Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterthesis

Oleg Petrak

Signal processing algorithms for magnetic sensor arrays

Fakultät Technik und Informatik
Department Informations- und Elektrotechnik

Faculty of Engineering and Computer Science
Department of Information and Electrical Engineering

Oleg Petrak

Signal processing algorithms for magnetic sensor arrays

Masterthesis eingereicht im Rahmen der Masterprüfung
im Masterstudiengang Automatisierung
am Department Informations- und Elektrotechnik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. -Ing. Karl-Ragmar Riemschneider
Zweitgutachter: Prof. Dr. Klaus Jünemann

Abgegeben am 26. Dezember 2018

Oleg Petrak

Title of the paper

Signal processing algorithms for magnetic sensor arrays

Keywords

Magnetic sensor arrays, Stray field suppression, angular error correction, magnetic field approximation, curvature extraction, magnetic encoder misalignment compensation, projective transformations, geometric distortion correction, THD reduction, 2D-DFT, spatial frequencies evaluation, image processing algorithms, magnetic sensors in automotive applications

Abstract

This thesis deals with the development of signal processing algorithms for magnetic angular position sensor arrays. Multiple algorithms for stray field suppression and compensation of misalignment effects for angular accuracy enhancement including a teach-in procedure have been developed. Their practical feasibility has been tested on the basis of simulation data. Finally, based on obtained results algorithms for implementation on an ASIC are proposed.

Oleg Petrak

Thema der Masterthesis

Signalverarbeitungsalgorithmen für magnetische Sensorarrays

Stichworte

Magnetische Sensorarrays, Streufeldunterdrückung, Winkelfehlerkorrektur, Approximation des magnetischen Feldes, Krümmungsextraktion, Kompensation des Ablagenfehlers des Gebermagneten, projektive Transformation, Verzeichnungskorrektur, THD Reduktion, 2D-DFT, Auswertung der Ortsfrequenzen, Bildverarbeitungsalgorithmen, magnetische Sensoren in Automotive-Applikationen

Kurzzusammenfassung

Diese Arbeit befasst sich mit der Entwicklung von Signalverarbeitungsalgorithmen für magnetische Winkelsensorarrays. Verschiedene Algorithmen für Streufeldunterdrückung und Kompensation des Fehlausrichtungseffektes für Erhöhung der Winkelgenauigkeit, verbunden mit einer Lernphase, werden entwickelt. Ihre praktische Tauglichkeit wird anhand der Simulationsdaten getestet. Basierend auf den Testresultaten wird anschließend ein Vorschlag für die Implementation der Algorithmen auf einem ASIC gegeben.

Acknowledgments

With the following lines I would like to express my gratitude, which applies to all persons, who have supported me during my master thesis.

I would like to express my deep gratitude to Prof. Dr.-Ing. Karl-Ragmar Riemschneider for his supervision as first examiner as well as for the given opportunity to conduct my thesis and extend my knowledge in such an interesting area.
Furthermore, I would like to express my gratitude to Prof. Dr. Klaus Jünemann for his assessment of this thesis as a second examiner as well as for specific discussions and his mathematical assistance.
I am particularly grateful for the assistance given by Dr. Stephan Marauska. His continuous guidance during my work within the project, constructive and suggestions during the planning and execution of investigations were supremely helpful. His assistance and help are greatly appreciated.
My grateful thank is also extended to M. Sc. Thorben Schüthe for his valuable support, numerous tips and advices as well as constructive discussions.
I would like to offer my special thanks to Dr. Valentin Roscher for his intense support during the finalization of this work.
Furthermore I would like to thank Dipl. -Ing. Günter Müller for investing his valuable time into the proofreading of this work and useful recommendations for the structure of this thesis.
Last but not least, I would like to thank all colleagues from the ISAR research team as well as all project partners who have supported and encouraged me during the creation of this thesis.

Contents

List of abbreviations **10**

List of symbols **12**

1 Introduction **14**
1.1 Motivation . . . 14
1.2 Objectives . . . 15
1.3 Thesis structure . . . 16

2 Magnetic sensor systems **17**
2.1 Magnetic sensors in automotive applications . . . 17
2.2 Electronic composition and data acquisition of magnetic sensor . . . 19
2.3 Magnetoresistive sensors . . . 21
2.3.1 Anosotropic magnetoresistance (AMR) . . . 22
2.3.2 Giant magnetoresistance (GMR) . . . 24
2.3.3 Tunnel magnetoresistance (TMR) . . . 26
2.4 Hall sensors . . . 28
2.5 Technology comparison . . . 29
2.6 Sensor requirements . . . 30
2.6.1 Encoder magnet - sensor system and mechanical misalignment . . . 31
2.6.2 Stray field suppression . . . 32
2.7 Comparison of a single sensor and sensor array system . . . 33

3 Development of algorithms **35**
3.1 Required system properties . . . 36
3.2 Signal processing approach . . . 36
3.2.1 Functional split . . . 37
3.2.2 Signal processing concept . . . 38
3.2.3 Calibration stages . . . 39
3.3 Algorithms for angle information extraction . . . 41
3.3.1 Basic principle for angle information extraction . . . 41
3.3.2 2D-DFT based algorithm . . . 43
3.3.3 Approximation approaches . . . 48
3.3.4 Image processing filter based algorithms . . . 56
3.3.5 Effort comparison . . . 60

3.4 Preprocessing . . . 62
3.4.1 Median filter . . . 62
3.4.2 Butterworth low pass filter . . . 64
3.5 Handling of misalignment effects . . . 65
3.5.1 Misalignment detection . . . 65
3.5.2 Realignment of field components . . . 67
3.6 Angular error compensation by teach-in phase on spatial domain . . . 68
3.6.1 General simplification for magnitude determination . . . 72
3.6.2 Butterworth low pass filter based algorithm . . . 74
3.6.3 Gradient filter based algorithms . . . 75
3.6.4 Approximation based algorithms . . . 79
3.7 Angular error correction by angular teach-in phase . . . 82
3.7.1 Linear angular error correction algorithm based on homography . . . 84
3.7.2 Nonlinear angular error correction algorithm based on geometric distortion cancelation . . . 94
3.8 Stray field detection . . . 100
3.9 Total algorithmic and mathematical chain . . . 102

4 Simulation and evaluation **104**
4.1 Simulation setup . . . 104
4.1.1 Encoder magnets . . . 105
4.1.2 Coordinate system, units and initial values . . . 106
4.1.3 Simulation limits and parameter variation . . . 109
4.2 Evaluation software . . . 112
4.2.1 Import, interference and acquisition of simulation data . . . 112
4.2.2 Data and program flow . . . 114
4.2.3 Algorithms implementation . . . 115
4.2.4 Relevant functions for investigations . . . 116
4.3 Mathematical descriptions for evaluations . . . 128
4.3.1 Angular accuracy . . . 128
4.3.2 SNR sweep dependent angular accuracy . . . 128
4.3.3 AMM localization accuracy . . . 129
4.3.4 Optimization and field deformation indication . . . 130
4.3.5 Signal quality analysis . . . 131

5 Investigations **133**
5.1 Field conditions analysis . . . 133
5.2 Preprocessing . . . 138
5.2.1 Filtering in spatial frequency domain . . . 138
5.2.2 Median filter application . . . 144
5.3 Optimizations . . . 147
5.3.1 LoG filter bandwidth optimization . . . 147

5.3.2 Exponent optimization for modified approximation approach 150
5.4 Preliminary investigations 152
5.4.1 Misalignment test 152
5.4.2 Quantization noise sensitivity test 153
5.5 Applied algorithm conditions 155
5.6 Investigation plan 157
5.6.1 Content of block 1 159
5.6.2 Content of block 2 161
5.6.3 Content of block 4 162
5.7 Investigation results 163
5.7.1 Proof of concept 163
5.7.2 Misalignment test 176
5.7.3 Tracking algorithms test 181
5.7.4 Extended algorithms test 187
5.7.5 Angular error correction 192
5.8 Results summary, discussions and proposals 211
5.8.1 Algorithms implementation proposal 213
5.8.2 Proposal for encoder system 214

6 Summary and outlook **215**
6.1 Summary 215
6.1.1 Results of algorithms 216
6.1.2 Contribution to the project 216
6.1.3 Obstacles and difficulties 217
6.2 Outlook 218
6.2.1 Independent proof of results 218
6.2.2 Real data investigations 218
6.2.3 Software extension 218
6.2.4 Algorithm extensions and improvements 218
6.2.5 Further algorithm approaches 220
6.2.6 Further investigations 221
6.2.7 Extended system proposal 222

Bibliography **223**

List of Tables **232**

List of Figures **234**

A Task description **242**

B Least squares method **246**
B.1 Problem definition 246

B.2 Least squares solving methods . . . 247
B.2.1 Direct normal equations solution . . . 247
B.2.2 QR factorization . . . 247
B.2.3 Singular value decomposition (SVD) . . . 248
B.3 Least squares alternatives . . . 249

C Development of algorithms **250**
C.1 Saturated and not saturated sensor array comparison . . . 250
C.2 Application of Template Matching algorithm on a saturated sensor array system 251
C.3 2D-DFT algorithm for angle information extraction for complex input signals . . 255

D Evaluation software **257**
D.1 Front-end . . . 257
D.1.1 Graphical User Interface: overview . . . 257
D.1.2 Graphical User Interface: control and evaluation panels . . . 258
D.2 Back-end . . . 259
D.2.1 Main GUI update function . . . 259
D.2.2 Angle information extraction . . . 261
D.2.3 Tracking algorithms . . . 268
D.2.4 Angular error correction . . . 271
D.2.5 Data and program flow . . . 277
D.3 Relevant functions for investigations . . . 279
D.3.1 Analog-to-Digital Converter . . . 279
D.3.2 Signal-to-noise ratio . . . 280
D.3.3 Gain and offset deviations . . . 281
D.3.4 Fixed-point arithmetic . . . 282

E Preprocessing **284**
E.1 Application of Butterworth low pass filter . . . 284
E.1.1 Zero padded data arrays . . . 284
E.2 Median filter application . . . 285

F Optimizations **286**
F.1 Exponent n optimization . . . 286
F.1.1 Sphere magnet . . . 286
F.1.2 Disc magnet . . . 287
F.2 LoG filter bandwidth optimization . . . 287

G Investigation results **291**
G.1 Stray field suppression . . . 291
G.2 Reasons for 2D-DFT extension by dynamic realignment of field components . . 292
G.3 Misalignment sensitivity of algorithms . . . 293
G.4 Reasons for exclusion of the modified approximation algorithm . . . 293

G.5 Proof of concept 296
G.5.1 Subblock 1.0 296
G.5.2 Subblock 1.1 301
G.5.3 Subblock 1.2 311
G.5.4 Misalignment test (block 2) 314
G.5.5 Misalignment robustness (subblock 2.0) 314
G.5.6 Array size and misalignment effect dependency (subblock 2.1) 317
G.6 Trackings algorithms test (block 3) 319
G.6.1 Results for disc magnet 319
G.7 Extended algorithms test (block 4) 320
G.7.1 Extended algorithms test 2 (subblock 4.1) 320
G.8 Angular error correction (block 5) 326
G.8.1 Harmonics cancelation 326
G.8.2 Misalignment compensation: extreme case 328
G.8.3 Misalignment compensation: worst SOA case 335

H Sensor array demonstration system **344**

List of abbreviations

Used abbreviations

ADC	Analog-to-Digital Converter
ASIC	Application Specific Integrated Circuit
ASIL	Automotive Safety Integrity Level
AMR	Anisotropic magnetoresistance
CORDIC	Coordinate Rotation Digital Computer
CIP	Current in plane
CPP	Current perpendicular to plane
dBW	Decibel Watt
DFT	Discrete Fourier transform
DOF	Degrees of freedom
DoG	Derivative of Gaussian
DSP	Digital signal processor
ECU	Electronic Control Unit
EEPROM	Electrically Erasable Programmable Read-Only-Memory
EMI	Electromagnetic Interference
EOL	End of line
FDM	Finite Elements Method
FEM	Finite Elements Method
FFT	Fast Fourier transform
FPGA	Field Programmable Gate Array
GMR	Giant magnetoresistance
GUI	Graphical User Interface

