.1: Exemplary samples of the measurement runs conducted. A single test chip was used in all runs. Only the given parameters were changed. All other Degrees of Freedom (compare Table 3.2) were unknown but remained unaltered. *AG* air gap ($\hat{=}$ sensor displacement in z-direction), x_S sensor displacement in x-direction, y_S sensor displacement in y direction

Run	*AG*	x_S	y_S
	mm	mm	mm
A	0.8	0	0
B	0.8	- 0.15	0.15
C	1.1	0	0
D	1.1	0.15	0.15
E	1.4	- 0.15	0.15
F	1.7	0	0.15

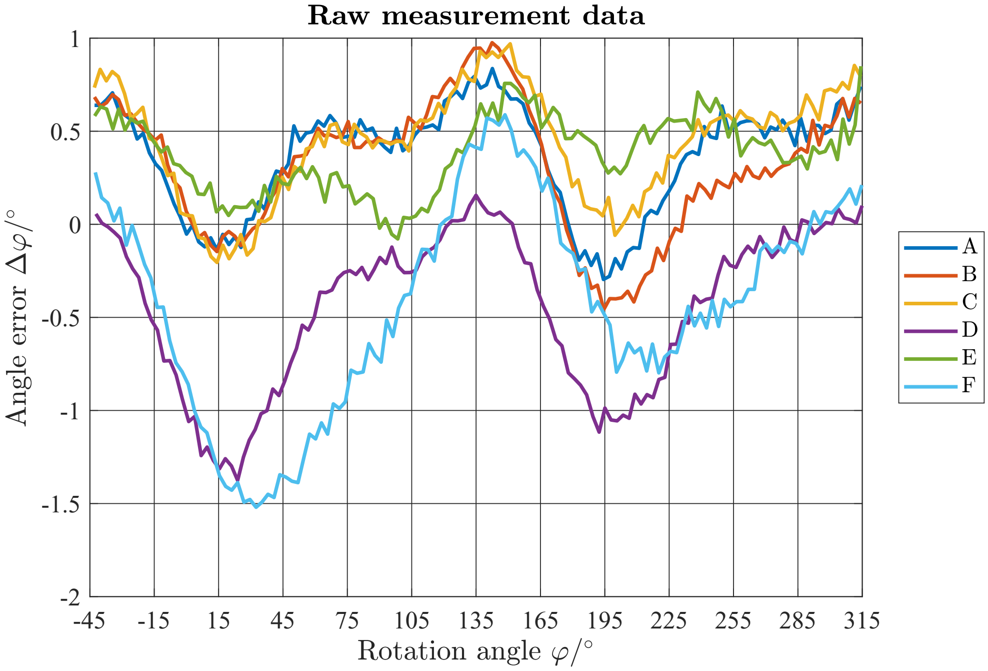

Figure 4.4: Uncalibrated angle error $\Delta\varphi$ over rotation angle φ of six representative runs A-F as given in Table 4.1. The contribution of systematic errors due to imperfect positioning of the magnet and sensor is the dominating portion of the error curves.

Systematic errors are the dominating portion of the angle error curves in Figure 4.4. The vertical shift of individual curves are caused by varying sensor position tolerances between each curve. In order to perform noise analysis this systematic behaviour has to be filtered from the data.

A well known approach to detect periodic behaviour in time series is Fourier Analysis. In Fourier Analysis a periodic function is described by a infinite sum of sine and cosine function. In practice the Fast Fourier Transformation (FFT) is performed making advantage of the basis functions' structure. It is theoretically identical to a Fourier transform, where all components of the Fourier series are computed. The first and last eight elements of the spectrum were cut off. The following code snippet shows how the periodical error portion can be elliminated from the angle error curve using FFT:

```
% true angle values/ mechanical angle
phi
% Angle Error (raw measurement)
dPhi
% Compute the Full Spectrum using FFT
fullSpec=fft(dPhi);
% Cut the spectrum
cut=8;
fullSpec(cut:end-cut+1)=0;
% Synthesize the filtered data (inverse Fourier Transform)
fftE=real(ifft(fullSpec));
% Compute the residual
fftR=dPhi-fftE;
```

Listing 4.1: MATLAB code to compute the residual angle error with Fast Fourier Transformation (FFT)

Only periodic portions of the systematic error contributions can be filtered with Fourier transform. An alternative solution is applying polynomial approximation of the data. Given the large sample count in the data ”Discrete Orthogonal Polynomials” (DOP) are a good approach for extracting systematic behaviour with polynomials of high order (the polynomial order chosen in the computations was ”20”). The Discrete Orthogonal Polynomials were calculated using DOPBox [91]. A MATLAB code snippet showing how the polynomial approximation of the systematic error contribution is modelled using DOPBox is shown below:

```
% true angle values/ mechanical angle
phi
% Angle Error (raw measurement)
dPhi
% Define degree of polynomial
degree=20;
% Compute the basis functions using DOPBox
B=dop(phi,degree);
% Compute the approximation
dopE=B*B'*dPhi;
%Compute the residual error
dopR=dPhi-dopE;
```

Listing 4.2: MATLAB code to compute the residual angle error with polynomial approximation of the systematical angle error portion using DOPBox

The results of both the FFT and the DOP model are shown for Run C in Figure 4.5. The results for all other runs (A-B,D-F) can be found in Appendix C.1 of this thesis.

The residual errors $\Delta\varphi_{res}$ (`fftR` for the FFT and `dopR` for the DOP model) are additionally plotted. The remaining residual error appears of statistical nature and it is one magnitude smaller than the angle error of the raw measurement data.

The standard uncertainty of the residual error σ_{res} was computed for each run A-F. The values for σ_{res} lie in a range of $\approx 0.05 - 0.06^{\circ}$. This coheres with the assumption made in the Kalman Filter example in section 3.3.2.

The residual error curves are then analysed regarding their auto-correlation, do determine residual systematic error behaviour. Additionally the auto-correlation was also computed of the raw measurement data and an exemplary Gaussian distribution for better comparison of the curves. This is shown for Run C in Figure 4.6. The auto correlation functions for all other measurement runs are shown in Appendix C.1.

The auto-correlation of the raw data is evident (compare upper left plot of Figure 4.6) leaving no doubt in clear systematic error behaviour. The auto-correlation of the residual errors of the FFT and DOP model clearly indicate that this systematic portion has vanished. A slight periodicity is still detectable as in the auto-correlation of the exemplary Gaussian function plotted in the bottom right corner of 4.6. Such periodicities are inevitable for Gaussian distributions of a finite sample size.

Next, the Fourier Spectrum of the raw measurement error, of DOP and FFT residual error was generated and compared to that of a Gaussian distributions. This is shown in Figure 4.7 again for Run C (the results for all other runs can again be found in Appendix C.1). Considering the raw error distribution in Figure 4.5 we assume that the strongest error contribution is that of 2^{nd} periodical order. This is validated by the indistinguishable peak in the Fourier Spectrum at (Upper left plot in Figure 4.7.

Gaussian distributions on the other hand, should appear as white noise in the Fourier spectrum in theory. The standard deviation of the Gaussian distribution does not converge causing a slight periodicity still detectable (compare bottom right plot of Figure 4.7 and also the auto correlation plot in Figure 4.6). Periodicity in high period orders can also be detected in the spectra of the DOP and FFT model (upper right and bottom left plots of Figure 4.7), respectively. The periodicity of low order terms, which dominated the error contribution of the raw measurement, are almost cancelled out entirely.

The first order period corresponds to offset errors, the dominating second order period arises from orthogonality and amplitude errors and the third order period is a mere result of non-linear effects. They are the dominating error sources in the raw angle error data.

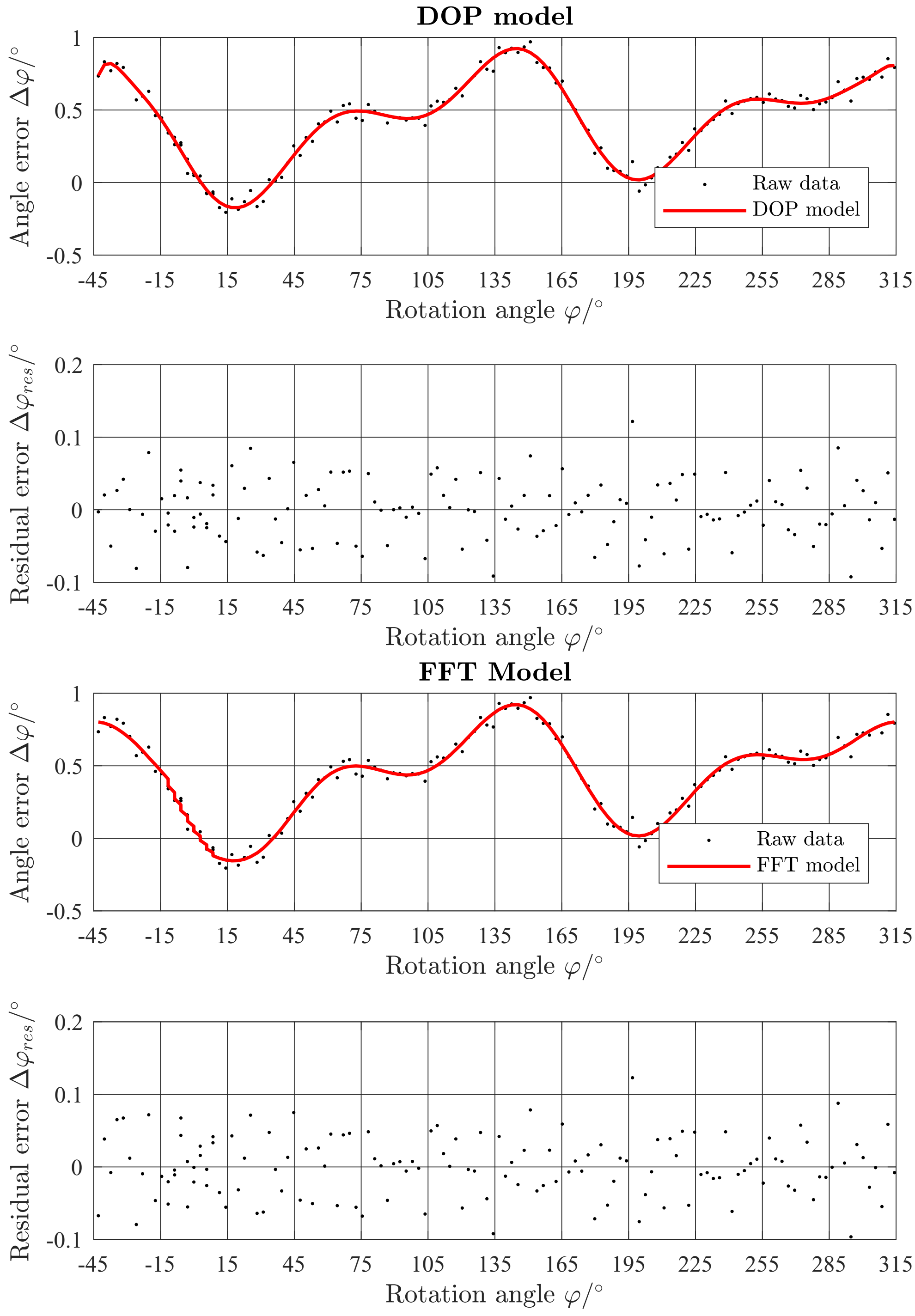

Figure 4.5: Results of the DOP and FFT filter models for Run C (assembly tolerances according to Table 4.1) . **Top Plot:** Uncalibrated angle error $\Delta\varphi$ and DOP filter model results over rotation angle φ. **Upper Central Plot:** Resulting residual error $\Delta\varphi_{res}$ of DOP model to raw data. **Lower Central Plot:** $\Delta\varphi$ and FFT filter model results over φ. **Bottom Plot:** $\Delta\varphi_{res}$ of FFT model to raw data.

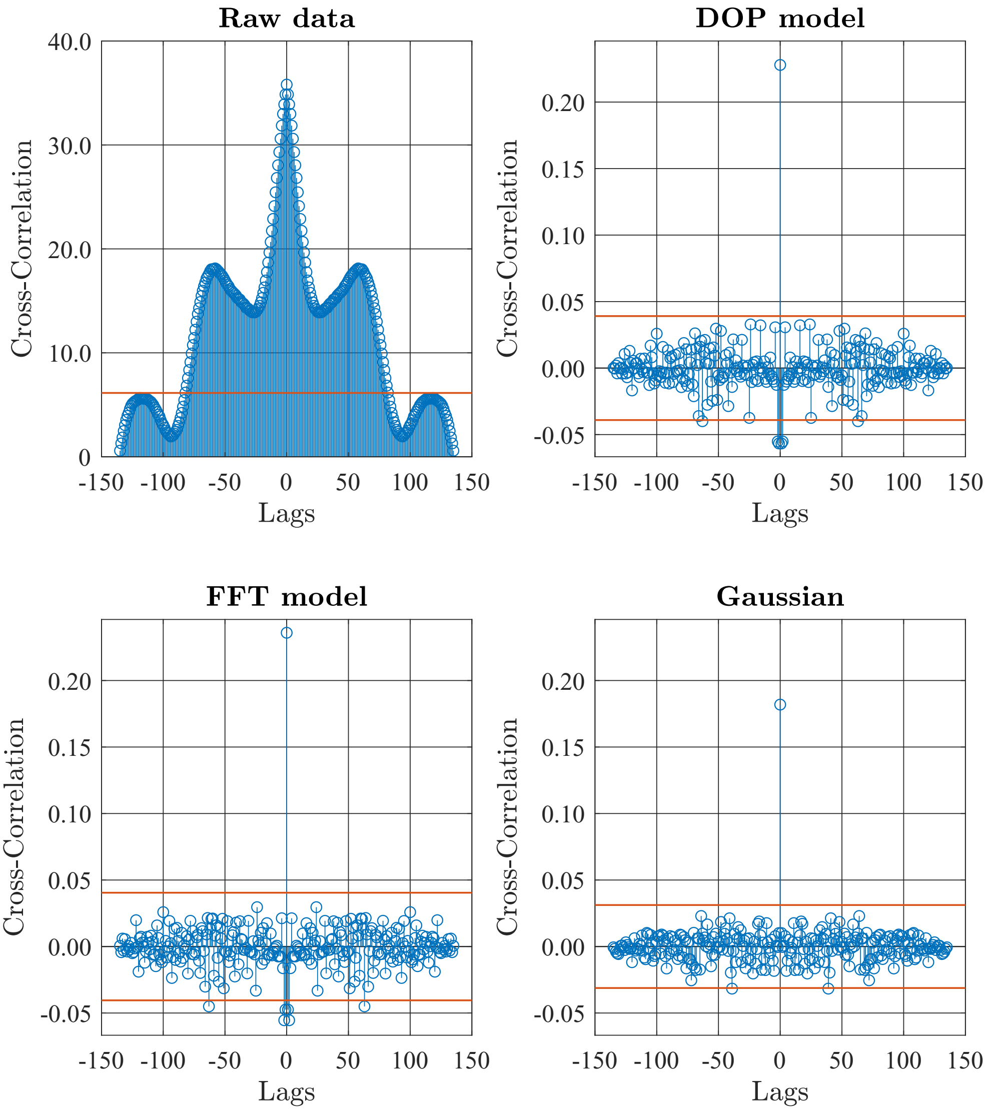

Figure 4.6: Cross-Correlation functions for Run C (assembly tolerances according to Table 4.1) of **Lower Left Plot:** raw measurement data, **Upper Right Plot:** Residual angle error of DOP filter model, **Lower Left Plot:** Residual angle error of FFT filter model and **Bottom Right Plot** of a Gaussian distribution with the same sample size. The red lines indicate the 2σ confidence interval for the given sample size.