LoG	Laplacian of Gaussian Filter
MTJ	Magnetic tunnel junction
MUX	Multiplexer
OEM	Original Equipment Manufacturer
OP	Operational amplifier
PDF	Probability density function
PWM	Pulse-width modulation
QR	Decomposition of matrix A into product A = QR
SNR	Signal-to-noise ratio
SOA	Safe Operating Area
SV	Spin valve
SVD	Singular value decomposition
THD	Total harmonic distortion
TMR	Tunneling magnetoresistance
xMR	Various magnetoresistance technologies
2D-DFT	Two dimensional discrete Fourier transform

Introduced abbreviations

AMM	Absolute magnitude maximum *Point in the xy-plane of the sensor array of maximum field strength*
MDD	Mean distance deviation *Measure for the deviation of localized AMM points from the expected*
MEFC	Mean error of field components *Measure for the field deformation associated with filtering and approximation*

List of symbols

Physical quantities

Symbol	Description	Unit
$\vec{H}$	Magnetic field	$\frac{A}{m}$
$\vec{B}$	Magnetic flux density (induction)	$T = \frac{Vs}{m^2}$
$\vec{M}$	Magnetization of material	$\frac{A}{m}$
μ	Permeability of a magnetic body	
μ_0	Vacuum permeability	$\frac{4\pi 10^{-7} N}{A^2}$
μ_r	Relative permeability	
χ	Magnetic susceptibility	
F_S	Sampling frequency	Hz

Mathematical symbols and operators

Symbol	Description
P	Projective space
R	Euclidean space
$\mathbb{R}$	Real numbers
$\circ$	Hadamard product (element wise multiplication)
$*$	Convolution operation
$\mathcal{F}\{\}$	Fourier transform operation
$Re\{\}$	Real part of components
$Im\{\}$	Imaginary part of components
mod	Modulo operation
∇	Laplace operator
$\mathbf{E}$	Identity matrix
$\mathbf{R}$	Rotation matrix
$\mathbf{H}$	Homography matrix

Introduced symbols and operators

φ	Angular encoder magnet position [°]
$\varphi_{expected}$	Failure-free angle representing the angular encoder magnet position [°]
$\varphi_{calculated}$	Calculated angle representing the angular encoder magnet position [°]
φ_{err}	Angular error [°]
$\mathbf{H_x}$	Matrix containing ambient magnetic field components in x-direction
$\mathbf{H_y}$	Matrix containing ambient magnetic field components in y-direction
$\mathbf{B_x}$	Matrix containing measured magnetic field components in x-direction
$\mathbf{B_x}$	Matrix containing measured magnetic field components in y-direction
$\mathbf{M_f}$	Matrix containing magnitudes of measured field components
D	Denominator matrix containing field weakening terms
A	Design matrix for an estimation of model parameters
b	Vector containing the experiment data for model parameter estimation
x	Vector containing model parameter
X	Matrix containing x-coordinates of the sensor array
Y	Matrix containing y-coordinates of the sensor array
x, y	Coordinates of the xy-plane of the sensor array
x, y	Sensor signals used for angle calculation (also denoted as $x(n)$ and $y(n)$)
p	Vector containing coordinates of a 2D point (here the x and y signals act as coordinates)
$\mathbf{p}'$	Vector containing corrected coordinates of a 2D point: correspondence to **p** (here the x' and y' signals act as coordinates of a virtual plane)
.	Element wise operation (same as Hadamard product, but used for square and square root operations)

1 Introduction

1.1 Motivation

Magnetic sensor systems are of great importance in industrial, power and consumer electronics as well as in automotive, due to its contactless measurement capability. Especially in automotive applications the magnetic sensors are widely used for position and speed sensing of rotating mechanical systems. These types of sensors are versatilely used in engine electronics and its control, anti-lock braking systems and in applications of comfort area [81].
Hamburg University of Applied Sciences cooperates with industry partners on research project called **ISAR** -„Signal processing for integrated sensor arrays based on tunnel magnetoresistive effect for application in automotive electronics“, which is funded by the German Federal Ministry of Education and Research. In this project, signal processing and ASIC system architecture for a novel type of sensor constructed as an array is developed [25].
The application of a sensor array allows a new type of magnetic field acquisition. In this way the information about the field of an applied magnetic encoder is represented by its spatial position. The spatial resolution obtained by a sensor array provides a possibility for stray field suppression as well as a high potential for error corrections and a system accuracy enhancement. The capability for stray field suppression represents recently formed and obligatory requirement for magnetic sensors of the next generation[1]. In order to be able to use the concept for stray field suppression based on the evaluation of spatial field information, newer magnetic encoder systems with a high field inhomogeneity need to be used. Furthermore and in general, by using a sensor array an enhancement of the system reliability, stability and redundancy is obtainable.
The main challenges within the framework of the project ISAR are:

1. Development of suitable algorithms for the processing of measured field data in order to obtain the desired functional features for coverage of system requirements.
2. Hardware architecture design and functional implementation of algorithms on an ASIC.
3. Development of analog data acquisition techniques which satisfy the dynamic requirements of applications.

This master thesis covers the first part of these challenges. Within the context of this thesis, suitable and practically feasible signal processing algorithms for magnetic sensor arrays shall be developed. Previous work of project members shall be considered [53], [90], [75], [52].

[1] International standard ISO 11452-8:2015(E) Road vehicles - Component test methods for electrical disturbances from narrow band radiated electromagnetic energy - Part 8: Immunity to magnetic fields [22]

The focus with respect to the application is set on the determination of angular position in specific automotive applications. It is assumed that the sensor array system exhibits linear behavior.
The following three main targets are aspired in line with this work:

- Stray field suppression
- Precise angle information extraction
- Angular error correction

In order to achieve the suppression of stray field and system accuracy enhancement, the application of signal processing techniques such as two-dimensional Fourier transformation, image processing, regression methods and analysis are followed in the context of this work. Especially the mechanical misalignment between the sensor system and the magnetic encoder plays a crucial role for the accuracy of the angular position sensing device. To compensate the misalignment effects and correct the angular error associated with it, it is allowed to include a teach-in phase into the total approach. In combination with rating of arithmetical complexity required by the algorithms and their individual performance the most suitable and appropriate algorithm candidates shall be proposed for the implementation on the ASIC within the framework of project continuation.

1.2 Objectives

From the set targets the following questions can be derived, which are answered in the context of this thesis:

1. Is it possible to rely on the developed concepts practically in accordance to the current technological state?
2. Which signal quality measured in SNR is required to be able to obtain the desired angular accuracy of the sensor array system?
3. What is the necessary amount of sensor elements on an available sensor die area in combination with the ADC resolution which fulfills the accuracy requirement?
4. How does the mechanical misalignment between the sensor array and the magnetic encoder affect the system accuracy? How is the robustness of algorithms against mechanical misalignment effects such as tilt and translation of the encoder magnet?
5. What are the misalignment limits of encoder magnets which are acceptable within the set accuracy requirements for the sensor system?
6. Is a pre-calibration of the sensor array elements indeed strictly necessary or do the process and fabrication tolerances plays a negligible role, if the information about the angular position of the encoder magnet is extracted from the spatial area?
7. How does the limitation of arithmetical accuracy (limited widths of data registers for arithmetic operations) affect the angular accuracy of the system?

8. Is there indeed a need to change the magnetic encoders in order to obtain stray field robustness or can also nowadays commercially used encoder magnets of smallest dimensions still be used?
9. Is it possible to compensate misalignment effects and obtain a satisfied angular accuracy enhancement by additional teach-in procedures and application of angular error correction algorithms so that the accuracy requirements are met?

1.3 Thesis structure

As an introduction into the subject area, Chapter 2 explains the fundamental functionality of magnetic sensors in today's automotive applications. The most important magnetic field sensing technologies are described in detail. Furthermore, the chapter introduces the requirements for angular accuracy of the sensor and the limitations of its operating area in the application. Advantages as well as disadvantages of the novel sensor array system, if it is operating including stray field suppression, are discussed.
In the Chapter 3, the development of the signal processing concept and the algorithms is documented. Multiple algorithm candidates for angle information extraction including stray field suppression were considered within this work. Their assets and drawbacks are compared. Furthermore, development methods for handling of misalignment effects are documented in the chapter. In order to correct the angular error caused by encoder magnet misalignment two developed approaches in combination with a teach-in phase are documented: a) a teach-in procedure which allows the misalignment effect compensation on the spatial domain and b) a spatial domain independent angular error correction method. The mathematical basis and procedures of the developed algorithms and methods are described in detail.
Chapter 4 documents the created simulation setup in accordance to the specified operation area of the sensor. For the investigations two magnet shapes are constructed, a two pole commercial disc encoder magnet and a sphere magnet which posses the maximum field inhomogeneity. An evaluation software including the implementation of the algorithms and a graphical user interface were developed and are documented in this chapter. Furthermore, the mathematical formalisms for evaluation of algorithm results are explained.
In Chapter 5, an extensive investigation of the algorithms for their performance and robustness is presented. At first, the field condition of the encoder magnets is analyzed. Subsequently, pre-investigations as well as optimizations of algorithms parameters are carried out. The created investigation plan in correspondence to the key questions about the system is introduced and the investigations in full accordance to the test plan are carried out. Looking back to the conducted investigations and based on the obtained test results an implementation proposal for the most suitable algorithms is given.
The last chapter summarizes the work. In the form of further work important steps into the direction of further investigations as well as for the algorithmic extensions are proposed and the chapter is completed with a broad outlook.

2 Magnetic sensor systems

This chapter provides an introduction into the theory background of magnetic sensing technologies as well as comprehensible understanding about functionally of magnetic sensors and their specific applications in automotive. It highlights the key physical features of a sensor, which are necessary for the developed algorithms.
At first, the application area of magnetic sensors is covered. The current existing used sensing technologies with regard to position detections in automotive are explained. Afterwards a brief comparison of considered technologies is given and the actual requirements for the sensors are summarized. The potential advantages as well as disadvantages of a magnetic sensor array with respect to the application requirements are discussed.

2.1 Magnetic sensors in automotive applications

In general, the main application areas of magnetic sensors in industrial, automotive and consumer electronics branches is the acquisition of 3D position, off-axis position, linear position, speed and angle position [73], [23]. Especially, in automotive domain the detection of speed and angle position is of particular importance. The focus lies on motor control, turbochargers, transmission, anti-lock braking and steering system applications [68]. Illustration 2.1 points out the related applications.

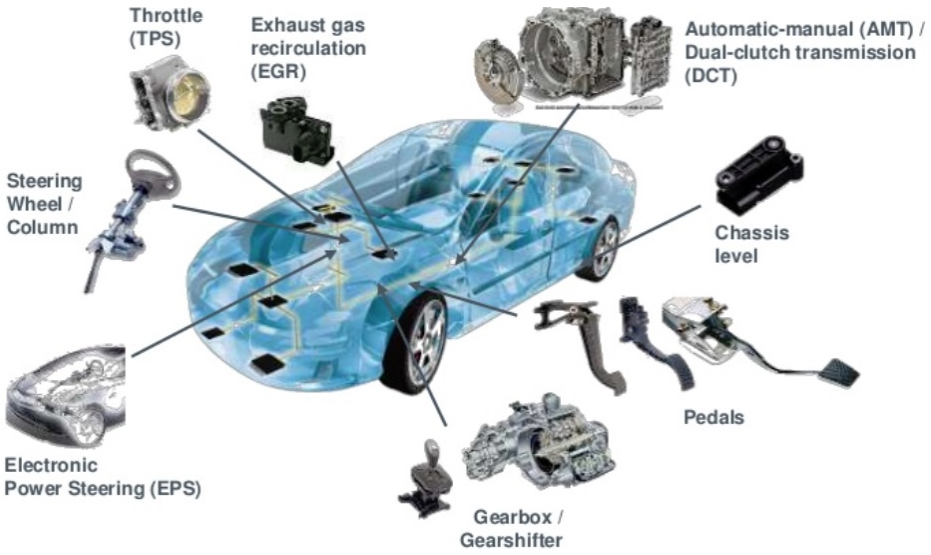


Figure 2.1: Automotive applications for magnetic field sensors [73].

The main advantage of magnetic sensors is a contactless measurement of positions by acquisition of field components generated by an encoder magnet which are measured in the sensor plane. The evaluation of the field of encoder magnet are categorized in two main applications:

- **Angle sensing:** Measurement of angular positions of a encoder magnet. The magnetic field $\vec{H}$ is measured in the x- and y-direction. The measured signals represents the sine and cosine component of the rotating field vector $\vec{H}$. The measurement of strength of the magnetic field $|\vec{H}|$ is therefor not necessary but it needs to be above a particular value in order to excite the sensor adequately, so called saturation. In technical language it is said to be saturated. Sensor documented in [20] is an example for such a system.
- **Speed sensing:** Measuring of rotation of a encoder wheel magnet. The magnetic field $\vec{H}$ is measured in one direction only. In this case the information about the field strength $|\vec{H}|$ is evaluated in one particular direction. Zero crossings of the magnetic field at transitions between two poles of encoder magnet are detected by the sensor. Internal sensor electronic generates pulses, which are evaluated in the ECU for speed calculation. A suitable example for this type of sensing is [21].

Figure 2.2 shows schematically the sensing principle of introduced sensing categories.

Figure 2.2: Rotation measurements by magnetic positions sensors [8]. Left: On axis angle measurement. Center: Angular measurement on circumference. Right: Two configurations of measurement of field of a pole wheel for speed sensing.

The maximal rotational frequencies to be evaluated by an angle sensor in current automotive applications are in a range up to 500 Hz. The required angle resolution with respect to the rotational frequency is usually 1°. Speed sensors operate at frequencies up to of 20 kHz [68]. For the described automotive applications nowadays extended sensing technologies are existing, which offer more application specific versatility and flexibility, e.g so called Tri-Axis® technology[1]. Also, dual chip solutions[2] fabricated on dual isolated die for more redundancy and robustness have been developed in the recent years.

[1] A magnetic sensor technology based on Hall effect capable of 3 axis magnetic field measurement from one sensor device. For details see https://www.melexis.com/en/insights/knowhow/triaxis-position-sensing-solution and [26]

[2] e.g. Infineon TLE5014: magnetic angle sensors with ISO 26262 ASIL C and ISO 26262 ASIL D specifications. https://www.infineon.com/cms/en/product/sensor/magnetic-position-sensor/angle-sensor/tle5014c16/

2.2 Electronic composition and data acquisition of magnetic sensor

A magnetic sensor consists in general of multiple sensing resistance elements constructed in two or more symmetrical positioned Wheatstone bridges. The bridges constellation is used in order to:

- obtain a symmetrical measurement.
- compensate the temperature dependencies of bridge elements.
- increase the sensitivity.
- obtain the 0V reference.

For angular sensing, the bridges are mounted rotated by a particular angle relative to each other, in order to obtain two different sensing axis. Figure 2.3 shows a simplified circuit diagram for a double Wheatstone bridge sensor.

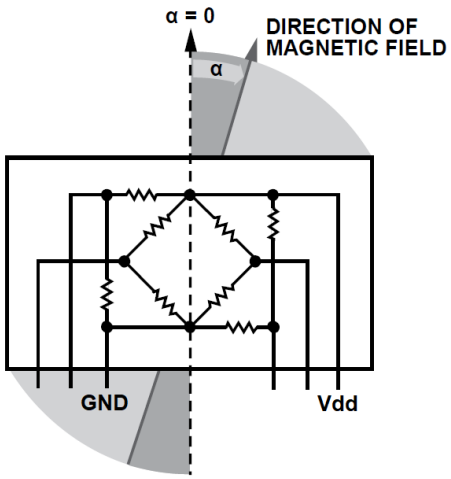


Figure 2.3: Simplified circuit diagram for a double Wheatstone bridge AMR sensor [47, modified]. The mounting position of each bridge is rotated by a particular angle relative to each other for different measurement references.

In newer sensor the amount of Wheatstone bridges is doubled. Each field component H_x and H_y is measured by two bridges with an opposite reference directions. In this way the amplitudes of the output signals can be increased by differential evaluation of measured components, e.g. for angular measurement the +/-sine and -/+ cosine signals differences [73]. Due to doubled amplitudes the signal-to-noise ratio as well as higher dynamical range is obtained what leads to a significant increase of angular accuracy.
Nevertheless, the geometry of bridge elements itself plays a particular role in sensing of magnetic field. A perfect sensing of magnetic field can only be done by use of a single dot sensor,

means the acquisition of field shall be done at an infinitesimal point, what is practically not applicable. Due to physical extent of the bride elements of the sensor the mapping of magnetic field components, e.g. of a rotating encoder magnet, into electrical signals is accompanied by harmonics waves which superimpose the fundamental rotational frequency measured by the sensor. This effect has a negative impact on the system accuracy. Due to this the sensor output signals obtain a nonsinusoidal shape. Some manufacturers have already developed physical techniques for improvement of occurring harmonics, e.g. by curving of resistor stripes shape, which act as spatial filters.[3] Such designs improves the linearity and signal quality, especially for smaller magnetic fields.
The electrical sensor signal measured by bridges are appropriate pre-processed by analog electronics and afterwards either directly sampled by on-chip ADC and processed on its DSP or evaluated on the ECU when the sensor has been mounted in the customer specific application. On the ECU application specific evaluation programs are executed, e.g. such as angle calculation using the CORDIC algorithm or interrupt and timer based speed determination.
Before transition into the field deployment or operation, the sensors are generally calibrated. This can be done in two steps:

1. Calibration phase during the final test. The sensor is trimmed in this phase.
2. Calibration at the end of line (EOL) after mounting of the device into a specific application and / or additional optional calibrations on the ECU

In the first step the calibration procedures are applied directly in the factory where the sensors get specified. Here, the calibration and trimming procedures of deviations such as gains and offset errors of conditioning electronic OP circuits, effect due to physical mismatches of the bridges and its elements displacements as well as setting of initial references are carried out. This is done either directly by programming, trimming of values of the device, or using additional μC hardware. If the sensor system does not exhibit the calibration capability, the procedure is taken over by ECU in step 2, after the mounting of devices on the customer side. The ECU is capable to performs device diagnostics during the operational phase of the sensor system as well as simple real time calibrations, e.g. offset drift compensation [47]. In particular cases the sensor itself performs on board system diagnostics and inform the ECU about sensor system condition [61].
However, the current being used algorithms for the mentioned calibration purposes are based mainly on linear approaches, e.g. in case of angular sensors only for gains and offsets calibrations as well as initial angle correction. The current existing error correction methods are barely able to suppress the effects of higher harmonics and other high order system inaccuracies [68].

[3] FreePitch® and PerfectWave® technology, Sensitec GmbH. The sensor elements are curved-designed. Due to extraordinary shape of resistive stripes a reduction of higher order harmonics and an improve in accuracy is achieved. https://www.sensitec.com/de/produkte-loesungen/mr-sensortechnologie/sensor-designs

2.3 Magnetoresistive sensors

The functionality of magnetoresistive sensors, generalized as xMR, is based on the magnetoresistive effect. A magnetoresistive effect describes the change of the resistance R of a ferromagnetic material in dependence of external magnetic field $\vec{H}$. This technology exhibit usually a high magnetic noise susceptibility (especially the AMR), but is associated with hysteresis[4] behavior. Magnetic hysteresis of a ferromagnetic material draws back on the domain theory from Pierre [95]. It is one technology-related disadvantage of these sensor types. Hysteresis effects cause a nonlinearity in the sensor behavior due to magnetic domain processes, if the sensor operation comes into a distinct hysteresis range.
The magnetic response to the excitation of a body by an external magnetic field is called magnetic susceptibility χ_s and is described by [69]:

$$\chi_s = \frac{M}{H} \tag{2.1}$$

where M and H are generalized. The magnetic flux density $\vec{B}$ for solid-state magnetic bodies depends on the external field $\vec{H}$ and the material magnetization $\vec{M}$. It is comprised of [69]:

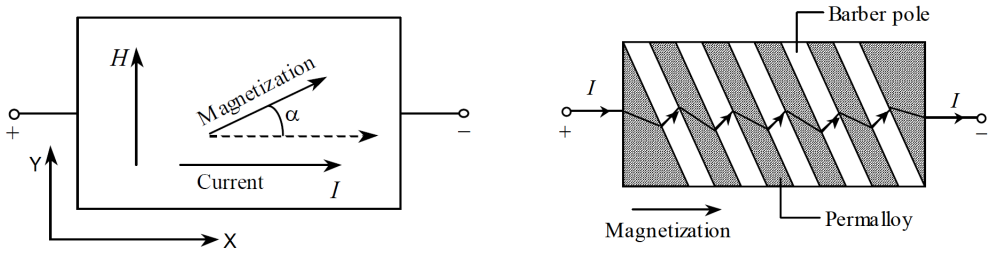
$$\vec{B} = \mu_0(\vec{H} + \vec{M}) = \mu_0(1 + \chi_s)\vec{H} = \mu_0\mu_r\vec{H} = \mu\vec{H} \tag{2.2}$$

The parameter μ_0 is the vacuum permeability, μ_r is the relative permeability and μ the permeability of the magnetic material. It is necessary to mention that the approximation $\mu = const.$ is and only valid for certain considerations as well as in magnetic steady states. This material specific parameter can be nonlinear and can depend on the excitation by the external field $\vec{H}$ its previous magnetization state.
In oder to describe an xMR sensor system behavior simple macro scale models such as Stoner-Wohlfarth are widely applied for description of sensor elements behavior as a single-domain, as well as, much more detailed on the micro scale range, using a micromagnetic model, which is described by Landau-Liftshitz-Hilbert equation.
The xMR effects which describe the dependence of R on $\vec{H}$ are based on physical phenomena which depends on different compositions of magnetic and antiferromagnetic materials. The compositions are divided into three fundamental groups and are described in the following sections.

[4] A hysteresis describes a dependence of the material state on its history. In magnetics it describes the nonlinearity which can be viewed as a lag in the magnetic energy transition during excitation by an external field $\vec{H}$ and internal magnetization M of the material exposed to it. The structure of material is divided into multiple domain states consisting of parallel aligned magnetic moments, which are without the presence of external field statically neutral distributed [69]. In case of an increase of an external magnetic field the domains obtain the resulting magnetization $\vec{M}$ in accordance to direction of $\vec{H}$ only partially. This is due to exchange interaction between parallel aligned magnetic moments in other domain groups and strive for energy minimization. In presence of higher field strengths of $\vec{H}$ the material state reaches its saturation magnetization M_s where all domains are aligned in parallel to the external field direction. By repeating of the excitation the dependence between H and M is distinctive in form of a s-shaped hysteresis loop. For more details refer to [29].

2.3.1 Anosotropic magnetoresistance (AMR)

One of the widely used magnetic field sensors is the AMR sensor which principle is based on the anisotropic magnetoresistance[5]: The electrical resistivity of the stripe is depending on the angle α between the direction of the current $\vec{I}$ and the magnetization $\vec{M}$ of the permalloy. In case of parallel alignment of $\vec{M}$ and $\vec{I}$ the resistance of the stripe is at maximum, for a perpendicular alignment at minimum [91]. Figure 2.4(a) illustrates this principle.



(a) (b)

Figure 2.4: Illustration of AMR effect. (a) Principle of AMR effect in a permalloy stripe. (b) Linearization of resistance behavior of the permalloy stripe by use of barber poles. [44, modified]

For a rotating magnetic field $\vec{H}$ of a fixed field strength the resistance of the permalloy stripe is given by [44]:

$$R = R_0 + \Delta R cos^2(\alpha) \tag{2.3}$$

where $R_0 = R_{min}$ and $\Delta R = (R_{max} - R_{min})$, also called as MR ratio. Specific for the AMR effect is the fact that the direction of magnetization $\vec{M}$ of the permalloy stripe by measurement of its resistance cannot de distinguished from positive and negative. Therefore, the resistance change in dependence of the angle α is doubly periodic.
In consideration of exclusively excitation of the stripe by a field in y-direction H_y its resistance dependence is given by [76]:

$$R = R_0 + \Delta R \left(1 - \left(\frac{H_y}{H_0}\right)^2\right) \tag{2.4}$$

where H_0 is a parameter for description of saturation field of the stripe defined by its shape anisotropy, which depends on permalloy stripe properties and its geometry.