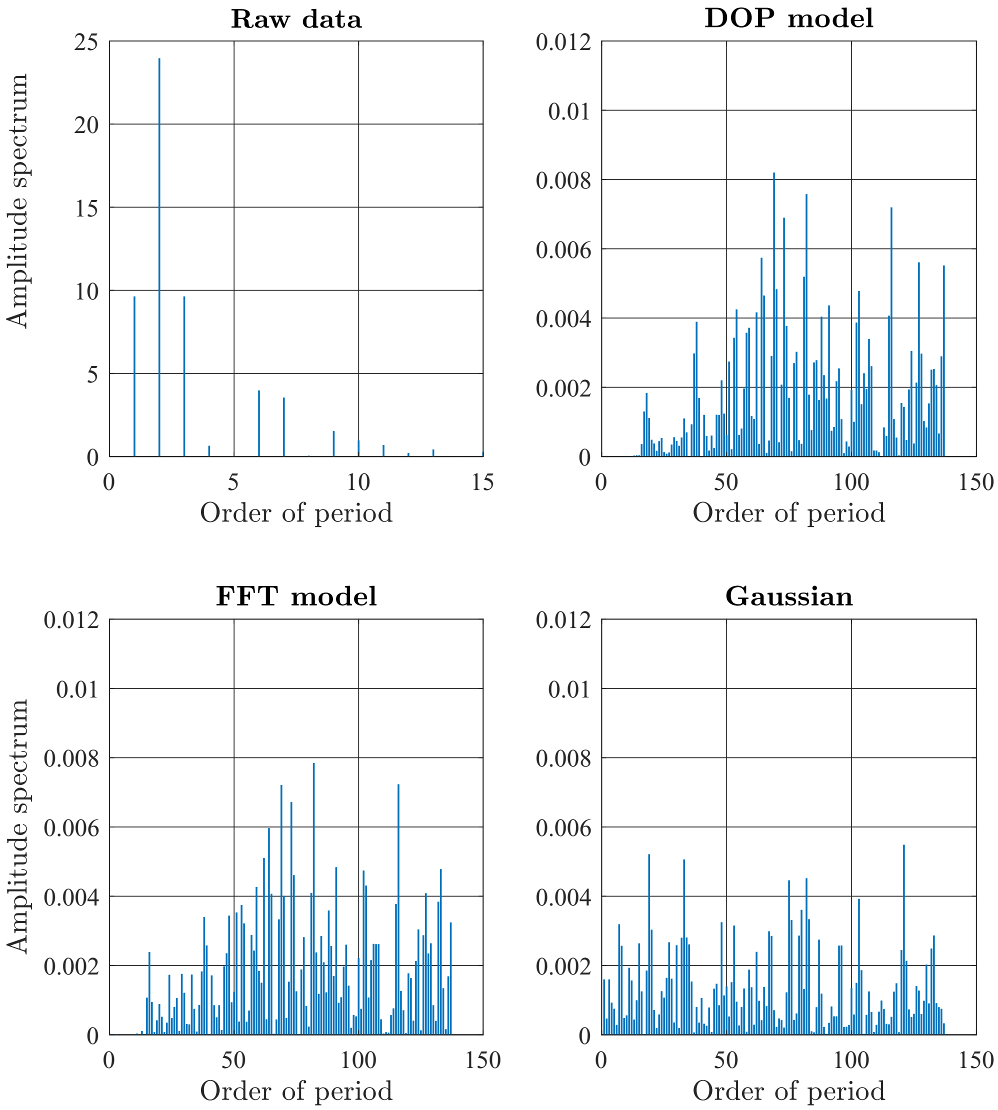

Figure 4.7: Results of the FFT spectral analysis for Run C (assembly tolerances according to Table 4.1) **Upper Left Plot:** raw measurement data, **Upper Right Plot:** Residual angle error of DOP filter model, **Lower Left Plot:** Residual angle error of FFT filter model and **Lower Right Plot:** of a Gaussian distribution with the same sample size.

The first orders of the Fourier Spectra of the DOP Model and the FFT Model are shown in Figure 4.8. The amplitudes of those low order terms have declined by six magnitudes for the residual angle error of the DOP Model in contrast to the amplitudes of the raw measurement data. The amplitudes of the low order terms of the residual angle error of the FFT Model has decreased by five magnitudes as well.

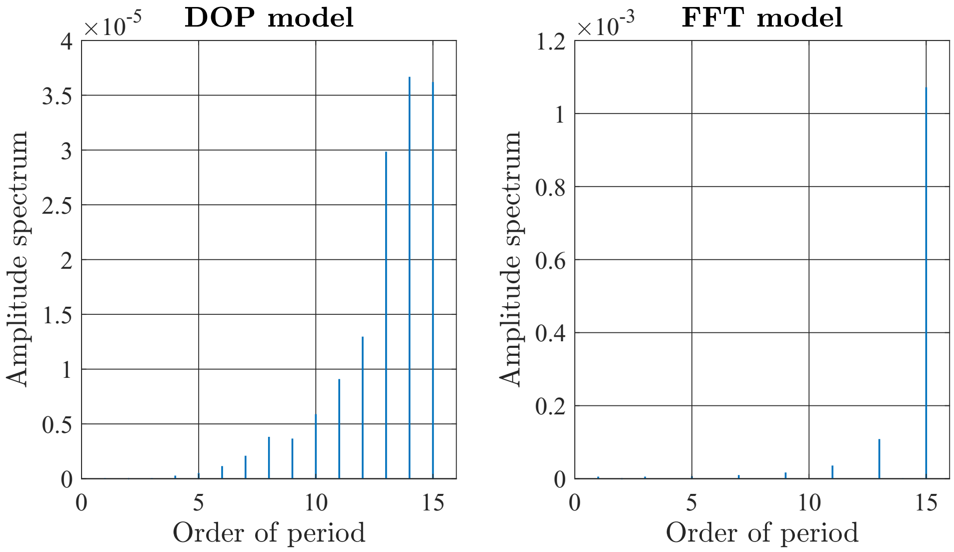

Figure 4.8: Zoom view (Orders 0 to 15) of FFT spectral analysis results of Figure 4.7 for **Left Plot:** Residual angle error of DOP filter model and **Right Plot:** Residual angle error of FFT filter model

A Probability Plot Correlation Coefficient Plot (PPCC) (section 2.5) was created of each run (A-F) to analyse the statistical distribution of the residual angle error $\Delta\varphi_{res}$ (see Figure 4.9). The measurement data was then centred and scaled and other statistical tests such as the Kolmogorov-Smirnov test (KS test), Lilliefors test (Lillie test) and Anderson-Darling test (AD test) were performed (the latter two of which being modifications of the KS test with respect to higher perceptibility of non Gaussian behaviour) to test against the null hypothesis that the measurement comes from a Gaussian distribution at the standard significance level of 5%. The results of these tests (”Accept” if the null hypothesis was accepted or ”Reject” if it the null hypothesis was rejected and showing that the data does not come from a Gaussian distribution) is shown with the extracted results of the PPCC plot above in Table 4.2. Technically speaking, this tests are not valid for proofing Gaussian distribution as they are not applicable for time-series analysis, as the values are sorted by magnitude beforehand.

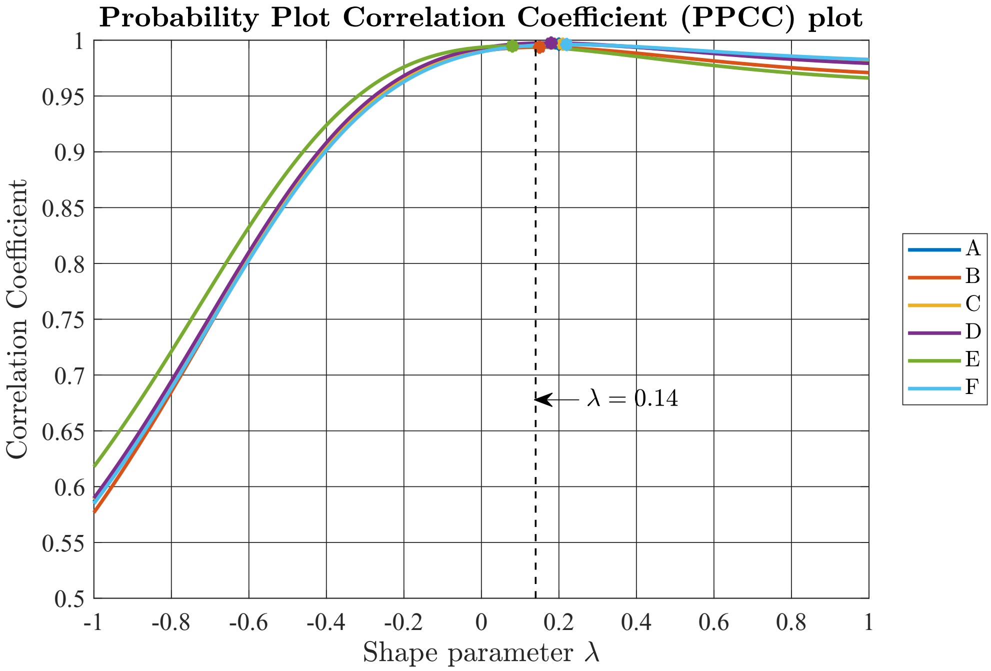

Figure 4.9: Probability Plot Correlation Coefficient (PPCC) Plot of the residual error of runs A-F (compare Table 4.1)

Table 4.2: Overview of statistical test results for all runs A-F (according to Table 4.1) of the residual error after performing polynomial approximation of the systematic error contribution for Tukey-Lambda (Location and correlation coefficient at the peak / correlation coefficient at $\lambda = 0.14$), Kolmogorov-Smirnov test (KS), Lilliefors test (Lillie) and Anderson-Darling test (AD). "Accept", if the null hypothesis was not rejected, "Reject" if otherwise.

Run	Samples	Tukey-Lambda	KS	Lillie	AD
A	136	0.20: 0.9965 / 0.14: 0.9962	Accept	Accept	Accept
B	136	0.15: 0.9937 / 0.14: 0.9934	Accept	Accept	Accept
C	136	0.21: 0.9965 / 0.14: 0.9860	Accept	Accept	Accept
D	136	0.18: 0.9975 / 0.14: 0.9974	Accept	Accept	Accept
E	136	0.08: 0.9948 / 0.14: 0.9942	Accept	Accept	Accept
F	136	0.22: 0.9962 / 0.14: 0.9954	Accept	Accept	Accept

With the information gathered from the tests above an assumption of Gaussian distributed angle errors seems valid. Owing to the sample count in each run, the PPCC plot may allow drifts of the curves' peak. The Correlation coefficient at $\lambda = 0.14$ is not significantly smaller than the peak value. Test runs with higher sample counts should even deliver better results.

Even the PPCC plot of a purely Gaussian distributed random variable may show a λ significantly higher or lower than 0.14. Such an example plot for 136 Monte Carlo samples (identical to sample count of measurement runs) of a Gaussian distributed random variable is shown in Figure 4.10.

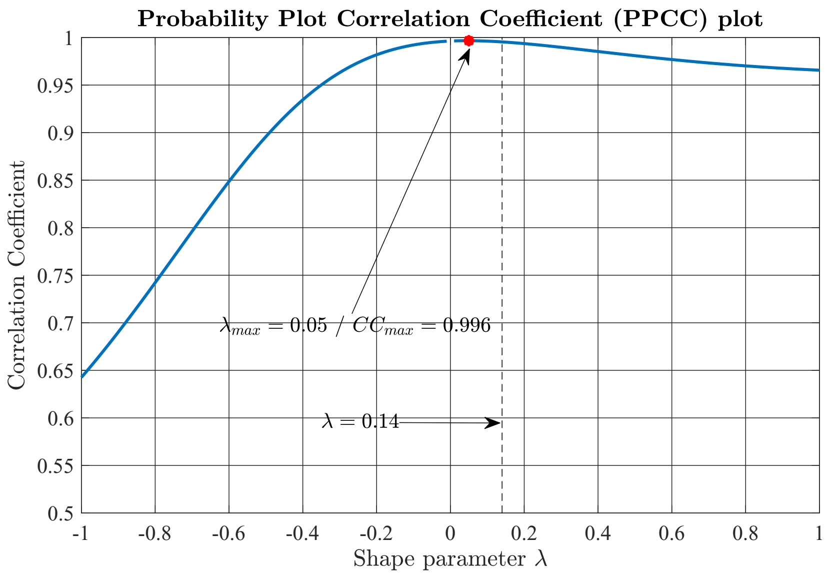

Figure 4.10: Example Probability Plot Correlation Coefficient (PPCC) plot for a Gaussian distributed random variable with 136 Monte Carlo samples taken. The peak correlation coefficient is 0.996 at λ=0.05 and 0.9918 at λ=0.14

All these methods and tests performed on the experimental data where necessary but not sufficient to *prove* a Gaussian distribution of the noise. During the analysis no evidence of non-Gaussian behaviour was found in the evaluated measurement data. A non-Gaussian behaviour is highly unlikely but cannot be ruled out.

Chapter 5

Discussion and Conclusion

5.1 Discussion

In this work, the entire signal processing path of an integrated gradient field based Hall angle sensor starting with the calculation of the magnetic field via the calculation of the Hall voltage under consideration of various factors (temperature, mechanical stress, back-biasing effects, ..) has been described with subsequent digitization and error compensation. This entire data path was modeled in MATLAB. The aim was to reproduce this complex system with the simplest possible models. In addition, the GUM uncertainty calculation concept was applied to this measurement path using a MATLAB toolbox (AUT) to show application fields for this toolbox and to verify how differential Hall sensors can be interpreted.

In the course of this signal processing path was often pointed out the presence of Gaussian distributed angle errors, which was also an elementary prerequisite for the proper functioning of the Kalman Filter was assumed. The presence of Gaussian distributed angular errors over time was analysed and demonstrated by the evaluation of measurement data. In [63] it had been shown that assembly tolerances at a fixed time for many sensors do not cause Gaussian distributed angle error values. Despite this fact, under consideration of the influence of the measuring path, only Gaussian distributed noise is present after a subsequent calibration. The applicability of the Kalman filter was thus guaranteed.

The practicality of the signal processing path was compared with simulation results and sufficient accuracy was determined (compare Section 3.2.2). The results of the signal processing path cohere with more complex simulation models. The calibrated results of the signal processing path are also consistent with raw experimental data. The residual angle error is reduced to a mere noise level of 0.05° (1σ value).

Specific experiences with AUT will be discussed in the next subsection.