[5]The magnetic anisotropy itself describes the following effect: The direction of „elementary magnets“ (which are responsible for spontaneous magnetization of a material) without the presence of an external magnetic field tries to align to a preferred direction also called easy axes defined by the shape anisotropy [69]. This effect is specific e.g. for a permalloy stripe (NiFe). An external magnetic field $\vec{H}$ in the perpendicular direction to the easy axis leads to a change of the magnetization $\vec{M}$ of the permalloy stripe.

According to the Formula 2.4, the behavior of the AMR resistance is not linear. Figure 2.5(b) (blue curve), illustrates this behavior. In applications for position or speed sensing, a linear AMR behavior is required [21]. A Linearization of an AMR stripe is obtained by modification of the permalloy element with 45° rotated metallic barber poles stripes placed on it. Due to this the current changes its flow direction and follows the path of least resistance. In newer applications the linearity is obtained by a displacement of the permalloy stripe on the sensor die by 45° with regard to the direction of measured external field component [68]. In this way the characteristic curve of the resistance is linearized, shown in Figure 2.5(b) (blue curve).

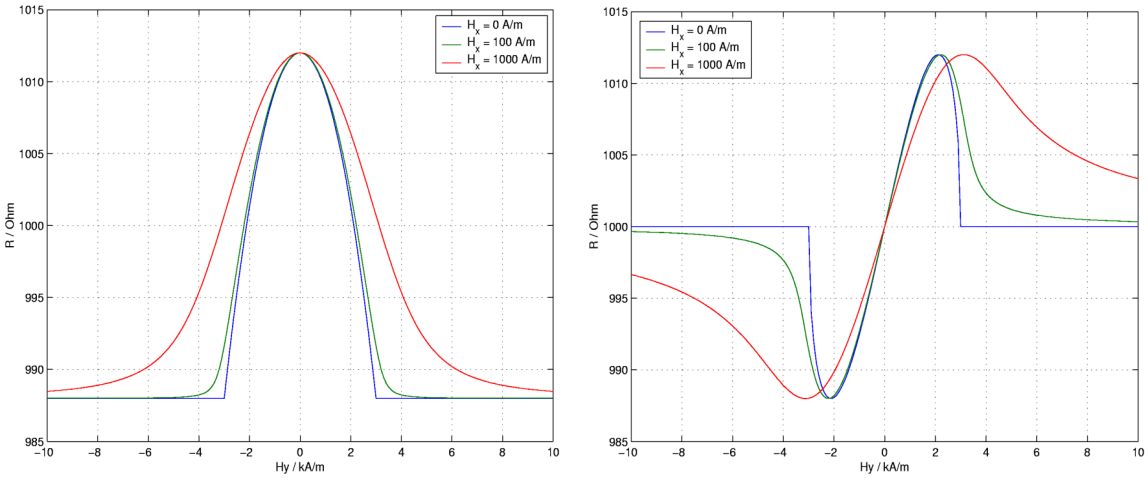


Figure 2.5: A typical resistance curve of an AMR permalloy stripe with and without the presence of cross fields [87]. (a) Resistance change of a not linearized permalloy stripe. (b) Linearized permalloy stripe behavior. *For linearization explanation see also [3]*.

For angle sensing applications where the sensor is operating in the saturation and only the angle information is of interest no linearization of sensors is needed. But for speed sensing application a linear sensor behavior is required. In this application the sensor has to measure the field component along one axis, the sensitive axis [21]. In the presence of cross fields perpendicular to the measurement axis the measurements are perturbed. This leads to a change of the sensitivity and therefore a malfunctioning of the device (the linearity gets lost)[6]. The disturbance of the linearity is clearly visible in the green and red lines in Figure 2.5(b).

[6]The resistance of am xMR element is depending on its complete magnetization which is in this case affected by both field components H_x and H_y.

2.3.2 Giant magnetoresistance (GMR)

In contrast to the anisotropic magnetoresistance a magnetic field sensor based on the giant magnetoresistance consists of a multilayer structure. A GMR stack is composed of at least three layers: a ferromagnetic layer at the top[7], a non magnetic conducting spacer layer[8], also called separator, and an underlying ferromagnetic layer. Due to RKKY[9] coupling of both ferromagnetic layers and antiferromagnetic interlayer an initial alignment of magnetizations is obtained [93]. In order to obtain a reference, the underlying layer is usually pinned by a antiferromagnetic layer, which causes an exchange biasing field at the interface to the underlying layer. The exchange biasing can be understood as a unidirectional anisotropy which causes a shift in the hysteresis curve of the xMR stack or modify and linearize its behavior [74], [93].
The GMR effect results from spin dependent scattering of conduction electrons when a spin polarized current interact the magnetics layers. In case of a parallel magnetization of adjacent magnetic layers in the GMR stack the scattering is at minimum and the current passes the magnetic layers well. By opposite magnetization the magnetic scattering is at maximum and the highest resistance value is present [33], [91]. Spin dependent scattering can be explained by the two current network model for spin-up and spin-down electrons [93].
In Figure 2.6(a) a classical GMR trilayer structure and the corresponding hysteresis curve is shown. Usually, both magnetizations of the layers can be switched in such a structure. In order to linearize the behavior of the GMR stack and obtain a reference layer the stack is suitably biased by a appropriate strength of the antiferromagnetic layer. In this way a structure called spin-valve is created, where only one layer is free to respond [76], [93]. The resistance slope is shifted so that positive and negative values of the measured field component become distinguishable. Figure 2.6(b) illustrates such a structure. Speed sensors are composed as spin-valves because their application requires field linear behavior and are usually exposed to the field which variates only in one axis. In the presence of cross fields perpendicular to the measurement axis the operation of the sensor gets disturbed as well [68].

[7] often denoted as soft layer or free layer, if the ferrommagnetic layer at the bottom is pinned.
[8] general description for interlayer in a GMR sandwich structure. Nonmagnetic means antiferromagnetic in this context.
[9] Ruderman-Kittel-Kasuya-Yosida interaction. For more details see [93].

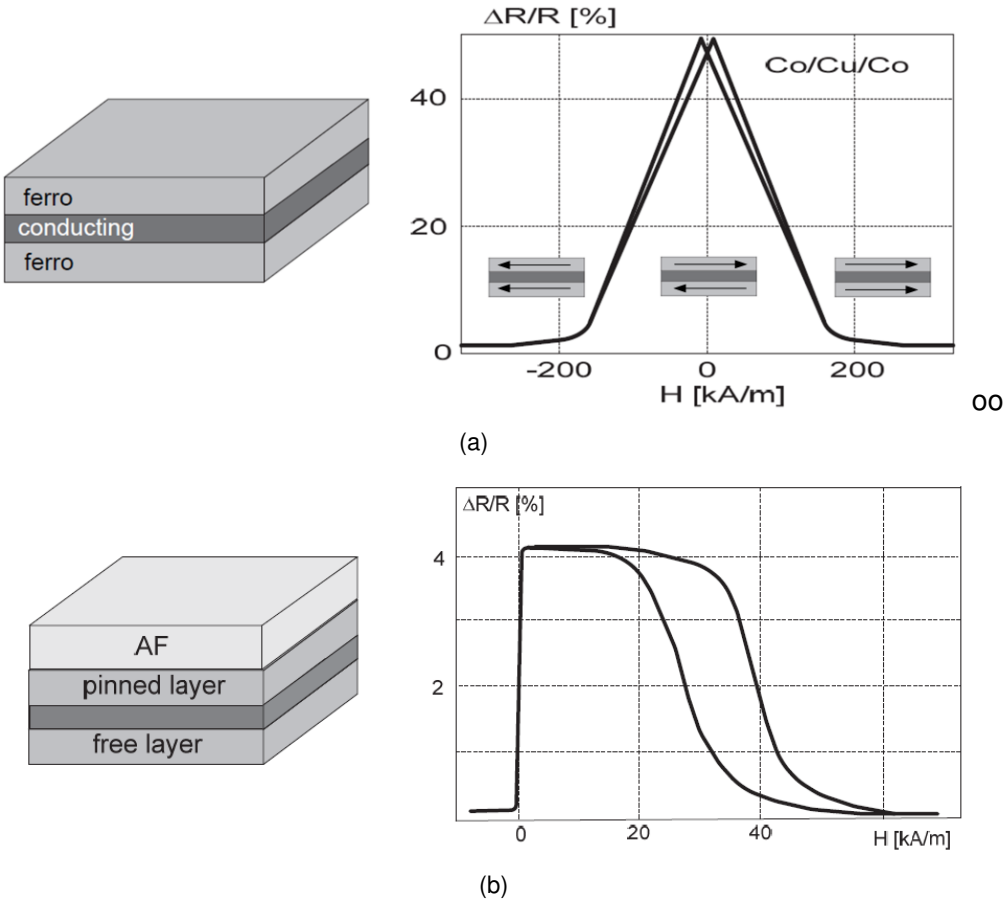


Figure 2.6: GMR trilayer and spin-valve structure and the corresponding hysteresis curves [92, modified]. (a) Left: general GMR trilayer. Right: Hysteresis of a GMR trilayer. (b) Left: Spin-valve structure obtained by coupling of the trilayer with an antiferromagnetic layer. Right: Shifted hysteresis curve and obtained linearization (around the 0 kA/m area). Sensor operation area is at lower field strength.

By excitation of the GMR stack by external rotational field $\vec{H}$ the magnetization of the soft layer follows its direction. As a result, the resistance vary as a function of the angle between the magnetizations of the layers. The resistance of the GMR stack in dependence of the angle is given by [76]:

$$R = R_p + \frac{\Delta R}{2}(1 - cos(\theta)) \tag{2.5}$$

where $R_p = R_{min}$ represents the resistance of the stack for parallel alignment of adjacent ferromagnetic layers, ΔR is the resistance difference $R_{ap} - R_p = R_{max} - R_{min}$, where R_{ap} is the resistance for antiparallel alignment of the ferromagnetic layers and θ is the angle between tho magnetizations $\vec{M}_1$ and $\vec{M}_2$ of the ferromagnetic layers.

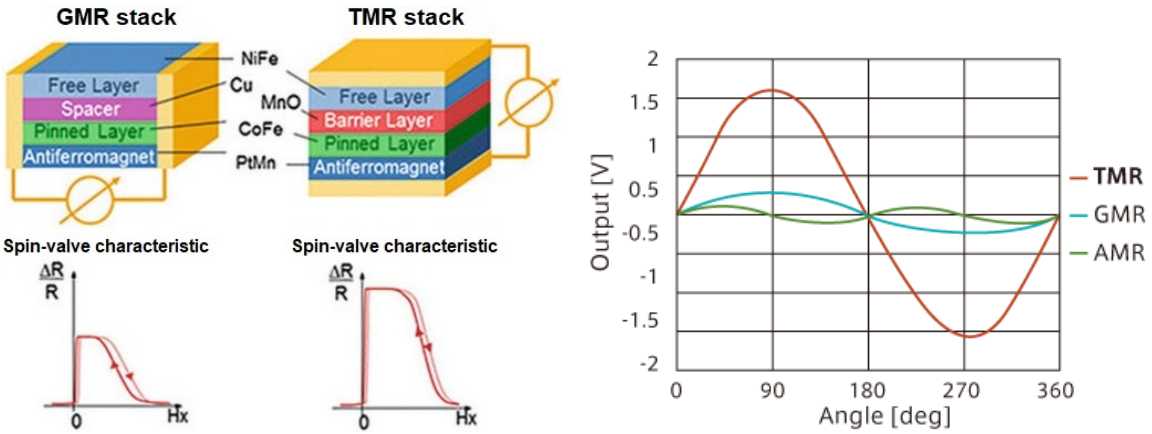
In a magnetic sensor based on GMR effect there are two current flow direction distinguished [93]:

- **CIP:** Current in plane system. The electrical current flows parallel to the plane of the stack, through all layers.
- **CPP:** Current perpendicular to plane. The current flows perpendicular to the plane of the stack.