5.1.1 Applicability of AUT

Based on my experience with AUT during this Master's thesis I can recommend the following fields of application. Assistance in teaching in lectures on metrology and statistical optimization. Because of their simple integration to the MATLAB environment, students are quickly familiar with the operations and can be introduced to the topic of statistical uncertainty using simple examples. Another application is certainly the statistical calculation of measurement paths of sensor products. A special application is surely this present master thesis.

The results of the toolbox must not be taken at face value. Whether the toolbox may be applicable for use in a particular application needs to be proven in advance. In simple cases it analytical solutions or in more complicated cases empirical or other statistical methods, such as the Monte Carlo simulation, which was also used here prove the applicability of the toolbox, can be chosen. As has been elaborated in this work, the toolbox on GUM is not suitable for the calculation of statistical uncertainties of differential magnetic fields. The applicability was by no means clear from the outset. Such a verification should definitely be made at the beginning of a statistical analysis.

What is the toolbox suitable for? With the toolbox, a statistical influence analysis can be carried out with a simple effort to find out which influencing factors affect how strongly the measurement result.

The main advantages and disadvantages of the toolbox are summarized below. Some of the advantages of the toolbox are:

1. Easy integration to the MATLAB environment
2. Easy definition of uncertain variables and perfect embedding using overwritten MATLAB functions
3. The most common MATLAB features are overwritten and unc() variables can be used as scalars and in vectors and matrices.
4. The toolbox can be modified (open source, once published)
5. Correlations between random variables are considered and a pre-definition is possible
6. The influencing parameters of a random variable can be displayed, showing not only which random variable have the biggest impact but also those with little or even no impact to the uncertainty of the result.

Some disadvantages are:

1. Increasing time consumption for computations of moderate complexity
2. Matlab functions need to be overwritten in order to be compatible with the toolbox (As far as they are not overwritten)
3. Overwriting so called ”in-built” functions is not easily possible as the source code of these functions is not available.
4. Some operations (such as Matrix/Vector operations or Matrix/Scalar operations) cannot be performed in matrix notation and need to be written as loops. This decreases the performance of the code significantly.
5. An additional name parameter needs to be assigned to each variable to compute correlations and covariances

6. As input variables are only defined by μ and σ it is not possible to distinguish between different probability distributions for input variables. Limited usage is possible with the unc_t() toolbox.

7. The toolbox is not fully bug-fixed.

During the work for this master thesis the functionality of the toolbox was tested thoroughly. In the following sub sections the main findings of each step will be discussed.

Magnetic Field Calculation

The toolbox is not suitable for the modelling of differential magnetic fields, since the method according to GUM only allows approximations of the Taylor series up to and including the first order. Exactly this first order terms are cancelled out due to the differential sensing principle. The combination of multiple assembly tolerances being present simultaneously causes non-neglectable angle errors that only appear in higher orders. Possible remedies include increasing the order of considered orders in the Taylor series expansion and Monte Carlo simulations.

Modelling Signal Processing

Using an abstract model of the real signal processing path, with simplifications in biasing and compensation circuits the Uncertainty Toolbox fully complied with Monte Carlo results. Evaluations of experimental results also returned a roughly accurate standard uncertainty of the angle error result.

Kalman Filtering

Some limitations arose for the Kalman filter with GUM uncertainty propagation. Each measurement point is linearly dependent on all previous data points caused by the recursive filter model. This results in a vast increase in computation time. However, this effect can be avoided if the linear dependence on the previous data points is cut off above a certain number. Performing Monte Carlo simulations instead is also a possible remedy.

5.1.2 Experimental verification

The experimental verification aimed to show that the noise of the angle error of a calibrated sensor is of Gaussian nature. Uncalibrated measurement data is massively perturbed by systematic errors, though. Fourier Analysis and Discrete Orthogonal Polynomials are able to remove this systematic portions. Statistical tests (such as Probability Plot Correlation Coefficient Plot, Kolmogorov Smirnov test,..) were run on the residual error together with spectral analysis. This methods are necessary but not sufficient to detect Gaussian noise in measurement. No indicators for non- Gaussian behaviour were found. This lead to the assumption that Gaussian noise is indeed present.

5.2 Conclusion

Taking numerical solutions of the 3D magnetic field solution for simple magnet shapes is sufficient enough for magnetic field calculations for sensing applications. Increasing magnet complexity usually leads to inevitable extensive FEM simulations. It was shown that basic kinematics knowledge are sufficient enough to model complex multi object systems with high accuracy. More complex models are not needed and increase the performance by a much smaller degree.

This is also somehow valid for the model of the signal processing path. The complexity of this model may be increased as desired, but increases the difficulty of recognizing individual influencing variables. The results in this thesis advise to choose a rather simple model, which remains manageable and makes an influence impact analysis easier.

In a way, this is also the goal of an engineer to reduce his task to a minimum of complexity in order to model the problem with sufficient accuracy. Technical standards and guidelines sort of choose this way. A deeper penetration into matter increases the level of complexity and makes it difficult for others to follow the thoughts and calculations. If one wants to design a program that is as easy to understand as possible that remains understandable for the entire target group, it must not exceed the level of comprehensibility of his target group.

Another important question addressed in this thesis was to analyse the applicability of Kalman Filters for predicting future states and smoothing noisy data. It was shown that Kalman Filters are fit to overcome angle errors caused by signal delay as long as the equations of state transition are fit to describe the motion sequence of the real system.

The concept of uncertainty propagation was introduced using the standards of ”Guide to the Expression in Measurement” (GUM) [54]. It was shown that GUM can deliver very good results, the simplifications due to linearisation (considering only up to first order terms of Taylor series expansion) can lead to fatal results when considering differential signals. Verification by means of a Monte Carlo simulation is indispensable in cases of doubt, it also represents the secure and more conservative approach. Monte Carlo simulations may be even faster if the complexity of the model increases up to a certain degree.

5.3 Outlook and future scope

To round off this Master's thesis, ideas for future research will be given based on the methods and results presented in this thesis. The models described in this thesis may build the basis in future sensor design, showing the tools available for performing fast sensor dimensioning with sufficient accuracy to significantly reduce time needed for development.

Magnetic field calculation

The magnetic field calculation script proved to be valuable companion for statistical analysis of assembly tolerances of gradient field based magnetic sensors with diametrically polarized disc magnets. During the work for this thesis a publication was written and published in October 2019 [63]. This script can be used for statistical optimization regarding the error contributions of assembly tolerances to the angle error result for any given magnet shape and sensor dimensions.

Signal process modelling

The signal processing model presented in this thesis will allow Concept Engineers to perform quick and reliable analysis of sensor performance under the given layout for various temperatures and stress levels. The underlying Matlab script can also be modified slightly to perform correlation analysis of single electrical components of the sensor layout. In future works the script can be modified to increase the performance of the results (compared to simulation results) by increasing the level of complexity in the script.

Kalman filtering

The concept of Kalman Filtering and its capabilities for smoothing and predicting measurements of Angular sensing products was introduced in this thesis. Further research may be conducted on optimizing filter algorithm and optimal firmware coding. Once perfectly tuned, Kalman filters can be used to improve the functionality of safety critical automotive sensing devices.

Uncertainty Toolbox

The work on the Uncertainty Toolbox is far from complete. Future work could include taking into account higher order terms of the Taylor series expansion to the Uncertainty Toolbox, as well as making the most important MATLAB functions compatible with the Toolbox. The extension for Monte Carlo simulations is already a step in the right direction, this could be adapted in that even correlations between random variables and non-normally distributed random variables can be created, as well as the compatibility with MATLAB functions could be produced here.

Bibliography

[1] Alpen Adria University of Klagenfurt. (2019) Institute of Smart Systems Technologies. Last Accessed On: 2019-10-29. [Online]. Available: www.aau.at/sat

[2] Automotive Electronics Council. (2007) AEC-Q100-Rev-G: Failure Mechanism Based Stress Test Qualification For Integrated Circuits. AEC (Automotive Electronics Council). Last Accessed On: 2019-10-29. [Online]. Available: http://www.aecouncil.com/Documents/AEC_Q100_Rev_G_Base_Document.pdf

[3] wissenschaft.de. (2006) Die Anforderungen der Automobilindustrie. Last Accessed On: 2019-04-25. [Online]. Available: https://www.wissenschaft.de/allgemein/die-anforderungen-der-automobilindustrie/

[4] J. Andris and G. Kucera, "Hall-Sensoren im Automobilbau: Lösungen für die Entwicklung," 2016, last Accessed On: 2019-11-04. [Online]. Available: https://www.elektronikpraxis.vogel.de/hall-sensoren-im-automobilbau-loesungen-fuer-die-entwicklung-a-536711/index5.html

[5] E. Ramsden, *"Hall-Effect Sensors (Second Edition)"*. Burlington: Newnes, 2006.

[6] P. E. Ripka, *Magnetic Sensors and Magnetometers*, 1st ed., P. Ripka, Ed. Artech House, 2001.

[7] W. Granig, S. Hartmann, and B. Köppl, "Performance and Technology Comparison of GMR versus commonly used Angle Sensor Principles for Automotive Applications," *SAE Transactions*, vol. 116, pp. 29–41, 2007, last Accessed On: 2019-07-18. [Online]. Available: https://www.jstor.org/stable/44719865?seq=1#page_scan_tab_contents

[8] Infineon Technologies Austria AG. (2018) TLE5046iC Technical Product Description. Last Accessed On: 2019-11-04. [Online]. Available: https://www.infineon.com/dgdl/Infineon-TLE5046_technProdDescription.pdf-DS-v01_00-EN.pdf?fileId=5546d46266f85d6301671711e5fa165b

[9] International Electrotechnical Commission, *IEC 61508:2010 - Functional Safety of Electrical/ Electronical/ Programmable Electronic Safety-Related Systems*, 2nd ed., 2010.

[10] International Standardisation Organisation, *ISO 26262: 2011 - Functional Safety on Road Vehicles*, 2011.

[11] Automotive Electronics Council. (2019) Automotive Electronics Council Homepage. Automotive Electronics Council. Last Accessed On: 2019-10-29. [Online]. Available: http://www.aecouncil.com/

[12] ——, *AEC-Q100: Failure Mechanism Based Stress Test Qualifcation for Intergrated CircuitsAECQ-100*, Rev-H, 2014.

[13] K. H. Ludwig and V. Schmidtchen, *Metalle und Macht (German).* Ullstein, 1992.

[14] National MAGLAB. (2014) Magnetometer – 1832. Last Accessed On: 2019-09-17. [Online]. Available: https://nationalmaglab.org/education/magnet-academy/history-of-electricity-magnetism/museum/magnetometer

[15] A. Günther, A. Einwich, E. Sjulstok, R. Feederle, P. Bolte, K. W. Koch, I. Solov'yov, and H. Mouritsen, "Double-Cone Localization and Seasonal Expression Pattern Suggest a Role in Magnetoreception for European Robin Cryptochrome 4," *Current Biology*, vol. 28, 2018, last Accessed On: 2019-11-04. [Online]. Available: https://www.cell.com/current-biology/pdf/S0960-9822(17)31605-6.pdf

[16] W. Granig, C. Kolle, C. D. Hammerschmidt, B. Schaffer, R. Borgschulze, C. Reidl, and J. Zimmer, "Integrated Gigant Magnetic Resistance based Angle Sensor," in *IEEE Sensors 2006*, vol. 1. Daegu, Korea: IEEE Sensors, 10 2006.

[17] W. Granig, M. Weinberger, C. Reidl, M. Bresch, M. Strasser, and G. Pircher, "Integrated GMR Angle Sensor for Electrical Commutated Motors including Features for Safety Critical Applications," in *Proceedings Eurosensors XXIV*, vol. 1. Linz, Austria: Eurosensors, 9 2010.

[18] U. Ausserlechner, "Inaccuracies of Giant Magneto-Resistive Angle Sensors Due to Assembly Tolerances," *IEEE Transactions on Magnetics*, vol. 45, pp. 2165–2174, 2009. [Online]. Available: https://ieeexplore.ieee.org/document/4816026/

[19] P. Ripka and M. Janosek, "Advances in Magnetic Field Sensors," *IEEE Sensors Journal*, vol. 10, pp. 1108–1116, 2010.

[20] L. A. Francis and K. Poletkin, *Magnetic Sensors and Devices: Technologies and Applications*. CRC Press, London, UK, 2017.

[21] M. Metz, A. Häberli, M. Schneider, R. Steiner, R. Maier, and B. H., "Contactless Angle Measurement Using Four Hall Devices on Single Chip," *Proceedings of International Solid State Sensors and Actuators Conference (Transducers '97)*, pp. 385–388, 1997.

[22] T. Kaulberg and G. Bogason, "An Angledetector based on magnetic sensing," *Proceedings of IEEE International Symposium on Circuits and Systems - ISCAS '94*, pp. 329–332, 1994.

[23] U. Ausserlechner, "A Theory of Magnetic Angle Sensors with Hall Plates and without Fluxguides," *Progress In Electromagnetics Research B*, vol. 49, pp. 77–106, 2013.

[24] Allegro Microsystems LLC, *Guidelines for Using Allegro Angular Sensors*, 2015, Last Accessed On: 2019-09-23. [Online]. Available: https://www.allegromicro.com/-/media/Files/Technical-Documents/AN269115-Guidelines-for-Using-Allegro-Angular-Sensors.ashx?la=en&hash=5E2D4145EB0B60E9D0BE27FD64F6BD837015FC68

[25] E. Hall, “On a new Action of the Magnet on Electric Currents,” *American Journal of Mathematics*, vol. 2, pp. 287–292, 1879, last Accessed On: 2019-11-04. [Online]. Available: http://cc.ee.ntu.edu.tw/~thlin/E_Hall_paper_1879.pdf

[26] J. Cesaretti, T. Wilson, M. Guillermo, and O. Brand, “Effect of Stress Due to Plastic Package Moisture Absorption in Hall Sensors,” *IEEE Transactions on Magnetics 2009*, 2009.

[27] W. Granig, L.-M. Faller, and H. Zangl, “Sensor System Optimization to Meet Reliability Targets,” *Elsevier*, vol. 87, pp. 113–124, 2018.

[28] Allegro Microsystems LLC. (1996) Chopped Hall sensor with synchronously chopped sample-and-hold circuit. Last Accessed On: 2019-04-26. [Online]. Available: https://www.allegromicro.com/en/Design-Center/Allegro-Technology/Hall-Effect-Sensor-Technology/Chopper-Stabilization.aspx

[29] J. Edberg and F. Hintz. (2013) Busting three myths of chopper stabilization. Last Accessed On: 2019-09-17. [Online]. Available: http://industrial.embedded-computing.com/articles/busting-three-myths-chopper-stabilization/

[30] Infineon Technologies AG, *TLE4998S3, TLE4998S3S4 - Programmable Linear Hall Sensor*, 2008, last Accessed On: 2011-11-02. [Online]. Available: www.infineon.com

[31] M. Motz, U. Ausserlechner, W. Scherr, and B. Schaffer, “An Integrated Magnetic Sensor with Two Continuous-Time ΔΣ-Converters and Stress Compensation Capability,” 2006, pp. 1151–1160.

[32] Honeywell Inc., “Hall Effect Sensing And Application,” last Accessed On: 2019-05-09. [Online]. Available: https://sensing.honeywell.com/hallbook.pdf

[33] F. Rasbornig, W. Granig, and J. Cunha, “Monolithic Digital Magnetic Hall Sensor with two Measurement Paths for an ASIL D Application,” 2017.

[34] W. Granig, F. Rasbornig, D. Hammerschmidt, M. Motz, T. Zettler, M. Strasser, and A. Michelutti, “Redundant and Diverse Magnetic Field Digital Linear Hall Sensor Concept for ASIL D Applications,” in *SAE Technical Paper 2017-01-0053*, 2017.

[35] Texas Instruments, *Principles of Data Acquisition and Conversion*, 2015, Last Accessed On: 2019-10-03. [Online]. Available: http://www.ti.com/lit/an/sbaa051a/sbaa051a.pdf

[36] Microchip Technology, Inc. (2019) ADC Quantization Error. Last Accessed On: 2019-10-03. [Online]. Available: https://microchipdeveloper.com/adc:adc-quantization-error

[37] ——. (2019) ADC Offset Error. Last Accessed On: 2019-10-03. [Online]. Available: https://microchipdeveloper.com/adc:adc-offset-error

[38] ——. (2019) ADC Differential Nonlinearity. Last Accessed On: 2019-10-03. [Online]. Available: https://microchipdeveloper.com/adc:adc-differential-nonlinearity

[39] Cypress Inc. (2019) ADC Errors. Last Accessed On: 2019-10-03. [Online]. Available: https://www.cypress.com/file/144536/download

[40] Microchip Technology, Inc. (2019) ADC Integral Nonlinearity (INL). Last Accessed On: 2019-10-03. [Online]. Available: https://microchipdeveloper.com/adc:adc-inl

[41] Infineon Technologies AG, *TLE5xxx(D) Calibration 360 deg of GMR/TMR-based Analog Angle Sensor*, 2018, last Accessed On: 2019-10-03. [Online]. Available: https://www.infineon.com/dgdl/Infineon-TLE5xxx(D)_Calibration_360_AN-v02_00-AN-v02_00-EN.pdf?fileId=5546d46264a8de7e0164f09d8bfa228d

[42] ——, *Out of Shaft with magnetic 3D sensor*, 2018, last Accessed On: 2019-10-03. [Online]. Available: https://www.infineon.com/dgdl/Infineon-Out_of_Shaft-AN-v01_00-EN.pdf?fileId=5546d46265257de801653898ba536074

[43] ——, *3D Magnetic Sensor for Angle Measurements*, 2018, last Accessed On: 2019-10-03. [Online]. Available: https://www.infineon.com/dgdl/Infineon-3D_Magnetic_Sensor_for_Angle_Measurements-AN-v01_10-AN-v01_01-EN.pdf?fileId=5546d46265f064ff01665a3d22f055e3

[44] Diodes Incorporated. (2015) AH8502 Datasheet. Diodes Incorporated. Last Accessed On: 2019-06-24. [Online]. Available: https://www.diodes.com/assets/Datasheets/AH8502.pdf

[45] R. Eisenbeis. (2017) Power Gating Systems with Magnetic Sensors. Texas Instruments. Last Accessed On: 2019-06-24. [Online]. Available: http://www.ti.com/lit/an/sboa196a/sboa196a.pdf

[46] Micronas. (2008) Product Information HAL 2850. Last Accessed On: 2019-06-24. [Online]. Available: https://www.micronas.com/en/system/files/downloads/files/hal2850_1pi.pdf

[47] R. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of Basic Engineering (ASME)*, vol. 82, pp. 35–45, March 1960, Last Accessed On: 2019-10-30. [Online]. Available: https://pdfs.semanticscholar.org/bb55/c1c619c30f939fc792b049172926a4a0c0f7.pdf?_ga=2.228633566.35040265.1572415709-761956564.1550748434

[48] T. Lacey and N. Thacker, “Tutorial: The Kalman Filter.” Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester,, 1998, last Accessed On: 2019-07-16. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.408.5594&rep=rep1&type=pdf

[49] P. Zarchan and H. Musoff, *Fundamentals of Kalman Filtering: A Practical Approach*, ser. Fundamentals of Kalman filtering: a practical approach. American Institute of Aeronautics and Astronautics, Incorporated, 2000,

no. Bd. 190, last Accessed On: 2019-11-04. [Online]. Available: https://books.google.de/books?id=AQxRAAAAMAAJ

[50] S. Julier and J. Uhlmann, "A new extension of the Kalman Filter to nonlinear systems," in *Proceedings of the 11th International Symposium on Aerospace/Defense Sensing, Simulation, Control*, 1997, pp. 182–193, last Accessed On: 2019-10-30. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.5.2891&rep=rep1&type=pdf

[51] A. Becker. (2018) Kalman Filter Tutorial. Last Accessed On: 2019-07-08. [Online]. Available: https://www.kalmanfilter.net/

[52] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice Using Matlab*. Wiley-Interscience Publication, September 2008, last Accessed On: 2019-10-30. [Online]. Available: https://www.researchgate.net/publication/243786641_Kalman_filtering_theory_and_practice_using_MATLAB

[53] Univeristy of Cambridge. (2019) Cambridge Dictionary- The dictionary of the University of Cambridge. Last Accessed On: 2019-02-17. [Online]. Available: https://dictionary.cambridge.org/dictionary/english/uncertainty

[54] Bureau International des Poids et Mesures. (2019) Official Website of BIPM. Last Accessed On: 2019-10-29. [Online]. Available: https://www.bipm.org/en

[55] *JCGM 100:2008 Guide to the expression of uncertainty in measurement*, 2008.

[56] *JCGM 101:2008: Evaluation of measurement data - Supplement 1 to the "Guide to the expression of uncertainty in measurement" - Propagation of distributions using a Monte Carlo method Evaluaton of measurement data*, 2008.

[57] *JCGM 102:2011: Evaluation of measurement data - Supplement 2 to the "Guide to the expression of uncertainty in measurement" - Extension to any number of output quantities*, 2011.

[58] *JCGM 104:2009 Evaluation of measurement data – An introduction to the "Guide to the expression of uncertainty in measurement" and related documents*, 2009.

[59] *JCGM 106:2012: Evaluation of measurement data – The role of measurement uncertainty in conformity assessment*, 2012.

[60] Joint Committee for Guides in Metrology, *JCGM 200:2012 - International vocabulary of metrology - Basic and general concepts and associated terms (VIM)*, 2012.

[61] (2019) GUM: Guide to the Expression of Uncertainty in Measurement. Last Accessed On: 2019-09-16. [Online]. Available: https://www.bipm.org/en/publications/guides/gum.html

[62] H. P. Hsu, *Schaum's Outline of Theory and Problems of Probability, Random Variables, and Random Processes.* McGraw-Hill, 1997.

[63] S. Ergun, U. Ausserlechner, M. Holliber, W. Granig, and H. Zangl, "A Statistical Investigation into Assembly Tolerances of Gradient Field Magnetic Angle Sensors with Hall Plates," *Mathematics*, vol. 7, no. 10, 2019, Last Accessed On: 2019-11-04. [Online]. Available: https://www.mdpi.com/2227-7390/7/10/968

[64] M. Zeier, J. Hoffman, and M. Wollensack, “Metas.UncLib—a measurement uncertainty calculator for advanced problems,” *Metrologia*, vol. 49, pp. 809–815, 2012, Last Accessed On: 2019-10-29. [Online]. Available: https://www.metas.ch/dam/data/metas/Fachbereiche/Hochfrequenz/UncLib/metasunclib-a-measurement-uncertainty-calculator-for-advanced-problems.pdf

[65] Standards Laboratory of New Zealand. (2019) Resources of Measurement Standards Laboratory of New Zealand. Last Accessed On: 2019-10-29. [Online]. Available: https://www.measurement.govt.nz/resources

[66] M. Zia. (2018) Uncertainty Toolbox. Last Accessed On: 2019-10-29. [Online]. Available: http://www.iosense.eu/index.php/2018/10/02/uncertainty-toolbox/

[67] (2019) Object-Oriented Design with MATLAB. Last Accessed On: 2019-10-30. [Online]. Available: https://de.mathworks.com/help/matlab/object-oriented-design-with-matlab.html

[68] H. Zangl and K. Hoermaier, “Educational aspects of uncertainty calculation with software tools,” *Measurement*, vol. 101, pp. 257–264, 2017, Last Accessed On: 2019-11-04. [Online]. Available: https://doi.org/10.1016/j.measurement.2015.11.005

[69] W. Granig, S. Aoudjit, L.-M. Faller, and H. Zangl, “Statistical Modeling of Integrated Sensors for Automotive Applications,” in *2018 International Conference of Electrical and Electronic Technologies for Automotive (AUTOMOTIVE2018)*, Milano, Italy, Jul. 2018, pp. 1–5.

[70] H. Zangl, M. Zine-Zine, and K. Hoermaier, “Utilization of Software Tools for Uncertainty Calculation in Measurement Science Education,” *Journal of Physics*, 2015, last Accessed On: 2019-10-29. [Online]. Available: https://www.researchgate.net/publication/272525508_Utilization_of_Software_Tools_for_Uncertainty_Calculation_in_Measurement_Science_Education

[71] D. Arronde Pérez and H. Zangl, “Introducing uncertainty of complex-valued quantities in measurement science education,” *SPIE Proceeding*, vol. 11144, no. 410, 2019, last Accessed On: 2019-10-30. [Online]. Available: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11144/1114410/Introducing-uncertainty-of-complex-valued-quantities-in-measurement-science-education/10.1117/12.2531840.full?SSO=1

[72] S. Auodjit, “Sensor Modeling under Uncertainty using MATLAB Uncertainty Toolbox,” Master’s thesis, 2018.

[73] J. Tukey, “The Practical Relationship Between the Common Transformations of Percentages of Counts and Amounts,” *Technical Report 36. Statistical Techniques Research Group, Princeton University*, 1960.

[74] J. S. Ramberg and B. W. Schmeiser, “An approximate method for generating symmetric random variables,” *Communications of the ACM*, vol. 15, pp. 987–990, 1972.

[75] P. Wessa. (2013) Tukey lambda PPCC Plot (v1.0.3) in Free Statistics Software (v1.2.1). Last Accessed On: 2019-10-29. [Online]. Available: http://www.wessa.net/rwasp_tukeylambda.wasp/

[76] NIST/SEMATECH. (2006) e-Handbook of Statistical Methods. Last Accessed On: 2019-09-05. [Online]. Available: http://www.itl.nist.gov/div898/handbook/ ,2006-10-03.

[77] E. Furlani, "A three-dimensional field solution for axially-polarized multipole discs," *Journal of Magnetism and Magnetic Materials*, vol. 135, pp. 205–214, 1994.

[78] V. T. Nguyen and T.-F. Lu, "Analytical Expression of the Magnetic Field Created by a Permanent Magnet with Diametrical Magnetization," *Progress In Electromagnetics Research C*, vol. 87, pp. 163–174, 2018, last Accessed On: 2019-10-30. [Online]. Available: http://www.jpier.org/PIERC/pierc87/14.18073001.pdf

[79] H. Zangl, T. Bretterklieber, G. Steiner, and K. Riedmueller, "Feasibility Study for a Three-Axial Joystick based on an Array of Lateral Hall Elements," in *Proceedings of IEEE Sensors*, Atlanta, GA, USA, October 28-31, 2007, pp. 1069–1072.

[80] M. Ortner. (2019) magpylib- A simple, user friendly Python 3 toolbox for calculating magnetic fields from permanent magnets and current distributions. Last Accessed On: 2019-09-19. [Online]. Available: https://pypi.org/project/magpylib/

[81] J. E. Szymanski, *Basic Mathematics for Electronic Engineers: Models and Applications*. Taylor&Francis: London, UK, 1989, 1989.

[82] J. Hanson, "Rotations in three, four, and five dimensions," p. 2, 2011, Last Accessed On: 2019-07-31. [Online]. Available: https://arxiv.org/pdf/1103.5263.pdf

[83] R. C. James, *Mathematics Dictionary*. Springer Science & Business Media, Berlin, Germany, 1992.

[84] H. Ali, "Stress-induced parametric shift in plastic packaged devices," *IEEE Transactions on Components, Packaging, and Manufacturing Technology*, vol. 20, pp. 458–462, 1997, lAst Accessed On: 2019-10-29. [Online]. Available: https://ieeexplore.ieee.org/document/641515/keywords

[85] U. Ausserlechner, M. Motz, and M. Holliber, "Drift of magnetic sensitivity of smart Hall sensors due to moisture absorbed by the IC-package," in *Sensors*. IEEE, 2004.

[86] M. Motz and U. Ausserlechner, "Electrical Compensation of Mechanical Stress Drift in Precision Analog Circuits," *Wideband Continuous ΣΔ ADCs, Automotive Electronics and Power Management, Chapter 16, Springer International*, pp. 297–325, July 2017.

[87] J. E. Volder, "The CORDIC Trigonometric Computing Technique," *IRE Transactions on Electronic Computers*, vol. 8, no. 3, pp. 330–334, May 1959, last Accessed On: 2019-10-31. [Online]. Available: https://www.jacques-laporte.org/Volder_CORDIC.pdf

[88] Mathworks Inc. (2019) atan2: Four-quadrant inverse tangent. Last Accessed On: 2019-10-16. [Online]. Available: https://de.mathworks.com/help/matlab/ref/atan2.html

[89] Sensitec GmbH, *Magnetische Mikround Nanotechnologie für robuste Sensorlösungen [German]*, 2015, last Accessed On: 2019-10-10. [Online]. Available: https://www.sensitec.com/fileadmin/sensitec/Service_and_Support/Downloads/Catalogue/Sensitec_Katalog2015_Web_LZ_de.pdf

[90] P. M. Feldman. (2009) MATLAB File Exchange- autocov.m. The Mathworks, Inc. Last Accessed On: 2019-10-09. [Online]. Available: https://de.mathworks.com/matlabcentral/fileexchange/24066-autocov-m

[91] M. Harker and P. O'Leary. (2014) Discrete Orthogonal Polynomial Toolbox: DOPBox Version 1.8 -MATLAB File Exchange. Last Accessed On: 2019-10-29. [Online]. Available: https://de.mathworks.com/matlabcentral/fileexchange/41250-discrete-orthogonal-polynomial-toolbox-dopbox-version-1-8?s_cid=ME_prod_FX

Appendix A

Glossary

Air Gap - The mean radial distance from the surface plane of a magnet to the housing of the sensor (or to its Hall plate)

Bandwith - A frequency domain range in which a device (e.g. sensor) can transmit signals

CORDIC -**C**oordinate **r**otation **d**igital **c**omputer; In the sense of this project: Tranforms sine and cosine data to angle and amplitude

Datapath -The flow of analogue (voltages,currents) and digital (bit values) data through the measurement chain

EEPROM -**E**lectrically **E**rasable **P**rogrammable **R**ead-**O**nly **M**emory; An EEPROM is a non-volatile memory used in microcontrollers and computers for storing small data amounts. Single bytes can be erased and re-written electrically

Measurand - The quantitty intended to be measured

MOSFET -**M**etal **O**xide **S**emiconductor **F**ield **E**ffect **T**ransistor; it controls the current flow by using an electric field

OEM - **O**riginal **E**quipment **M**anufacturer; Has a different meaning in the automotive sector compared to other industry sectors. In the general case it defines a company which produces parts and devices that are marketed by another company. The term OEM has no strict definition in the automotive sector, as leading car manufacturers around the globe call themselves OEM's, to separate them from the real OEM's and their suppliers, a pyramid like structure has been established. Automotive manufacturers are called ”Tier 1” OEM, the actual OEM's are then called ”Tier 2” OEM, their suppliers are called ”Tier 3” suppliers and so on. With the ”Tier” count the number of companies increases causing an immense price war which is usually dictated by the ”Tier 1” companies.