Due to a higher hysteresis characteristic of a conventional GMR sensor phase jitter effects are more prominent, also denoted as phase noise of the sensor. Due to the domain processes associated with hysteresis, when the measured field component of fixed rotation frequency crosses its zero value, the detection of zero crossing is less accurate. The result out of that is is the detection error of the sensor, what represents the rating for its quality.
However, according to the latest research results from University of Vienna, the hysteresis behavior of spin-valve sensors has been significantly improved by use of topologically protected vortex structures [40]. These types of structures exhibit a negligible small phase jitter effects.

2.3.3 Tunnel magnetoresistance (TMR)

The principle of the tunnel magnetoresistance is the tunneling effect of the conducting electrons through an energy barrier. The TMR structure is similar to that of the GMR, but instead of conducting separator between the two ferromagnetic layers a insulating layer is placed [93]. Instead of elongated of stack structures like in AMR or GMR CIP, multiple smaller TMR dots elements are connected. This composition is called magneto-tunnel junction MTJ. The TMR effect arises due to spin dependent tunneling of the conducting electrons and the origin of this effect can be explained by a quantum mechanical effects only, e.g. by use of Schroedinger equation for description of tunneling probability of the electrons [36]. A TMR system is a CPP system. In a TMR stack the current needs to pass through the insulation barrier.
However, the behavior of GMR and TMR is in analogy and both technologies obey the same laws regarding their electrical behavior. One significant difference of TMR in comparison to GMR is its larger $\Delta R/R$ ratio, a higher sensitivity and low power consumption. But unfortunately, as well as the GMR and AMR effect based sensors, a conventional TMR sensor exhibits likewise cross field dependencies [68].
Figure 2.7(a) shows a comparison of an GMR and TMR stack and illustrates their spin-valve characteristics. In Figure 2.7(b) a voltage output of a TMR sensor element in comparison to GMR and AMR elements in angular position sensing application is shown.



(a) (b)

Figure 2.7: Comparison of GMR and TMR stack structures and the output voltages levels of magnetic sensors using an AMR element, GMR element, and TMR element [16], [13]. (a) GMR and TMR stack structures and electrical characteristic [13]. (b) Voltage output of AMR, GMR and TMR angle sensor elements over full encoder magnet rotation at 5V Vdd [16]. The TMR element voltage swing is multiple times higher than the GMR and significant higher than AMR (double periodicity).

As a conclusion with respect to the described xMR functionality it follows:

- For angular sensing saturated systems are used where no field strength information is extractable. Such system behaves linearly only with respect to the angle of the magnetization and external magnetic field, respectively.
- An xMR element needs to be linearized, if an unambiguous sensing of magnetic field components in one directions is required from the application.
- Conventional linearized xMR sensor elements exhibit cross field dependencies.

For the algorithms developed in Chapter 3 the key feature of linear behavior of the sensing elements is required, where each field component is measurable exclusively. For this purpose linearized xMR elements are necessary, eg. construction of spin-valve structures whose cross field dependencies are negligible small.

2.4 Hall sensors

A Hall sensors is one of the most often used magnetic field sensors. The advantages of hall sensors is a high linearity and unidirectional sensitivity [92], [68]. However, there is an important drawback of hall sensors: their relative small sensitivity and in consequence of it, a high impact of magnetic noise on the output signals.
Figure 2.8 illustrates the working principle of a hall sensor. A Hall effect magnetometer of thickness d placed on the sensor die is exposed to the magnetic flux density $\vec{B}$ [76]. An electric current $\vec{I}$ passes through the Hall element perpendicular to $\vec{B}$, which points into the z-direction.

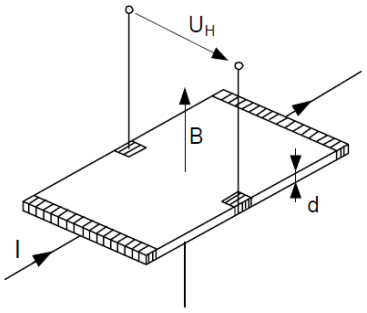


Figure 2.8: Principle of magnetic field sensing by a Hall element [76].

Due to Lorentz force given by [84]:

$$\vec{F} = Q \cdot (\vec{v} \times \vec{B}) \mathrel{\hat{=}} Q \cdot v \cdot B \tag{2.6}$$

where Q is the electric charge, B magnetic flux density and v velocity of electrons, a charge displacement occurs. This results in a Hall voltage generation between the lateral electrodes [76]

$$U_H = \frac{R_H \cdot I \cdot B}{d} \tag{2.7}$$

where R_H is the Hall-constant which is depending on the physical properties of the material.
Hall sensors are equipped with integrated magnetic flux concentrators and signal conditioning circuits [76].The flux concentrators are needed for redirection of the x and y field components into the z-direction in order to make able their acquisition. Additional signal conditioning are required in order amplify the signals and filter the noise impact. This signal conditioning causes also an operational bandwidth limitation.
This dependence between U_H and B in a Hall sensors is proportional and thus, a Hall sensor represents a linear sensor system. Another important point is that by use of the Hall technology an unambiguous measurement of each field components is possible. That means that no cross field dependencies are existent, if the flux concentrators posse negligible small tolerances and their nonlinearity effects are insignificant[10].

[10] A flux concentration used in a Hall sensor can also exhibit a hysteresis behavior [68].

2.5 Technology comparison

The considered technologies for sensing of magnetic field are compared by their main features with respect to automotive application and shown in Table 2.1.

Characteristics	AMR	GMR	TMR	Hall
Δ R/R	∘	+	++	−
Sensitivity	+	+	++	−
typ. values [mV/V/Oe]	1	3	100	~ 0.05
Dynamic range	−	∘	+	++
typ. values [Oe]	*10*	*100*	1000	*~ 100000*
Signal-to-noise ratio	++	∘	∘	−
Bandwidth	++	+	+	−
Power consumption	∘	+	++	−
typ. values for current consump. [mA]	1 ~ 10	1 ~ 10	0.001~ 0.01	5 ~ 20
Temperature stability	+	++	++	−
Hysteresis	++	∘	∘	++
Miniaturization	∘	+	++	−

Table 2.1: Comparison of magnetic sensing technology parameters. Combined and reduced from [24] and [85]. Extended by rating of dynamic range and reassessed the sensitivity.
Rating scale: ++ outstanding, + good, ∘ mean, −satisfying

The advantage of an AMR sensor is its excellent signal-to-noise ratio, relative wide operational bandwidth and marginal hysteresis effects. A GMR sensor has a better behavior regarding its dynamical range as well as higher MR ratio and temperature stability, but is associated with higher impact of hysteresis. A TMR sensor exhibits excellent voltage swing and its MR ratio is significantly higher then of all other candidates due to the highest sensitivity and wider dynamical range. Other main advantages of a TMR sensor are its minor power consumption due to insulation barrier and miniaturization possibility. Magnetic flux concentration and signal conditioning of a Hall sensor needed for its sensitivity improvement increases the size and weight of the sensor [24]. Although its dynamical range is larger, its $\Delta R/R$ is still the lowest due to the physical-related marginal sensitivity. The main advantage of a Hall sensor is the linearity. Furthermore, a hall sensor does not exhibit cross field dependences. The field components H_x and H_y can be measured without any common dependence, under the assumption that the flux concentrator is trimmed satisfactory and its nonlinearities are neglectable.

2.6 Sensor requirements

For sensors and other electronic components in automotive applications the two following industrial standards are essential:

- **ISO 26262: Road vehicles - Functional safety**
  This standard needs to be applied to safety-related systems. It aims to possible hazards caused by malfunctioning behavior of electrical / electronic safety-related systems, which includes also the interaction of these systems. The standard includes a risk classification scheme Automotive Safety Integrity Level, which specifies the integrity requirements on the products [19].
- **ISO 11452: Road vehicles - Component test methods for electrical disturbances from narrowband radiated electromagnetic energy**
  It defines terms, provides the conditions and guidelines as well as establishes the principles of the tests of components in order to determine the immunity of build in electronic components of passenger cars and vehicles against narrowband radiated electromagnetic energy [22].

In addition to the two standards, vehicles manufacturing companies called OEMs[11] defines their own standards and specification for components, e.g. VW, as well as Tier 1 companies[12], like Continental.
The standard ISO 26262 provides a statement of requirements for reliability and safeguarding against failure of the system. In automotive the Automotive Safety Integrity Levels B and D are the most widely used. The derivation from ASIL-B level is that the device must be able to perform self-monitoring and on board diagnostics and inform the main system e.g. ECU about its condition or give a warning in case of detected failure. ASIL-D is the highest safety standard. It demands that if function of a system fails once, it is still guaranteed from the device that it continues to operate, what e.g. can be achieved by a complete system redundancy consisting of 2 chips [68].
The newest ISO 11452 standard demands the ability of a sensor system for suppression of magnetic disturbances from narrowband radiated electromagnetic sources. This represents the absolute necessary requirement for sensors of the next generation.
Furthermore, there is an operation area of a sensor device with respect to mechanical conditions of the encoder magnet and the mounting tolerances of the sensing device. This area is called Safe Operating Area and denoted in the following as SOA.

In the following sections the framework for the algorithm investigations, which are carried out in Chapter 3, with respect to the ISO 11452 and SOA requirements is derived.

[11] Original Equipment Manufacturers.
[12] First Tier Suppplier: companies supplier of the first priority.

2.6.1 Encoder magnet - sensor system and mechanical misalignment

In the automotive application mostly cylindrical magnets with diametrical magnetization are used. According to various kinds of application the diameter of applied magnets vary from 6 mm to 18 mm and the height from 2 mm to 8 mm. The field strength of encoder magnets at the symmetrical central point of the rotation axis in an air gap distance in z-direction to the sensor plane of 2 mm should do not be less than 25 kA/m [68]. The system operation is always associated with unavoidable mechanical misalignment in the form of mounting tolerances resulting during the installation of the encoder device as well as mounting of the sensor or change over lifetime. Furthermore, the distance in z-direction between the sensor and the magnet is application specific and variates as well. The mechanical tolerances and the operational z-distance are recorded in Table 2.2.

Characteristics / Attributes	**Limits**
Air gap encoder magnet / sensor plane:	2 mm to 5 mm
Displacement of encoder magnet / sensor plane symmetry point (x- and y-translation):	$\pm$ 1 mm
Radial shift of encoder magnet:	< 0.5 mm
Tilt to x- / y-axis:	< 5°

Table 2.2: Technical application conditions and tolerances due to assembly and operation [68].

The values contained in Table 2.2 represent limits within the SOA of the sensor. Within this operational range the system shall be operating reliably without exceeding the specified angular error. Furthermore, the fabrication tolerances of the sensor device are also not neglectable. These tolerances occur during placement of the magnetic field sensitive sensor dies on the leadframe. The sensor die tolerances are recorded in Table 2.3.

Characteristics / Attributes	**Limits**
Sensor die displacement in x- / y-direction:	$\approx$ 0.2 mm
Tilt in relation to z-axis:	$\pm$ 2°
Maximum sensor die angular displacement:	$\pm$ 2°

Table 2.3: Mechanical leadframe and sensor die tolerances due to fabrication and packaging [20], [68].

The values contained in Table 2.3 represent the standard errors of the sensor which should be compensated by a calibrations phase already at the final test.
The size of the sensor die is 2 x 2 mm^2 and is significantly smaller than the sensor package.

In view of described mechanical misalignment the goal for the angular accuracy is to achieve a maximum angular error $\varphi_{err} \leq 1^{\circ}$. This setting represents the obligatory angular accuracy requirement of the sensor system [68]. In order to fulfill the set requirement it is allowed to perform a teach-in procedure after the mounting of the device at the customer side.