PPCC - Probability Plot Correlation Coefficient Plot: A plot computing the correlation between a set of noisy data to a set of given probability distribution. Most commonly, the Tukey-Lambda distribution is being used. The aim is it to identify the distribution of the noisy data.

ppm - Parts per million: Number of broken or malfunctioning (in a more general sense: marked or distinctive) parts in a batch size of one million parts

reading radius - The radial distance of each magneto-sensitive element to the centre of the sensor

Remanence - The remaining flux density after saturation and without further excitation in a permanent magnet

SmCo - Samarium-Cobalt: A rare earth-metal compound magnet with strong magnetization, typically used in magnetic field sensors

Transfer Function - Describes the output in terms of its input, the transfer function can be a mathematical function or a curve in a diagramm

White noise - Is a type of noise which has equal intensities at different frequencies. A Gaussian distribution shows such an behaviour in the frequency domain. Other colours of noise are: Pink, Red and Grey

Appendix B

MATLAB Codes

B.1 Kalman Filter initialization

```
%% Kalman Filter initialization
% Serkan Ergun
% Uncertainty Modelling in Sensor Systems

%% Defining constants and initial values

% sample Time/ sample interval
dT;
% Measurement Noise Variance
v=var(measurementNoise);
% Process Noise Variance
w=var(processNoiseSpeed);

%% Loop, usually less than 50 iterations are needed for converging
for i=1:50

% Kalman Gain matrix (K=[K_phi,K_deltaPhi]')
      % Common denominator
      K_denom=P_phi_phi_pred+v;
% K_phi
K_phi=P_phi_phi_pred/K_denom;
% K_deltaPhi
K_deltaPhi=P_deltaPhi_phi_pred/K_denom;
% Covariance Estimate update
%(P=[P_phi_phi, P_phi_deltaPhi;
%   P_deltaPhi_phi, P_deltaPhi_deltaPhi])
P_phi_phi      =     (1-K_phi) * P_phi_phi_pred;
P_phi_deltaPhi =     (1-K_phi) * P_phi_deltaPhi_pred;
P_deltaPhi_phi =    -K_deltaPhi * P_phi_phi_pred + P_deltaPhi_phi_pred;
P_deltaPhi_deltaPhi= -K_deltaPhi * P_phi_deltaPhi_pred + P_deltaPhi_deltaPhi_pred;
% Covariance Prediction
% (P_pred=[P_phi_phi_pred, P_phi_deltaPhi_pred;
         P_deltaPhi_phi_pred, P_deltaPhi_deltaPhi_pred])
P_phi_phi_pred    =  P_phi_phi      + dT * P_deltaPhi_phi +
         dT * (P_phi_deltaPhi + dT * P_deltaPhi_deltaPhi);
P_phi_deltaPhi_pred = P_phi_deltaPhi + dT *
         P_deltaPhi_deltaPhi;
P_deltaPhi_phi_pred = P_deltaPhi_phi + dT *
         P_deltaPhi_deltaPhi;
P_deltaPhi_deltaPhipred = P_deltaPhi_deltaPhi + w;
end
```

Listing B.1: Matlab code snippet for initilaizing the Kalman Filter algorithm

B.2 Kalman Filter loop

```
%% Kalman Filter loop
% Serkan Ergun
% Uncertainty Modelling in Sensor Systems

% Update Estimate
      % Compute Kalman Residual
            phi_Residual=phi_tilde_cal-phi_Hat_Predicted;
      % Update Position Estimate
            phi_Hat=phi_Hat_Predicted+K_phi*phi_Residual;
      % Update Speed Estimate
            deltaPhi_Hat=deltaPhi_Hat_Predicted+K_deltaPhi*
                     phi_Residual;
% Next State Prediction
      % Predict Position Estimate
            phi_Hat_Predicted=phi_Hat+dT*deltaPhi_Hat;
      % Predict Speed Estimate
            deltaPhi_Hat_Predicted=deltaPhi_Hat;
```

Listing B.2: Matlab code snippet for the Kalman Filter loop algorithm

Appendix C

Additional Experimental Results

C.1 Angle measurement evaluation

This sections feature the additional results of the angle error measurement evaluation of presented in section 4.2.

C.1.1 Run A

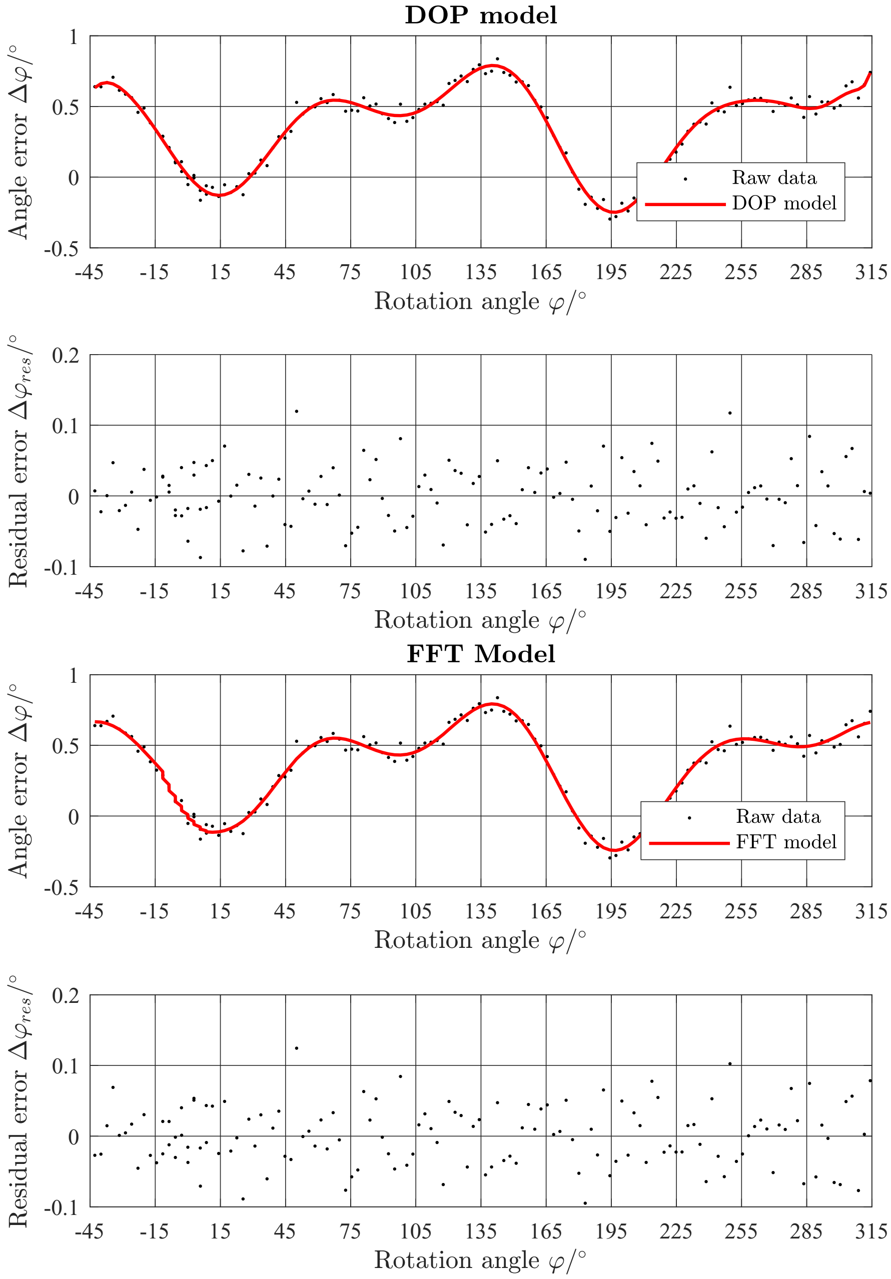

Figure C.1: Results of the DOP and FFT filter models for Run A (assembly tolerances according to Table 4.1). **Top Plot:** Uncalibrated angle error $\Delta\varphi$ and DOP filter model results over rotation angle φ. **Upper Central Plot:** Resulting residual error $\Delta\varphi_{res}$ of DOP model to raw data. **Lower Central Plot:** $\Delta\varphi$ and FFT filter model results over φ. **Bottom Plot:** $\Delta\varphi_{res}$ of FFT model to raw data.

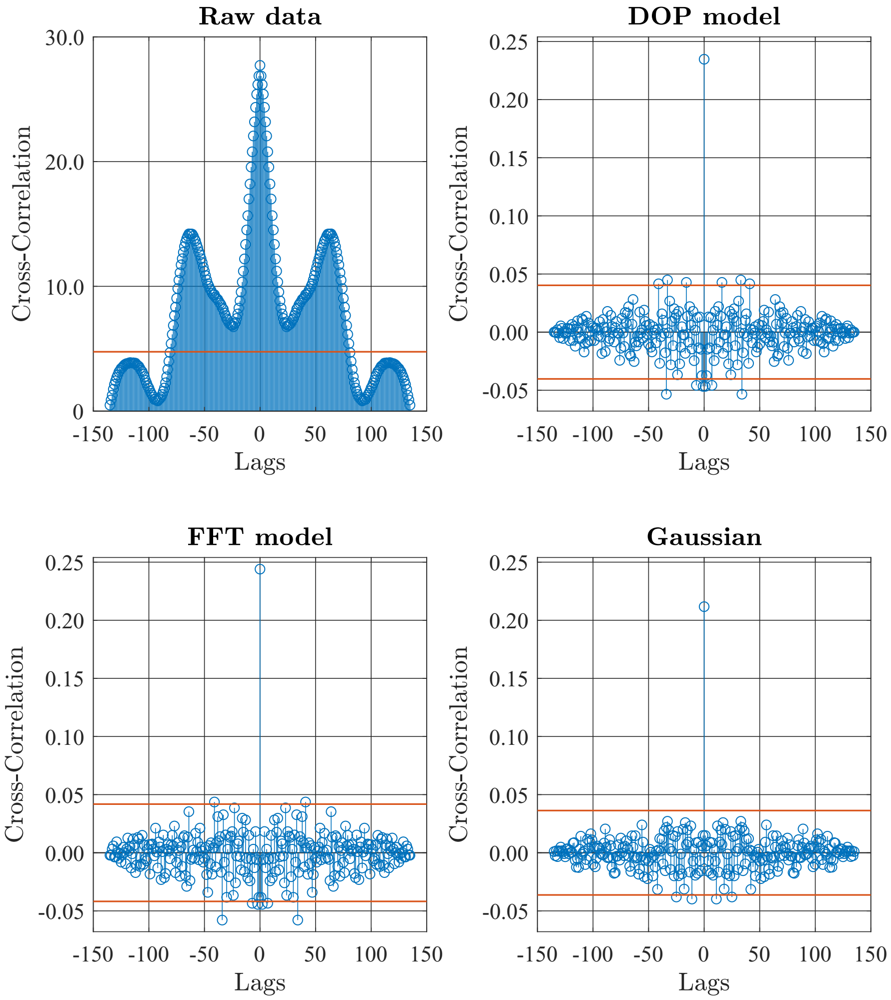

Figure C.2: Cross-Correlation functions for Run A (assembly tolerances according to Table 4.1) of **Upper Left Plot:** raw measurement data, **Upper Right Plot:** Residual angle error of DOP filter model, **Lower Left Plot:** Residual angle error of FFT filter model and **Lower Right Plot:** of a Gaussian distribution with the same sample size.The red lines indicate the 2σ confidence interval for the given sample size.

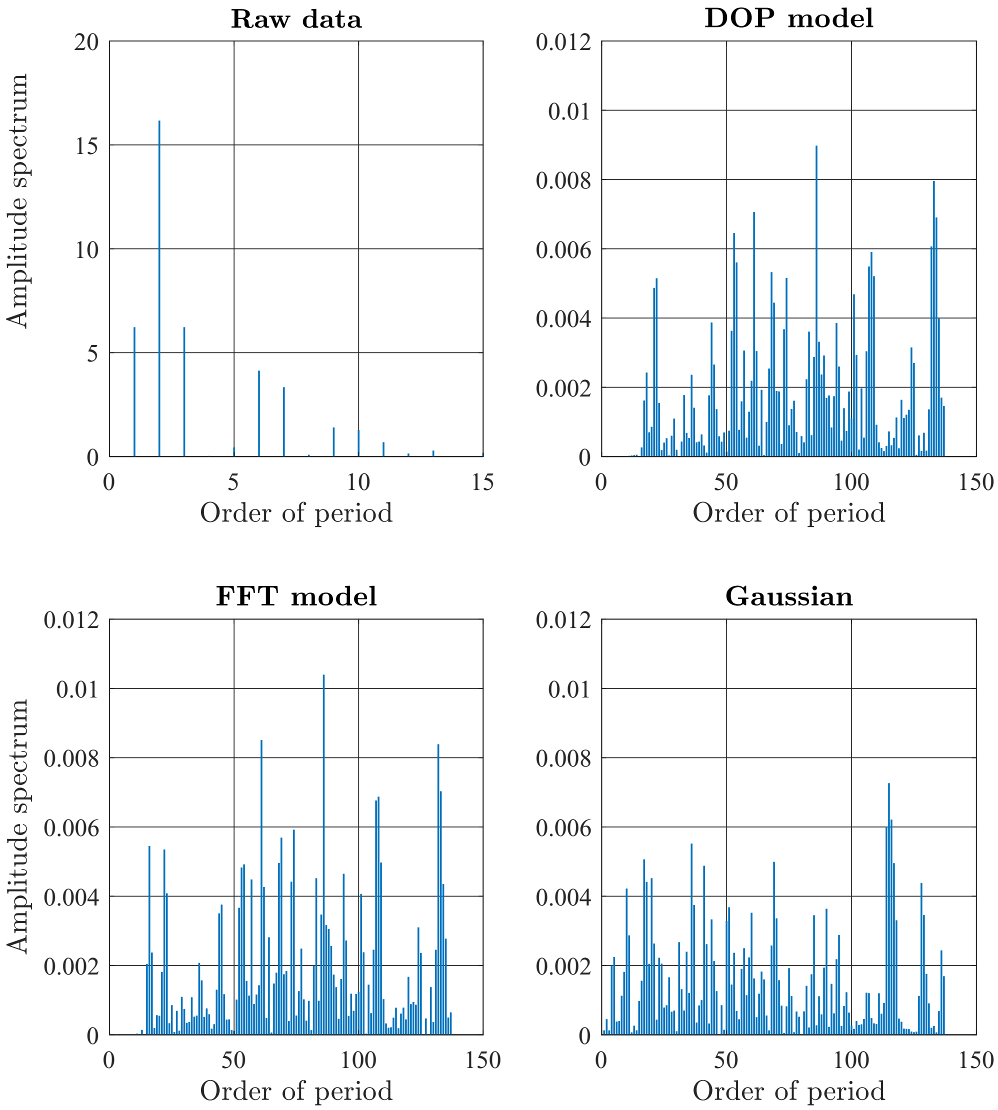

Figure C.3: Results of the FFT spectral analysis for Run A (assembly tolerances according to Table 4.1) **Upper Left Plot:** raw measurement data, **Upper Right Plot:** Residual angle error of DOP filter model, **Lower Left Plot:** Residual angle error of FFT filter model and **Lower Right Plot:** of a Gaussian distribution with the same sample size.

C.1.2 Run B

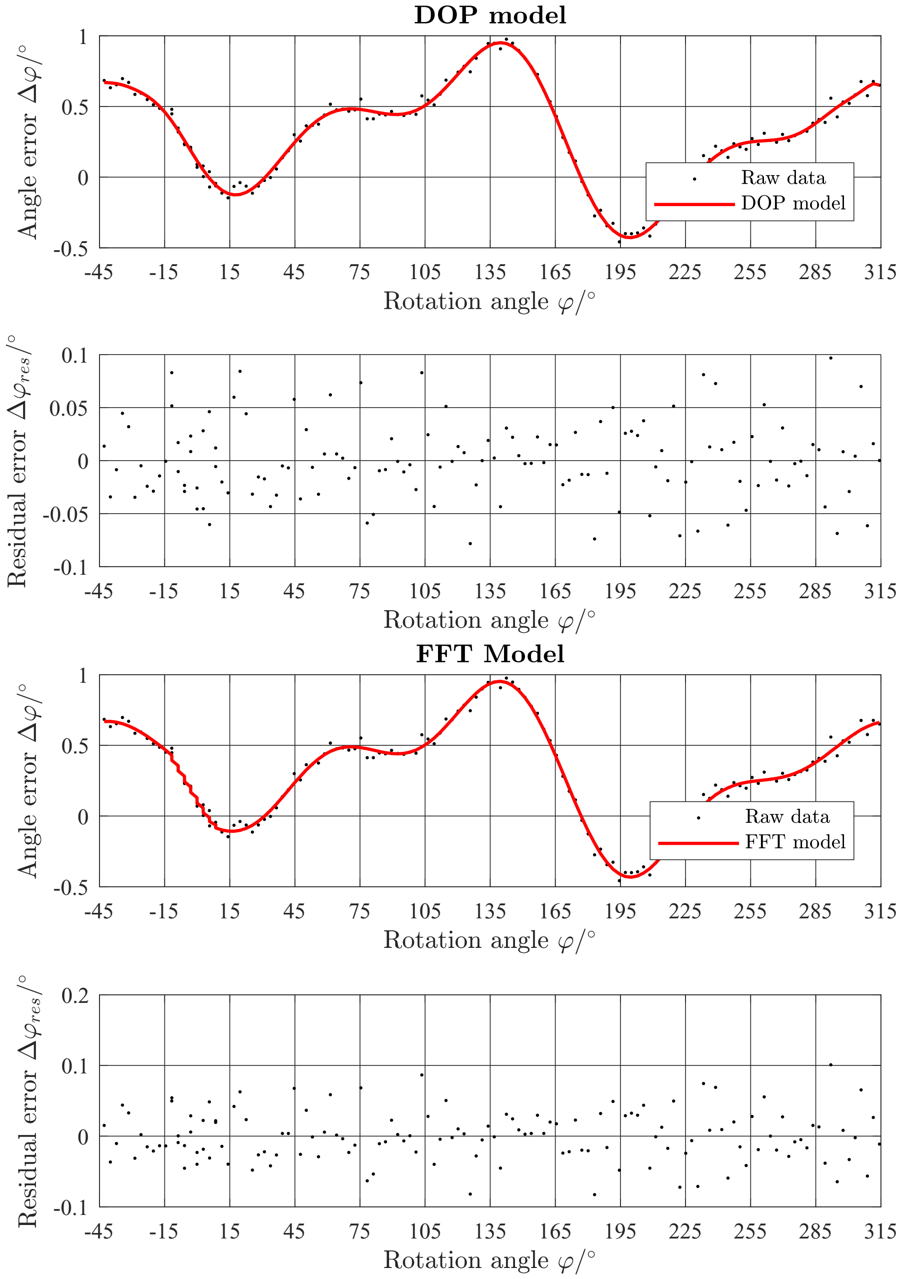

Figure C.4: Results of the DOP and FFT filter models for Run B (assembly tolerances according to Table 4.1) . **Top Plot:** Uncalibrated angle error $\Delta\varphi$ and DOP filter model results over rotation angle φ. **Upper Central Plot:** Resulting residual error $\Delta\varphi_{res}$ of DOP model to raw data. **Lower Central Plot:** $\Delta\varphi$ and FFT filter model results over φ. **Bottom Plot:** $\Delta\varphi_{res}$ of FFT model to raw data.

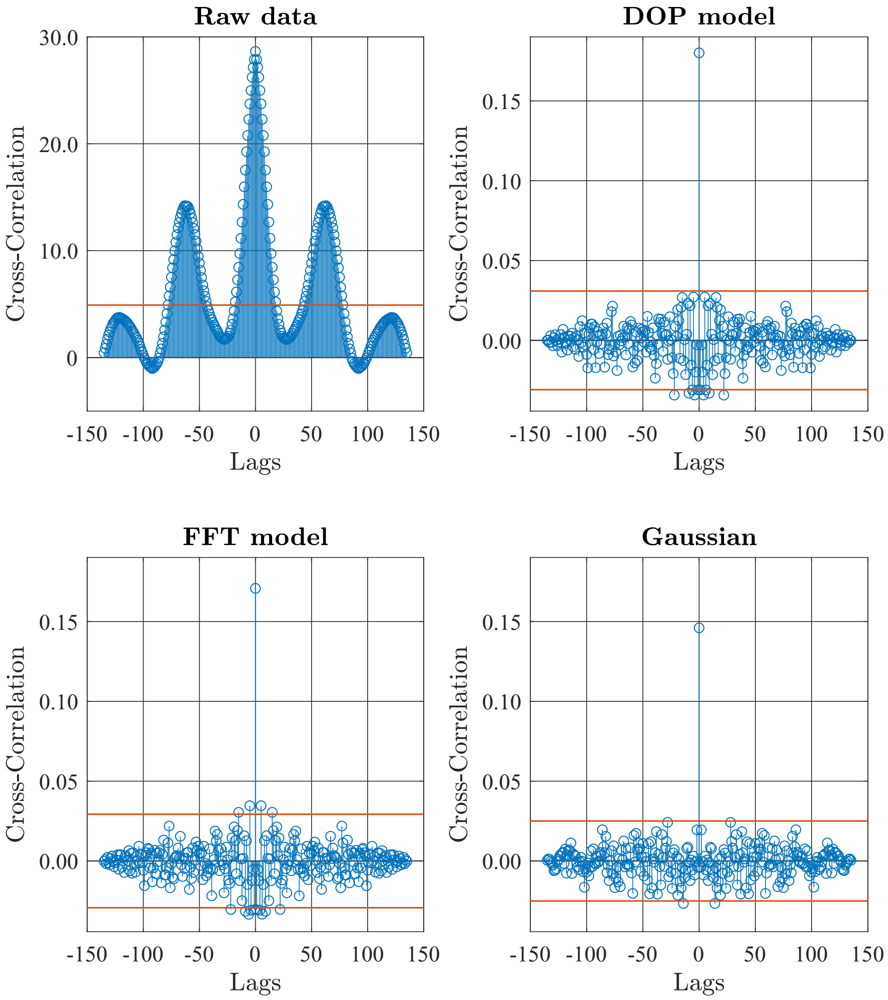

Figure C.5: Cross-Correlation functions for Run B (assembly tolerances according to Table 4.1) of **Upper Left Plot:** raw measurement data, **Upper Right Plot:** Residual angle error of DOP filter model, **Lower Left Plot:** Residual angle error of FFT filter model and **Lower Right Plot:** of a Gaussian distribution with the same sample size.The red lines indicate the 2σ confidence interval for the given sample size.

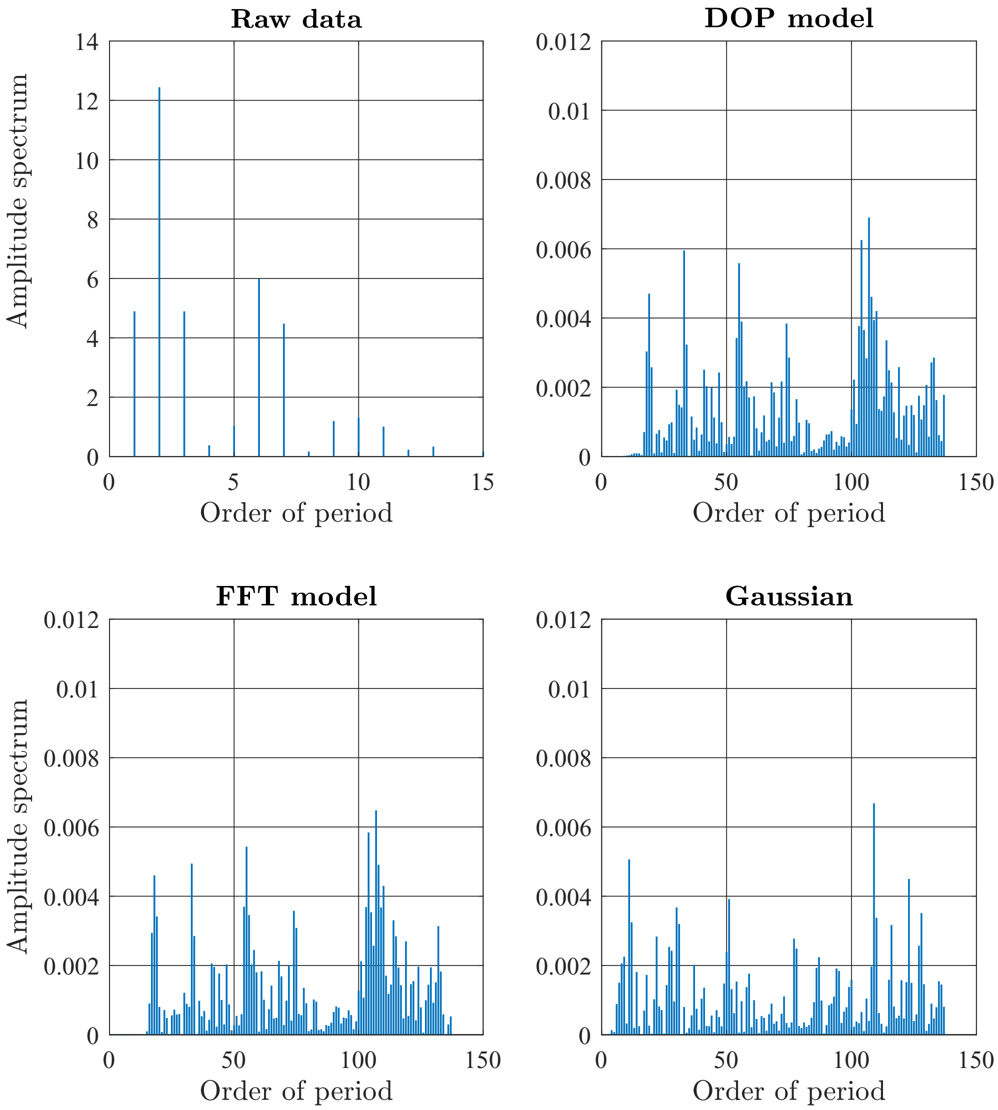

Figure C.6: Results of the FFT spectral analysis for Run B (assembly tolerances according to Table 4.1) **Upper Left Plot:** raw measurement data, **Upper Right Plot:** Residual angle error of DOP filter model, **Lower Left Plot:** Residual angle error of FFT filter model and **Lower Right Plot:** of a Gaussian distribution with the same sample size.

C.1.3 Run D

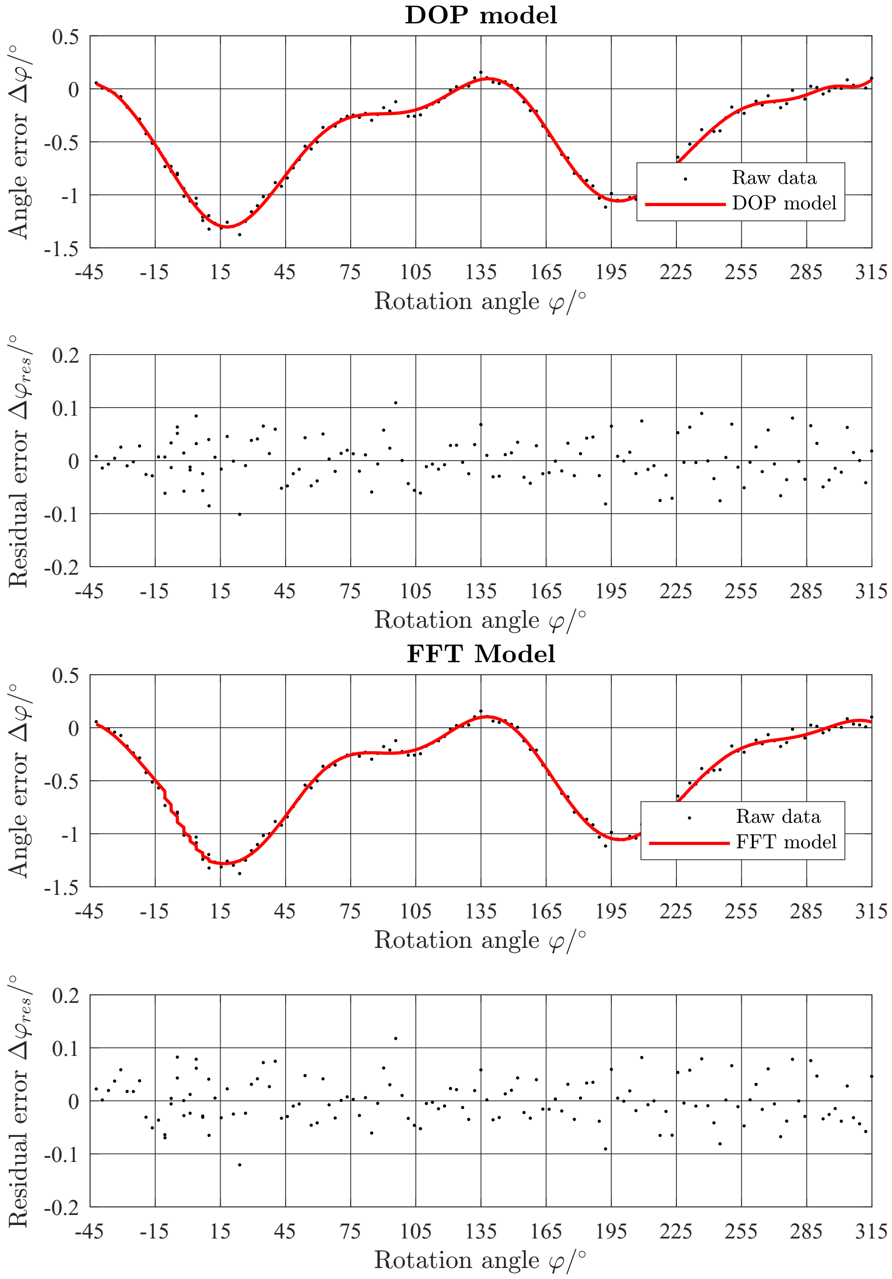

Figure C.7: Results of the DOP and FFT filter models for Run D (assembly tolerances according to Table 4.1) . **Top Plot:** Uncalibrated angle error $\Delta\varphi$ and DOP filter model results over rotation angle φ. **Upper Central Plot:** Resulting residual error $\Delta\varphi_{res}$ of DOP model to raw data. **Lower Central Plot:** $\Delta\varphi$ and FFT filter model results over φ. **Bottom Plot:** $\Delta\varphi_{res}$ of FFT model to raw data.