2.6.2 Stray field suppression

In accordance to the contents of the new standard ISO 11452: Part 8 [22] the electronic devices of passenger cars and vehicles have to be immune to the disturbing magnetic fields generated by narrowband radiated electromagnetic energy sources.Generally, the term stray field is used for the denotation of the disturbance fields and is therefore used in this thesis as well.
According to the given by ISO 11452 in combination with OEM specific norms the properties of the stray field are specified as recorded in Table 2.4.

Characteristics / Attributes	**Specification**
Field configuration:	from various direction but spatially homogeneous
Field strength:	up to 4 kA/m
Required angular accuracy:	$\leq 1^\circ$ in presence of stray field

Table 2.4: Stray field specification and requirements for system accuracy [22]. According to ISO 11452 the strength of the stray field is specified to max. 3 kA/m. However, this limitation was increased to 4 kA/m in order to cover also the OEM specific requirements.

In correspondence to Table 2.4 the stray field is specified as spatial homogeneous with a maximum magnitude of 4 kA/m. The angular accuracy of the sensor in presence of the stray field shall not be less than 1°. The angular accuracy requirement of $\varphi_{err} \leq 1^\circ$ is the set goal within the context of this thesis.

A stray field suppression can in general be achieved by following concepts:

1. Magnetic shielding of nonrelevant sensor environment, for example by use of permalloy rings. In this way a partially stray field suppression is obtained [28].
2. Application of multipole magnetic encoder systems and measurement of fields in opposite directions. The homogeneous disturbance components will shift both measured components to the same direction, but the common mean value remains not affected [27].
3. Measuring of field at multiple points in the spatial domain. For this concept an application of encoder magnets with inhomogeneous field is necessary. The spatial degree of freedom allows a mathematical separation of the homogeneous part of measured field from the inhomogeneous one, where the useful information can be extracted from.

The thesis targets exclusively on the latter stray field suppression concept. The principle explanation as well as advantages and disadvantages of such a system are given in the following section.

2.7 Comparison of a single sensor and sensor array system

In comparison to a single sensor system which measures the strength of the field components only at one point, the aspired sensor array system consisting of multiple “sensor cells“ offers by its spatial resolution a measurement of the field components over a certain area. This offers a unique advantage for suppression of stray field, which naturally cannot be obtained by a one sensor system. The stray field suppression is obtained by evaluation of the gradients contained in the measured field and completely nonconsideration of homogeneous field component. The principle of field measurement in the spatial domain and the capability to suppress the stray field interferences of a sensor array system is illustrated and explained in Figure 2.9.

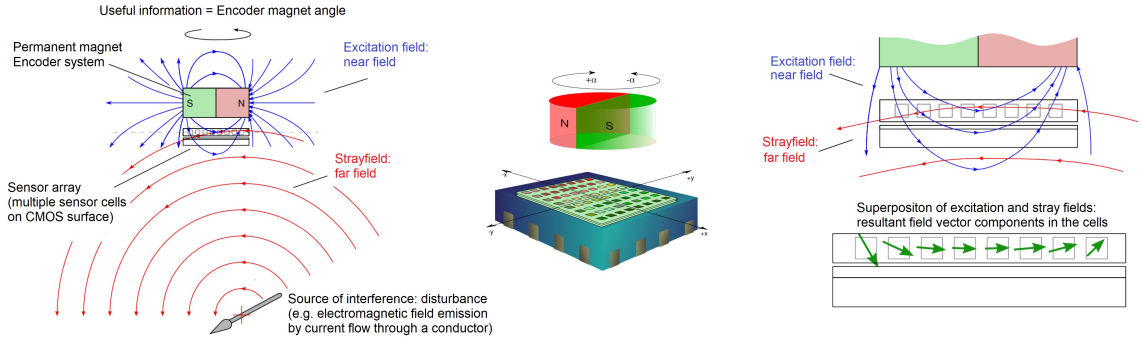


Figure 2.9: Schematic Illustration of the stray field suppression using a sensor constructed of a array with multiple measuring points distributed in the spatial domain. Left: Encoder magnet - sensor array system and stray field interference. Center: Schematic measurement configuration of magnetic field by a sensor array within the xy-plane. Each sensor pixel measures the field strength in x- and y-directions. Right: Near representation of field interference and measured field components. Blue field lines represent an inhomogeneous near field generated by encoder magnet. Magnetic disturbances in the form of a stray field are represented as red lines. The useful information can be viewed as high frequent spatial components from the encoder magnet field whereas the interfered stray field is less inhomogeneous due to larges distances to its origin and therefore exhibit only low spatial frequency components or completely homogeneous behavior. The stray field suppression is obtained by filtering of the homogeneous part from the measured field components [25].

In addition to the stray field suppression feature of a sensor array other advantages and potential for algorithmic processing are obtained by use of an sensor array system. In case of a failure of sensor elements online interpolation and approximation techniques can be applied, what leads to more functional robustness of the system. A part-redundancy[13] is automatically obtained if a sensor array chip is used in the customer application. Therefore, the fail-safety is increased. Furthermore, an appropriate preprocessing e.g. spatial and frequency domain

[13] A completely redundant system in accordance to the ASIL functional safety requirement can only by obtained by a multiple sensing system with full galvanic separation of the sensor, independent power supply and separated signal output lines: e.g. a dual die sensor solution [38].

filtering and methods for outliers removal can implemented which brings more safety as well as robustness against high frequent noise and enhancement of signal quality directly on spatial domain. This will automatically reduce the effort for signal conditioning needed for the postprocessing in time domain during operation.
However, to guarantee the fulfillment of the superior characteristic of a sensor array the related signal processing for angle calculation involves at the same time systematic disadvantages. In comparison to a standard single sensor a sensor array concept implies the following disadvantages[14]:

- Requirement of a relative high ADC resolution
- Increased impact of noise effects
- High sensitivity against misalignment effects
- Processing delay associated with analog data acquisition and computational operations

First of all, a high resolution will be required to be able to detect the field gradients. This is the case for a single sensor system operation on direct field components and excited by a homogeneous field. By using the sensors as magnetometers high impact of quantization noise associated with sampling of data and fixed dynamical range of the ADCs will occur. The ADC range with respect to spatial differences of components where the angle information is extracted form will be small in relation to the total dynamical coverage. Secondly, if the direct component is not evaluated, all the physical noise entering into the system will have stronger impact on the signal quality due to its more significant changes of direct components and gradients differences in this case. The third point is that due to a inhomogeneous encoder magnet field the system will exhibit an high sensitivity against mechanical misalignment and vibrations of the encoder magnet. This is not the case if the system is excited by a strong homogeneous field and the sensor operates in the saturation [20]. The last point is the fact that the algorithms needed for stray field suppression will entail a processing delay, which will reduce the angle resolution with respect to the rotational frequency.
A good compromise to avoid this weaknesses would be a hybrid system, which evaluates the spatial content from the sensor array only in case of the presence of the stray field. In oder to build such a system a detection of the stray field is necessary.

Magnetic sensor arrays has already been constructed and directly integrated on chips even in combination of signal conditioning and logic optimization integrated circuits for low field applications [34]. Further micron-sized magnetic sensors for scanning and imaging of magnetic media [86] as well as for biomedical application in nanomedicine imaging, therapeutics purposes and early disease detection [41] have also been designed and applied, mainly as GMR and SV structures.
The recently new requirement for stray field suppression opens the door for magnetic sensor arrays into the automotive branch as a suitable problem solution.

[14] The disadvantages are related to the case if the homogeneous part is still acquired by ADC units but not evaluated for angle calculation.

3 Development of algorithms

This chapter represents the first main part of this thesis.
In the first part of this chapter, the developed processing approach, evaluational conditions and the functional split into a teach-in and an operational phase is introduced. Subsequently, the development of algorithms for angle information extraction from data measured by the sensor array is documented. The approaches include stray field suppression functionality according to the ISO 11452 [22]. Their assets and drawbacks as well as the required computational and implementation effort are discussed and compared. Afterwards, chosen suitable filter candidates for possible preprocessing of the sensor array data are briefly explained. After it, the developed methods for handling of encoder magnet misalignments are documented, which allow to perform on board diagnostics according to ASIL-B requirements [19]. Thereafter, angular error compensation algorithms based on a teach-in phase on the spatial domain are explained and their advantages and disadvantages discussed. By application of this algorithms it is possible to compensate the misalignment effect of the encoder magnet already on spatial domain.At next, the developed linear and nonlinear angular error correction algorithms are presented. The developed approaches here represent spatial domain independent methods. In the second to last step, a possible stray field detection method is introduced. To sum up at the end of this chapter, the mathematical as well as combinatorial algorithmic chain for the application is illustrated and described as a chapter summary.

A great emphasis was placed on practical feasibility of the methods. Thus, a bottom-up methodology was followed, starting from the lowest degrees of mathematical basis functions, simplest approaches and conditions for the teach-in as well as operation phases. The mathematical formulations required for the estimation processes as well as signal processing flows are explained in detail. The development of the algorithms was carried out within the context of pre-analysis by using the simulation described in Chapter 3. Prior work of the ISAR project members was considered in the algorithm development [75], [52]. All algorithms are implemented in MATLAB®.

3.1 Required system properties

To be able to rely on the concept for stray field suppression, the following three system properties needs to be present:

- Linear behavior of sensor array elements: a linear transmission while mapping of the magnetic field into the resistance values of the sensor elements is required.
- Both field components in x- and y-direction are extractable from each sensor element within the array so that the field vector $\vec{H} = [x \quad y]^T$ and its strength or magnitude $M = |H| = \sqrt{x^2 + y^2}$ can be obtained.
- At each sensor element the field components are measured exclusively. There are no cross field dependencies.

Therefore, advanced GMR or TMR technology is needed, constructed as spin-valve with negligible cross field dependencies. Alternatively, a Hall sensor array can be applied for this concept.

3.2 Signal processing approach

The developed signal processing approach utilizes the following functional and evaluational conditions:

- Field components are measured directly, i.e. the sensor array elements are constructed based on magnetometer principle.
- Stray field suppression is achieved by suppression of the homogeneous part[1] of the field, i.e. the direct component which is measured within the xy-plane, or in other words, its nonconsideration in the angle information extraction. The direct part is not evaluated for the angle information extraction.
- The angle calculation is based on evaluation of the field gradients measured by the sensor xy-plane.

[1] In accordance to the specification of stray field in ISO 11452: Part 8 [22].

3.2.1 Functional split

During the development of the algorithm a function split of the total functionality was done. The system is consisting of two functional columns shown in Figure 3.1.

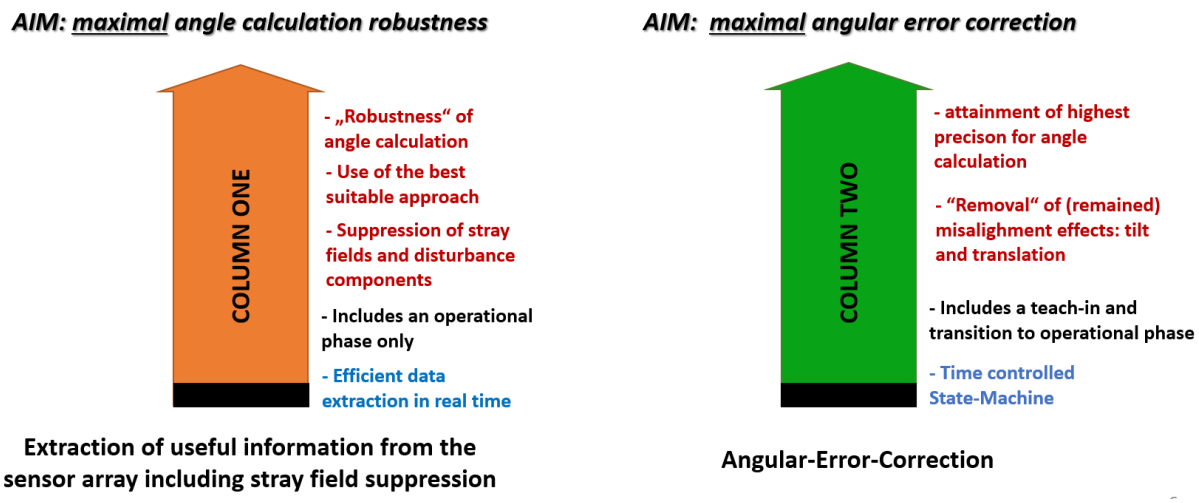


Figure 3.1: Functional split of the total processing approach.