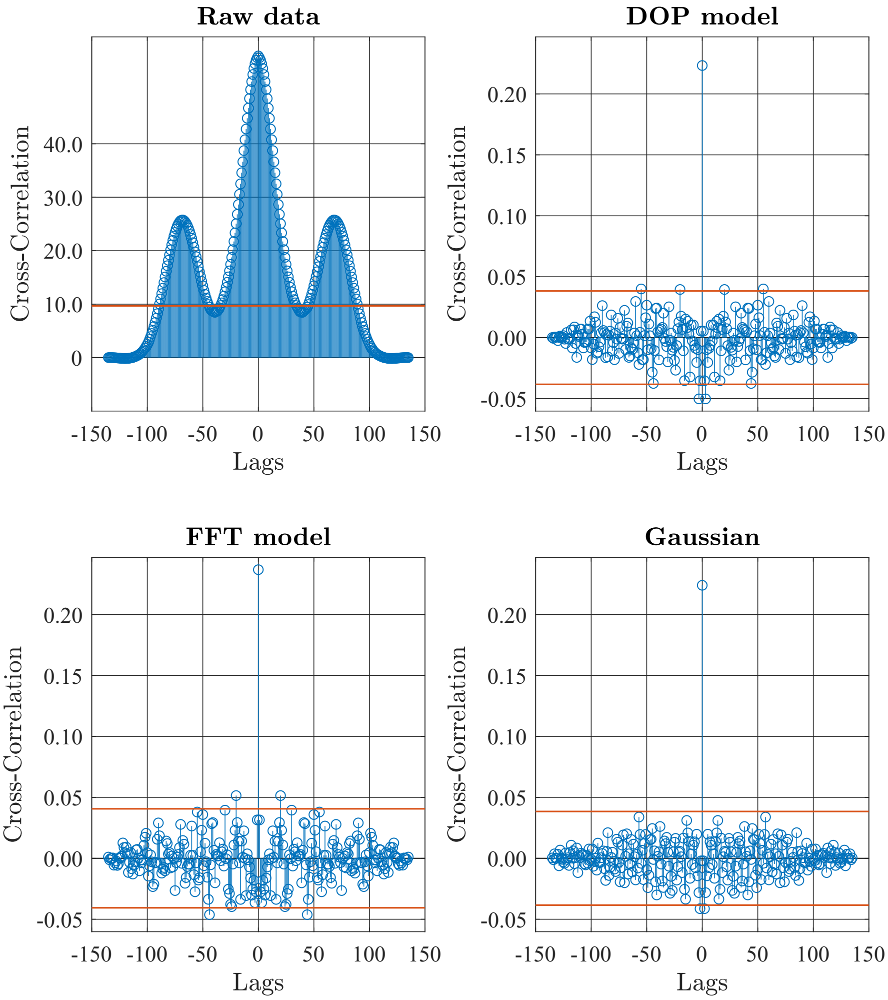

Figure C.8: Cross-Correlation functions for Run D (assembly tolerances according to Table 4.1) of **Upper Left Plot:** raw measurement data, **Upper Right Plot:** Residual angle error of DOP filter model, **Lower Left Plot:** Residual angle error of FFT filter model and **Lower Right Plot:** of a Gaussian distribution with the same sample size.The red lines indicate the 2σ confidence interval for the given sample size.

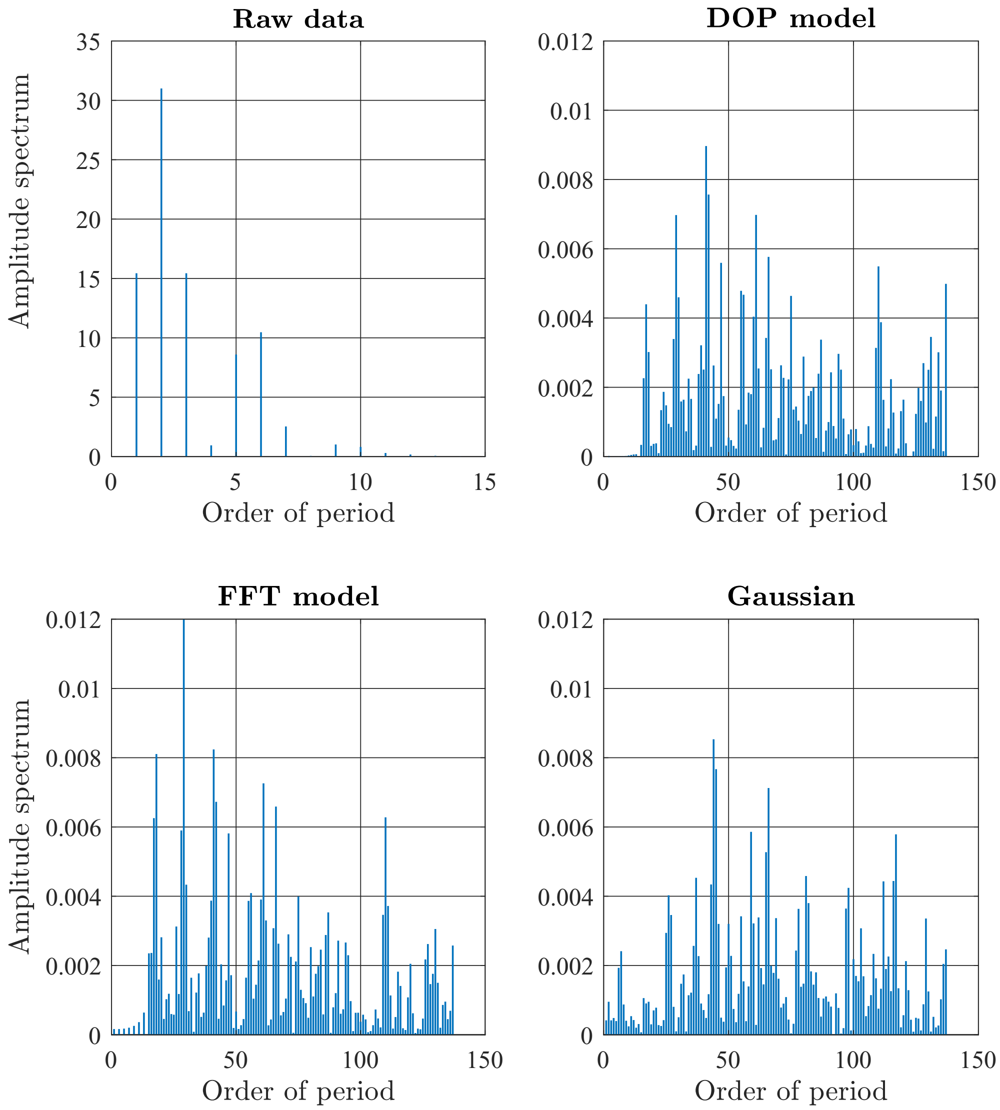

Figure C.9: Results of the FFT spectral analysis for Run D (assembly tolerances according to Table 4.1) **Upper Left Plot:** raw measurement data, **Upper Right Plot:** Residual angle error of DOP filter model, **Lower Left Plot:** Residual angle error of FFT filter model and **Lower Right Plot:** of a Gaussian distribution with the same sample size.

C.1.4 Run E

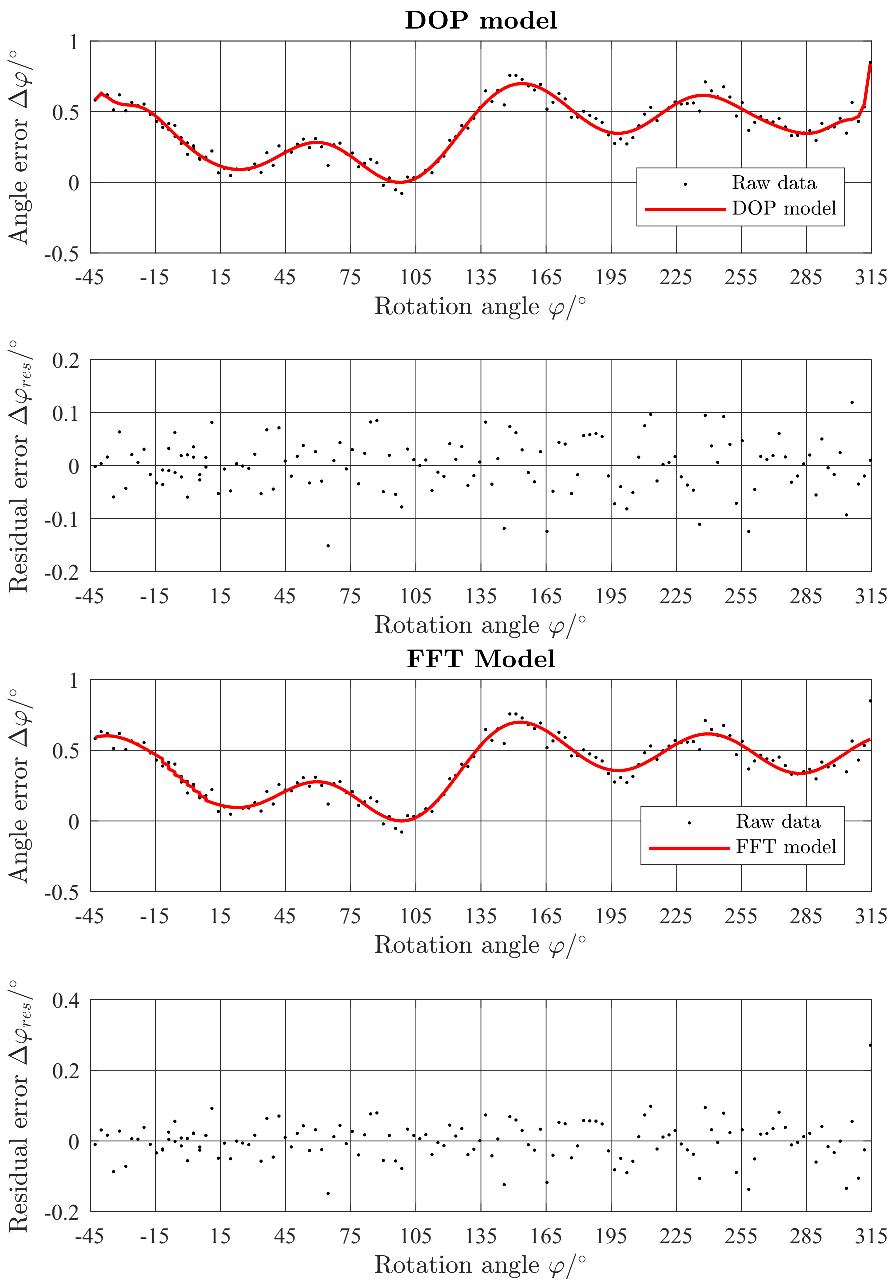

Figure C.10: Results of the DOP and FFT filter models for Run E (assembly tolerances according to Table 4.1) . **Top Plot:** Uncalibrated angle error $\Delta\varphi$ and DOP filter model results over rotation angle φ. **Upper Central Plot:** Resulting residual error $\Delta\varphi_{res}$ of DOP model to raw data. **Lower Central Plot:** $\Delta\varphi$ and FFT filter model results over φ. **Bottom Plot:** $\Delta\varphi_{res}$ of FFT model to raw data.

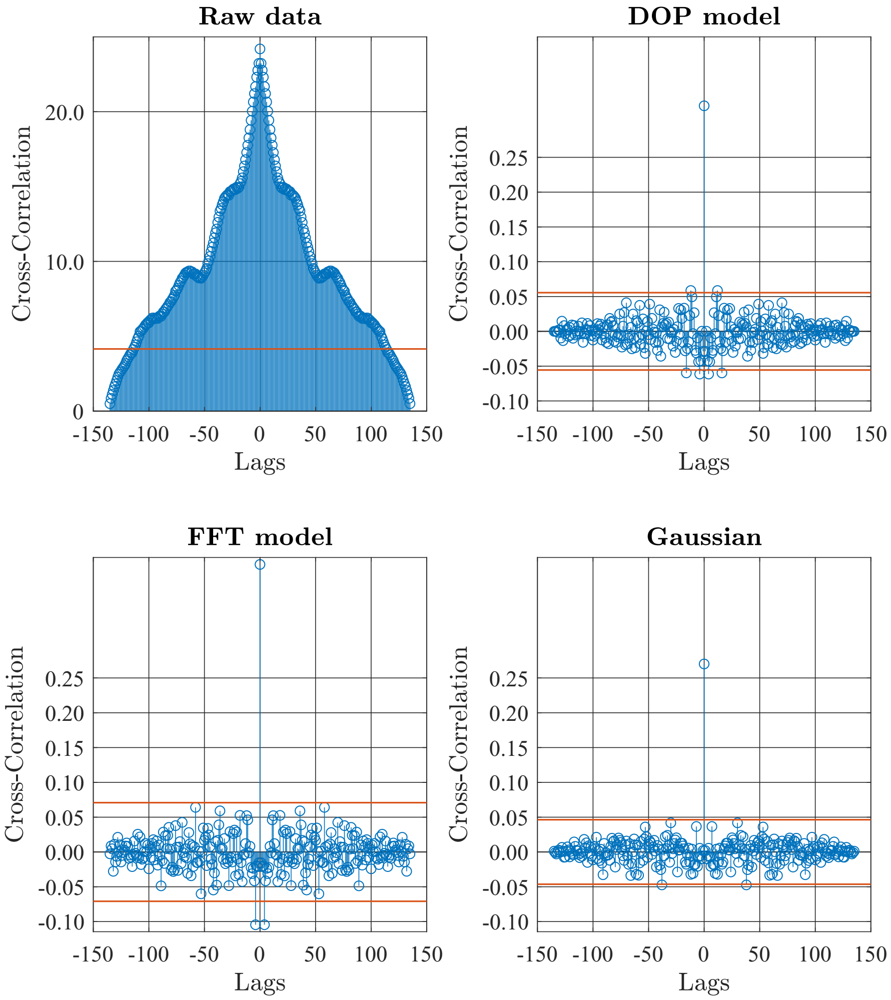

Figure C.11: Cross-Correlation functions for Run E (assembly tolerances according to Table 4.1) of **Upper Left Plot:** raw measurement data, **Upper Right Plot:** Residual angle error of DOP filter model, **Lower Left Plot:** Residual angle error of FFT filter model and **Lower Right Plot:** of a Gaussian distribution with the same sample size.The red lines indicate the 2σ confidence interval for the given sample size.