The first functional column includes the algorithmic methods for angle information extraction including stray field suppression. This column includes an operational phase only, which processing time has to be sufficient to guarantee the requirements according dynamic data acquisition and angle calculation in real time. For the coverage of this functionality, the most appropriate algorithm candidate for angle calculation including stray field suppression with a satisfied robustness against noise interfering and an acceptable sensitivity to misalignment effects of the encoder magnet has to be proposed. The processing time required by application of the algorithm has to be taken into the consideration as well.
The second column represents the algorithms which are associated with a teach-in phase for system accuracy enhancement. The teach-in phase is not time critical and and therefore the needed computational effort as well as the required processing time for the teach-in procedure play minor role. For this purpose the calculation can be done on the ECU or an external μC. After the teach-in phase, the system shall enter into the operational phase.
Two teach-in methods were developed:

- **Spatial domain teach-in**
  This teach-in phase includes the localization of the projected rotation axis of the encoder magnet on the xy-plane of the sensor. By detection of the projection point its spatial information can be used for compensation of possible encoder magnet misalignment effects on the angular accuracy of the system. After this teach-in phase the system enters into the operational phase where the information about the angle is extracted

symmetrically around the rotation axis projection point. So, in this approach a use of the field symmetry is made.

- **Angular teach-in**
  This teach-in phase represents a spatial domain independent correction procedure. Here, the signals which are obtained after the angle informations extraction are corrected. In this phase parameter of **correction functions** are determined whose goal is to fit the system behavior as close as possible to that of an ideal, perfectly aligned encoder magnet - sensor system. After this teach-in phase the system enters into the operational phase where the correction functions are constantly used.

3.2.2 Signal processing concept

In Figure 3.2 the signal processing concept represented as a processing flow and coupling structure of modules is illustrated.

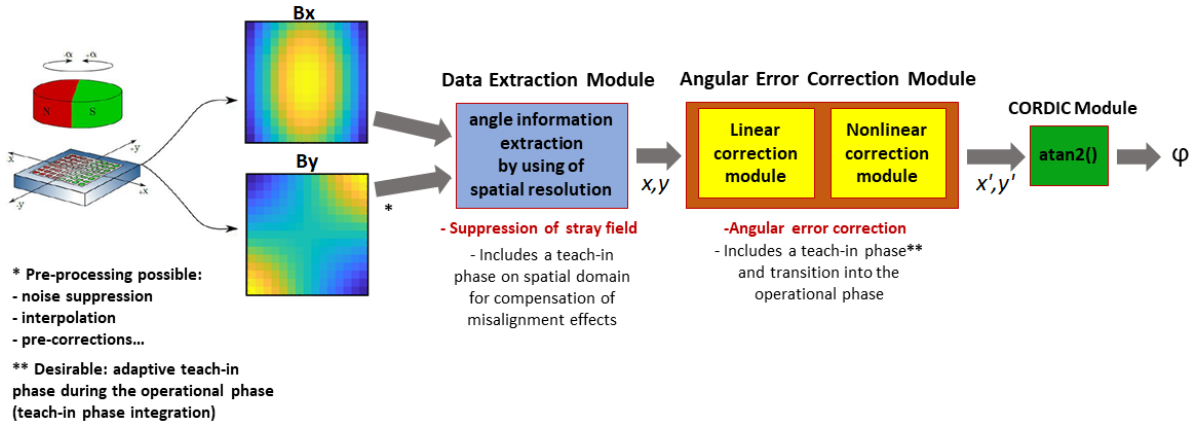


Figure 3.2: Developed processing flow. The signals are sampled and processed first in the *Data Extraction Module*, where the angle information is extracted from the sensor array and the x-/y- signals are propagated further into the *Angular Error Correction module*. After the angle correction procedure the system goes into the operational phase, where the angle φ is calculated in the CORDIC module by use of corrected signals x' and y'. *Magnet - sensor array illustration: T. Schüthe.*

The proposed system consists of 3 modules. In the first module *Data Extraction*, the algorithm for angle information extraction including stray field suppression is applied. Here, the corresponding x and y component which represent sine and cosine component of a rotational external field $\vec{H}_{ext}$ are extracted from the sensor array data[2] $\mathbf{B_x}$ and $\mathbf{B_y}$. The algorithms for the angular error compensation associated with a teach-in in spatial domain can be applied on in

[2]Since this work is related to magnetoresistive sensors, the sampled components of the magnetic field acquired in the xy-plane where sensor array elements are placed, are denoted in the following as components of magnetic flux density, in the notation of matrices / data arrays $\mathbf{B_x}$ and $\mathbf{B_y}$. If the elements stored in $\mathbf{B_x}$ and $\mathbf{B_y}$ are considered, the notation $\mathbf{B_x}$ and $\mathbf{B_y}$ field components is used.

this step. In the second module *Angular Error Correction*, the signals x and y can be corrected by application of angular error correction algorithms. A linear ad a nonlinear approach were developed for this purpose. After the correction, the system goes into the operation where the correction function are continuously used.

A schematic representation of physical signal flow from the sensor bridge elements to the signals read out and propagation to the ASIC is depicted in Figure 3.3.

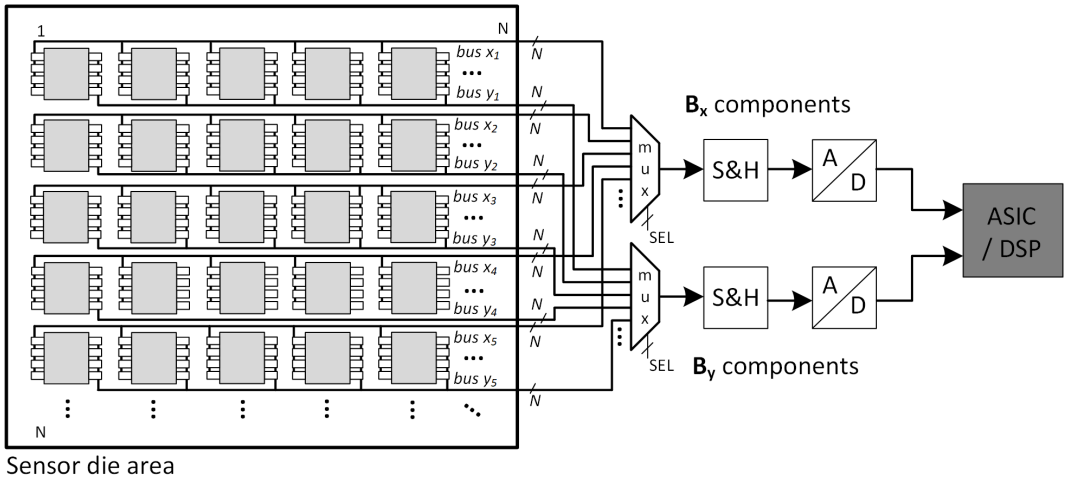


Figure 3.3: Schematic representation of physical signal flows from the sensor elements of a 5 x 5 array to the processing step in the ASIC / DSP. Corresponding signals of the sensor array elements are row wise connected to a multiple buses. Signals are sequentially sampled by ADCs by use of MUX units. Measured components of the external field $\vec{H}_{ext}$ in x direction are stored in $\mathbf{B_x}$ data array, components measured in y-direction in $\mathbf{B_y}$ data array.

3.2.3 Calibration stages

From the end of the manufacturing process of the sensor system to the build in into a specific final application the sensor array system shall pass through 2 calibration stages where the developed algorithms for angular error correction described in Section 3.7 shall be applied. The complete correction chain is illustrated in Figure 3.4.

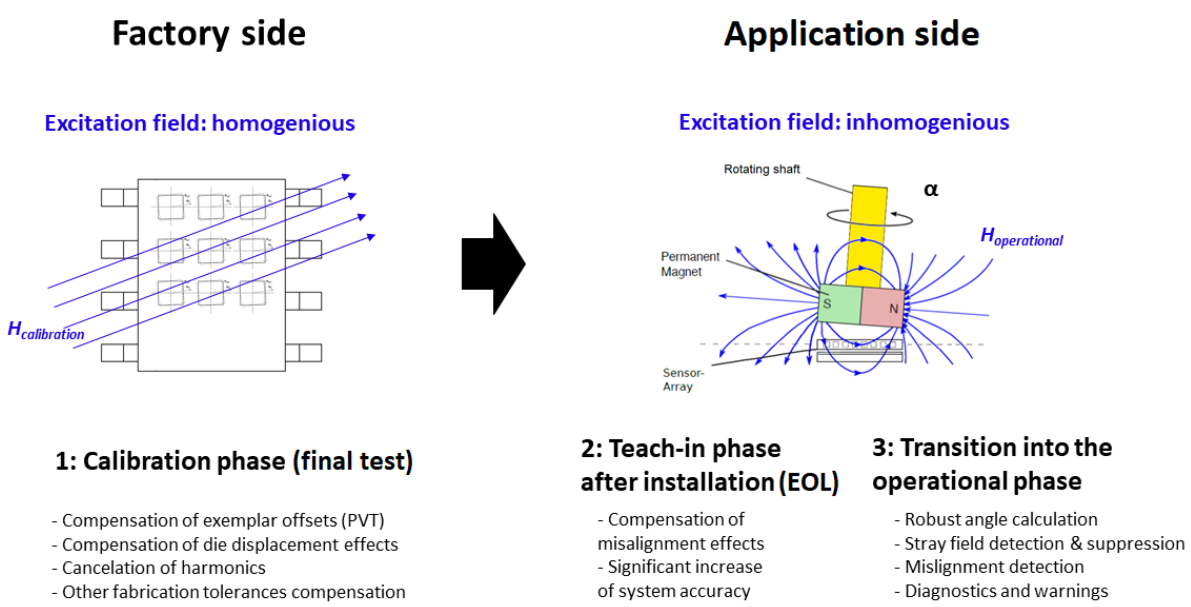


Figure 3.4: Calibration stages of the sensor array system. Left, 1: First stage. Each sensor array element shall be calibrated exclusively. Right, 2: Second stage. Here, the misalignment effect caused by mounting tolerances and encoder magnet displacement is compensated. Right, 3: Operational phase. System conditioning diagnostics can be performed here in real time. *Sensor sketch is taken from [20], magnet illustration from [25].*

The stages are as follows:

1. Calibration of sensor array elements at the final test for correction of gain and offset deviations, sensor die displacement effects and compensation of possible harmonics impacts. Here, the excitation field has to be homogeneous in order to expose all sensor elements to the same field. Due to the subsequent evaluation of field gradients a precise calibration of each sensor array element is necessary. Since the information about the angle is represented in the relative small field range in comparison to the direct component, the individual offsets and gains as well as other fabrication tolerances will presumably have a not negligible impact on angular accuracy. Therefore, their impact needs to be compensated as much as possible during the trimming procedure at the final test.
2. Subsequent teach-in phase after the mounting of the sensor in the specific application in oder to compensate the misalignment effect or correct it completely at the EOL.
3. Transition into the operational phase, where the determined functions for angular error corrections are constantly used in the main ECU, sensor DSP or ASIC. Here, the developed algorithm for stray field and misalignment detection described in Section 3.5 and 3.8 can be used to perform on board diagnostics and inform the main system in the ECU about the sensor condition. The latter will contribute to the fulfillment of the ASIL-B specification.