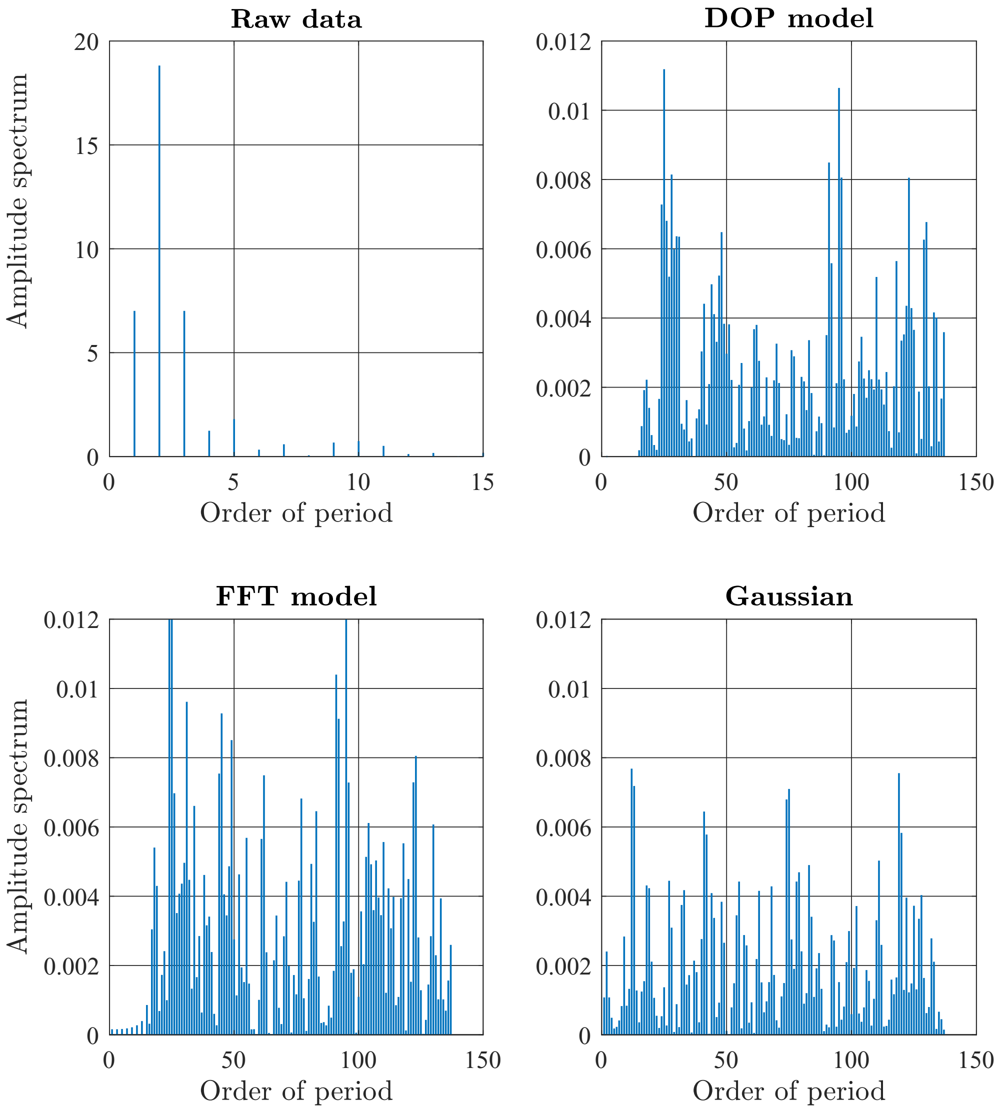

Figure C.12: Results of the FFT spectral analysis for Run E (assembly tolerances according to Table 4.1) **Upper Left Plot:** raw measurement data, **Upper Right Plot:** Residual angle error of DOP filter model, **Lower Left Plot:** Residual angle error of FFT filter model and **Lower Right Plot:** of a Gaussian distribution with the same sample size.

C.1.5 Run F

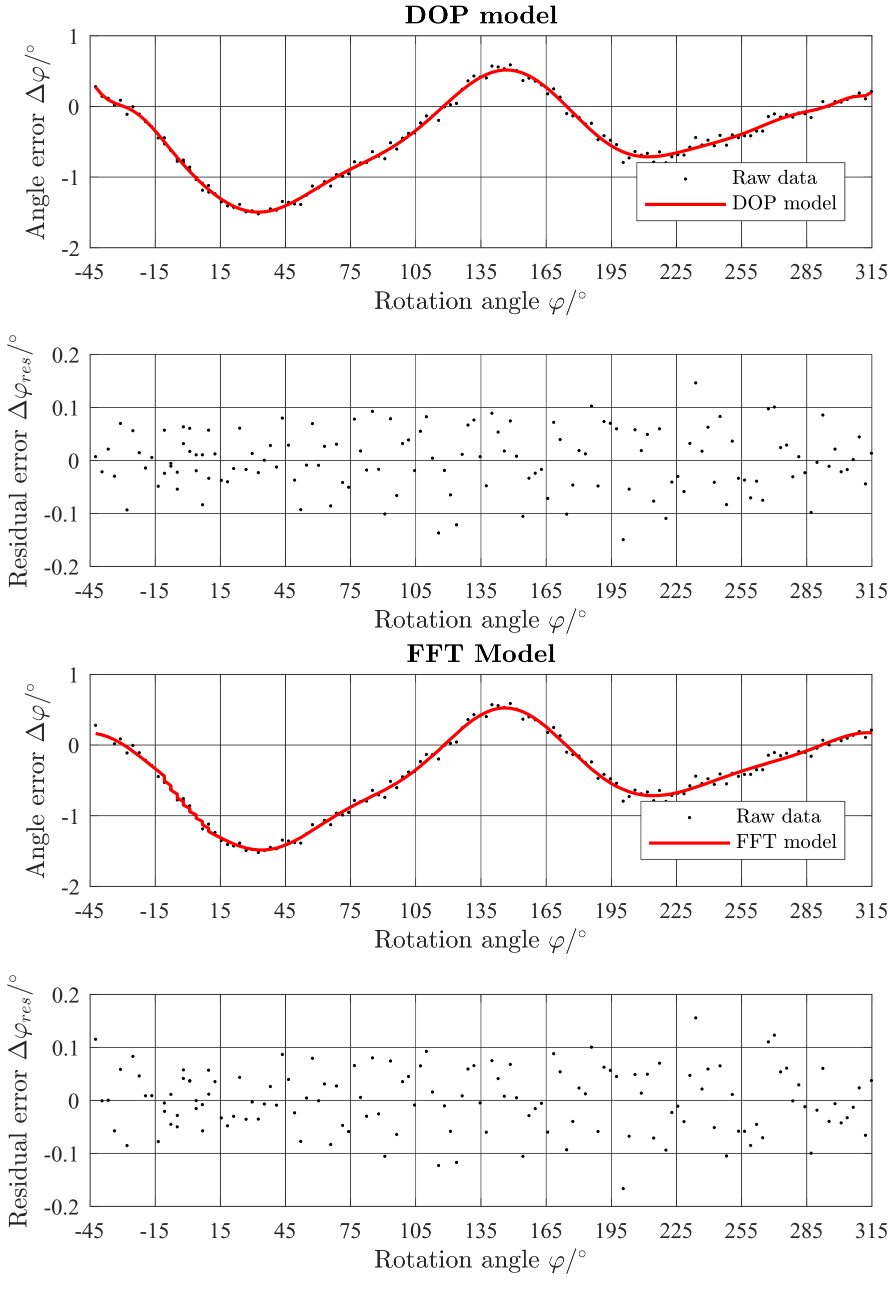

Figure C.13: Results of the DOP and FFT filter models for Run F (assembly tolerances according to Table 4.1) . **Top Plot:** Uncalibrated angle error $\Delta\varphi$ and DOP filter model results over rotation angle φ. **Upper Central Plot:** Resulting residual error $\Delta\varphi_{res}$ of DOP model to raw data. **Lower Central Plot:** $\Delta\varphi$ and FFT filter model results over φ. **Bottom Plot:** $\Delta\varphi_{res}$ of FFT model to raw data.

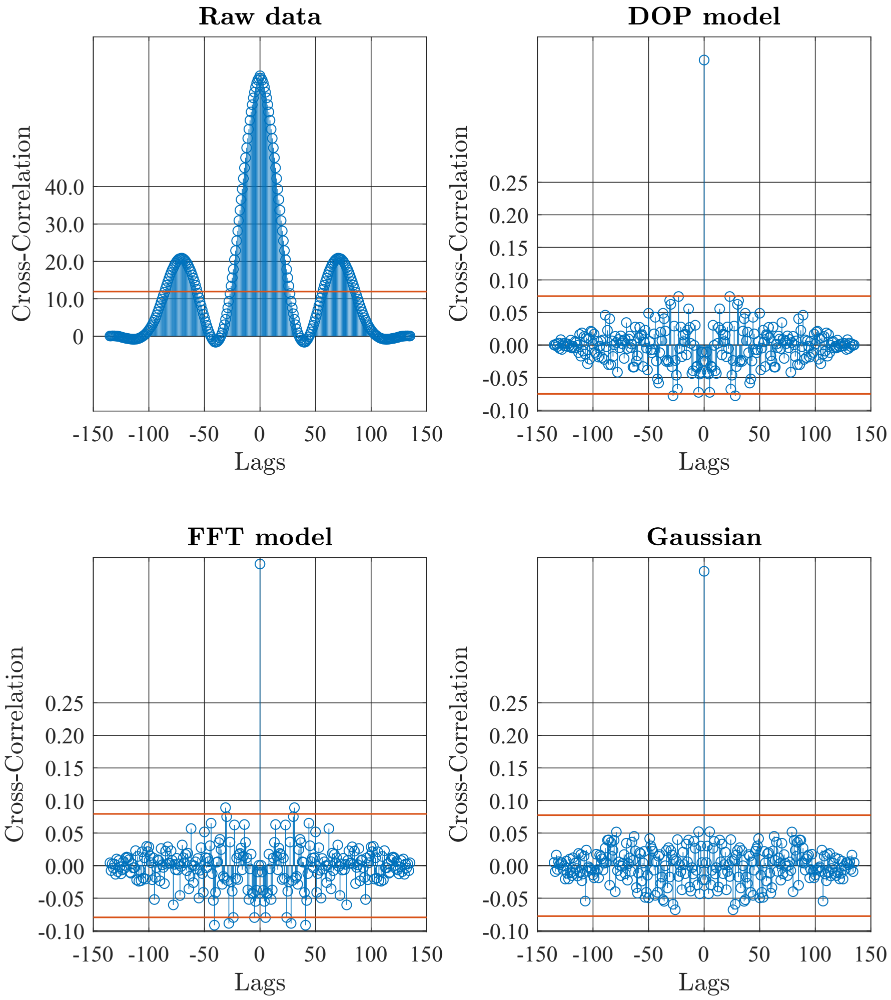

Figure C.14: Cross-Correlation functions for Run F (assembly tolerances according to Table 4.1) of **Upper Left Plot:** raw measurement data, **Upper Right Plot:** Residual angle error of DOP filter model, **Lower Left Plot:** Residual angle error of FFT filter model and **Lower Right Plot:** of a Gaussian distribution with the same sample size.The red lines indicate the 2σ confidence interval for the given sample size.

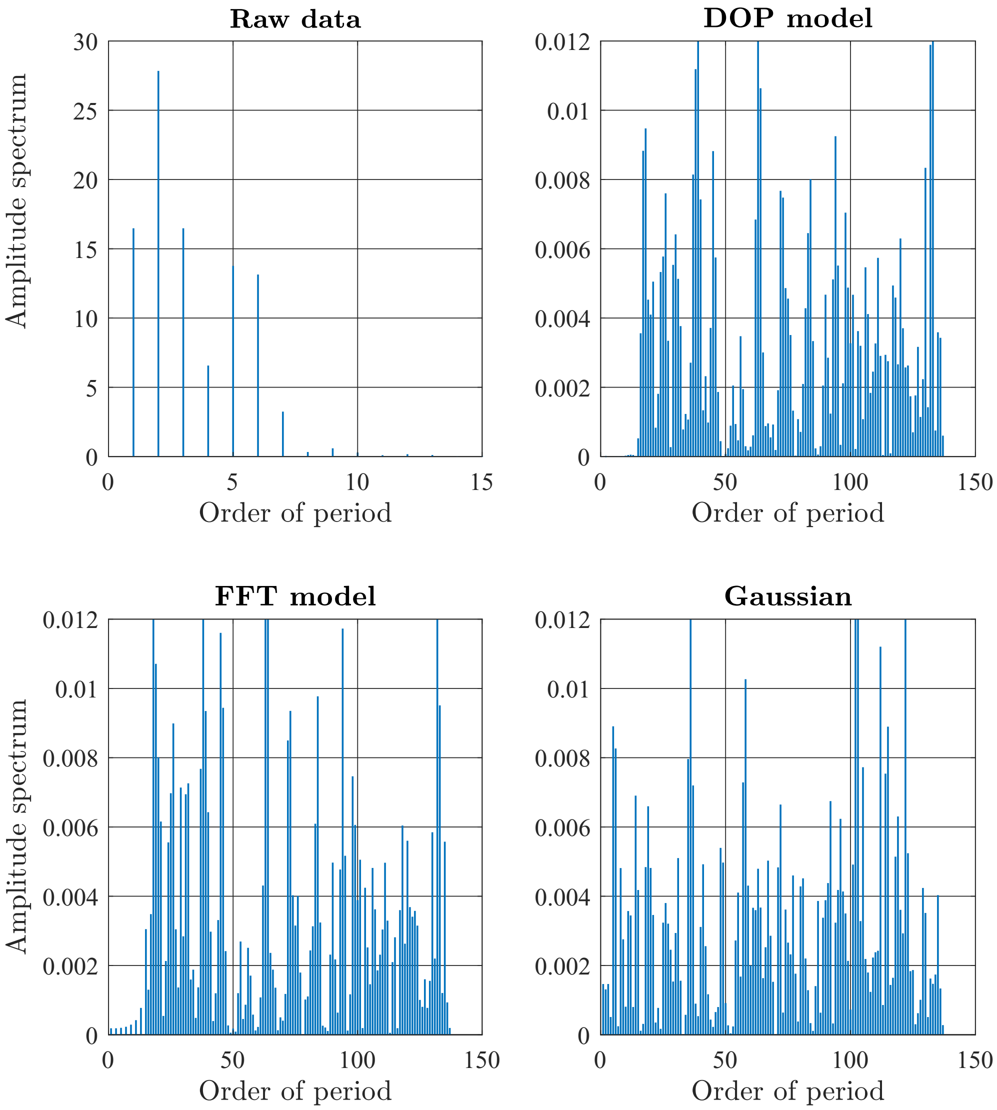

Figure C.15: Results of the FFT spectral analysis for Run F (assembly tolerances according to Table 4.1) **Upper Left Plot:** raw measurement data, **Upper Right Plot:** Residual angle error of DOP filter model, **Lower Left Plot:** Residual angle error of FFT filter model and **Lower Right Plot:** of a Gaussian distribution with the same sample size.