3.3 Algorithms for angle information extraction

The idea for the angle calculation is the extraction of information about the shape of the measured field $\vec{H}_{ext}$, i.e. the shape which is resulting from the contemplations of field components $\mathbf{B_x}$ and $\mathbf{B_y}$, separately. All methods for angle calculation come already with a low pass filtering effect and include the stray field suppression by nonconsideration of the homogeneous part automatically. The homogeneous part of the field is not evaluated during the angle information extraction. The algorithm candidates are divided in three groups:

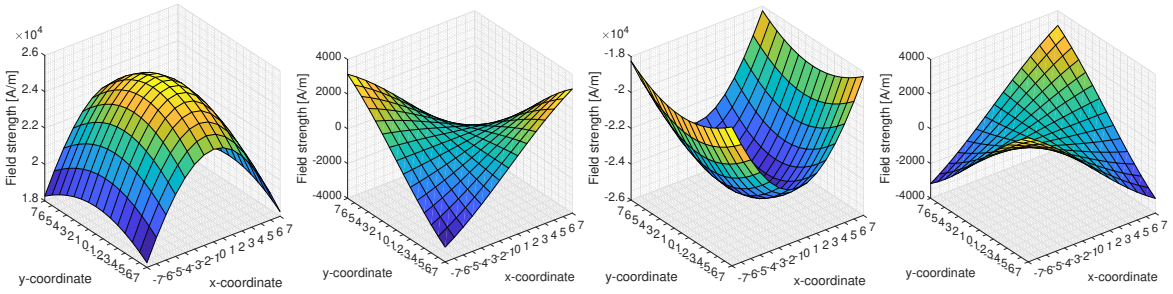
- Two dimensional Fourier transform based algorithm (2D-DFT).
- Approximation algorithms.
- Algorithms based on application of spatial filters from image processing.

In the approximation approach, the low pass filtering is obtained indirectly by estimation and obtaining of optimal solution for the chosen basis functions. The image processing based algorithms includes the Laplacian low pass filter kernel and in case of the 2D-DFT approach only the lowest frequencies are evaluated.

3.3.1 Basic principle for angle information extraction

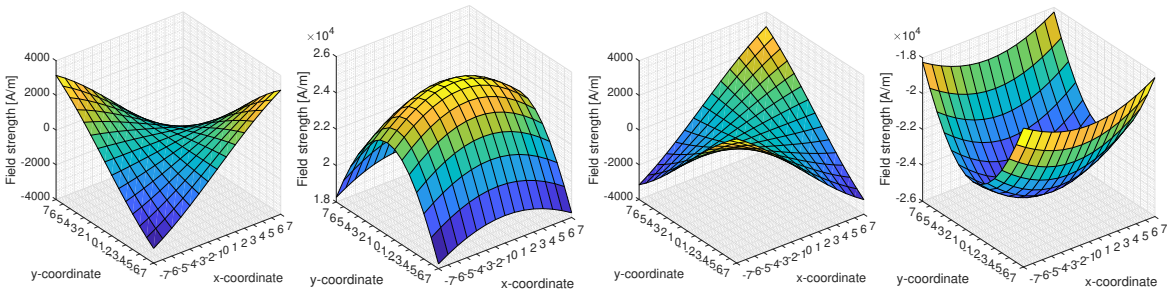
Algorithms for angle calculation including stray field suppression have already been developed within the framework of the ISAR project. The derived formalism for the angle information extraction in this thesis ca be understood as continuation or extension of the prior project work done by K. Jünemann, H. Pape [75], [52]. The algorithms for angle calculation in the prior work were derived from dipole equation based on its magnetic moments.
The principle for the angle calculation used in this work is the **extraction of the curvature** of the shape resulting from the field components $\mathbf{B_x}$ and $\mathbf{B_y}$ measured in the xy-plane of the sensor array. This is mathematically equal to partial derivative calculation of the second order in x- and y-direction from **both** $\mathbf{B_x}$ and $\mathbf{B_y}$ field components. Figures 3.5 and 3.6 illustrate the behavior of simulated field components $\mathbf{B_x}$ and $\mathbf{B_y}$ in a 3D view over 4 angular positions of a perfectly aligned encoder magnet - sensor array system. The magnetic field with respect to its spatial coordinates origin in the center is at equilibrium state: the filed components are symmetrical. At an encoder magnet angular position of 0° the measured $\mathbf{B_x}$ components exhibit a behavior similar to a parabolic function. The global curvature is negative. The $\mathbf{B_y}$ components exhibit only a xy-dependency, the shape formed by the measured components is twisted. Its global curvature is equal zero. The global gradients in x- and in y-direction in both data arrays over the total plane of the sensor array are equal zero as well due to the present symmetrical equilibrium state. The progression of curvature switches from negative to positive over the full rotation of

encoder magnet what stands in full accordance to the sinusoidal functions[3], curvature of $\mathbf{B_x}$ for cosine and $\mathbf{B_y}$ for sine function.



φ 0° φ 90° φ 180° φ 270°

Figure 3.5: Progression of $\mathbf{B_x}$ components measured by the sensor array in the xy-plane at 4 angular positions of the encoder magnet.



$\varphi = 0°$ φ 90° φ 180° φ 270°

Figure 3.6: Progression of $\mathbf{B_y}$ components measured by the sensor array in the xy-plane at 4 angular positions of the encoder magnet.

As a result of the examination of the field behavior it follows that the angle information can be extracted from the array data by e.g. approximation of 2D polynomial function of minimum 2nd degree and evaluation of the coefficients which describe the curvature of the polynomial function, direct numerical extraction of second derivative by following of image processing methods or analysis of spatial frequencies content in the $\mathbf{B_x}$ and $\mathbf{B_y}$ data arrays using the 2D-DFT. Especially the 2D-DFT approach represents a universal method for the present problem.

In case of a mechanical misalignment of the sensor or applied encoder magnet system the field symmetry with respect to the coordinate origin gets lost. Therefore, an angular error arises

[3] Attention: In the simulation setup the angular rotation is anti-clockwise and starts from the first quadrant. The averaged field component measured by the sensor plane points into the positive x-direction. Therefore the extracted curvature from $\mathbf{B_x}$ and $\mathbf{B_y}$ field components over the full 360° encoder magnet rotation represents the negative sine and cosine functions. The negative signs are not shown in the Equations for angle calculation. The values in subsequent depicted graphs with sine and cosine curves progressions are multiplied by -1. The setup of simulation is explained in detail in Section 4.1.2.

which can be compensated by additional teach-in phases that need to be performed before the sensor is set into the operation mode.

The x- and y- component needed for the angle calculation by arctangent function[4] is obtained from the magnetic field measured in the xy-plane of the sensor array by:

$$\varphi = arctan\left(\frac{y_{component}}{x_{component}}\right) = arctan\left(\frac{\frac{\partial^2 B_y(x,y)}{\partial x^2} + \frac{\partial^2 B_y(x,y)}{\partial y^2}}{\frac{\partial^2 B_x(x,y)}{\partial x^2} + \frac{\partial^2 B_x(x,y)}{\partial y^2}}\right) \tag{3.1}$$

3.3.2 2D-DFT based algorithm

For the first algorithm candidate for angle information extraction the two dimensional Fourier transform was taken into the consideration. The 2D-DFT has been proposed as possible algorithmic candidate within the ISAR project group by K-R. Riemschneider and T. Schüthe.
The 2D-DFT plays an important role in image processing and is widely applied for filtering of images, especially due to saving of computational time by use of convolution theorem[5]. Its advantage consists also in offering of a possibility for application of special filtering methods which cannot be carried out in the spatial domain, e.g. such as notch reject filter for cancelation of periodic noise of particular frequencies or disturbance signals [59], [78].

The discrete two-dimensional Fourier transform is defined as [59]:

$$F(u,v) = \frac{1}{NM}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)} \tag{3.2}$$

where $f(x,y)$ is a two-dimensional input signal sequence, i.e. the input image and $u = [0, M-1]$ and $v = [0, N-1]$ are the frequency coordinates of the spatial frequency domain. The other way round the inverse of the discrete two-dimensional Fourier transform is obtained by [59]

$$f(x,y) = \frac{1}{NM}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{+j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)} \tag{3.3}$$

[4] Normally the atan2(,) function is used here which covers the full angular area $[-\pi, \pi]$, where the first argument is the y component and the second the x component. In this thesis for the notation of angle calculation the standard arctangent formula $arctan(y/x)$ is used, due to better mathematical representation.

[5] I.e the transformation of both the input image and the filter kernel and using the chain *2D-DFT* $\rightarrow$ *filt* $\rightarrow$ *2D-IDFT*. This operation is used in case if the computational effort needed for the filtering chain in total is less than the filtering operation convolution of the image with the filter mask. This advantage is enormous especially for non separable filter masks, see comparisons in [78, Chap. 4].

where $x = [0, M-1]$ and $y = [0, N-1]$ are the spatial coordinates of the spatial domain. The term $e^{-j2\pi(ux/M+vy/N)}$ can be represented as:

$$e^{-j2\pi(ux/M+vy/N)} = cos(ux/M + vy/N) - jsin(ux/M + vy/N) \tag{3.4}$$

what is the Euler's formula where instead of y the term $(ux/M + vy/N)$ is used. By use of this term the multiplications coefficients of the 2D-DFT are computed, depending on the total amount of 2D-DFT elements and spatial coordinates. The values calculated by Formula 3.4 are called twiddle factors[6]. This factors are stored in the twiddle factor matrix $\mathbf{W}$, also called DFT matrix **D** [83]. The 2D-DFT is obtained by separable forms, one dimensional DFT transform by DFT matrix multiplication over the rows and subsequently one dimensional DFT over the columns of the picture, using separability property [59]. The information content from the 2D-DFT can be represented as its amplitude spectrum

$$|F(u, v)| = \sqrt{Re\{F(u, v)\}^2 + Im\{F(u, v)\}^2} \tag{3.5}$$

and phase spectrum

$$\phi(u, v) = arctan\left(\frac{Im\{F(u, v)\}}{Re\{F(u, v)\}}\right) \tag{3.6}$$

In most cases, the phase of the Fourier transformation contains more significant information about the structure of the image [59].

Within the framework of development of this algorithm a pre-analysis was carried out where the content of the spatial frequencies in $\mathcal{F}\{\mathbf{B_x}\}$ and $\mathcal{F}\{\mathbf{B_y}\}$ was analyzed and particular frequency pairs extracted of it for angle calculation. A special emphasis during the development of the 2D-DFT based algorithm was placed on the spectrum symmetry. Therefore, a separate application of 2D-DFT on both $\mathbf{B_x}$ and $\mathbf{B_y}$ field components was chosen for this approach, in order to obtain a symmetrical spectrum due to the fact that the spectrum symmetry $|F(u, v)| = |F(-u, -v)|$ of the Fourier transform is valid only for real input signals [83], [59]. This allows a definitely symmetrical spectral analysis and recognition of correspondence of harmonics frequency pairs in order to derive an angle calculation method. Out of this a use was made in this work. Alternatively, the **By** components can be treated as complex signals and the Fourier transform of the measured field components obtained by $\mathcal{F}\{\mathbf{B_x} + j\mathbf{B_y}\}$. But here one needs to be careful, if an extraction of particular frequencies and its analysis is desired. In this case it may happen that the progression of the chosen spatial frequencies won't be in full accordance to the arctangent angle calculation principle. An analysis of alternative 2D-DFT based algorithmic candidate by application of 2D-DFT on a complex input signal was carried out within the framework of this thesis as well and is documented and discussed in Appendix C.3. In this analysis a phase shift was observed in corresponding spatial frequencies groups.

Based on the pre-analysis and application of 2D-DFT separately on $\mathbf{B_x}$ and $\mathbf{B_y}$ components

[6] Commonly used notation for a twiddle factor is $W_N^{nk} = e^{-j2\pi(nk/N)}$

an algorithm was derived which uses directly the spatial frequencies contents of the image for the angle calculation. The analysis results based on simulation data are depicted in Figure 3.7. From figure it is event that the magnetic field exhibit a low frequent behavior. The main content of the information about the magnetic field is contained in the first spatial frequencies. Therefore, only the lowest spatial frequencies were considered in the second analysis step. Due to the spectrum symmetry only the first 3 spatial frequencies in the lower right corner from the direct component were analyzed. The analysis is depicted in Figure 3.8. It shows the progression of spatial frequency components at coordinates (u = 1, v = 0), (u = 0, v = 0) and (u = 1, v = 1) of all 4 obtained 2D-DFT images over the full encoder magnet rotation.

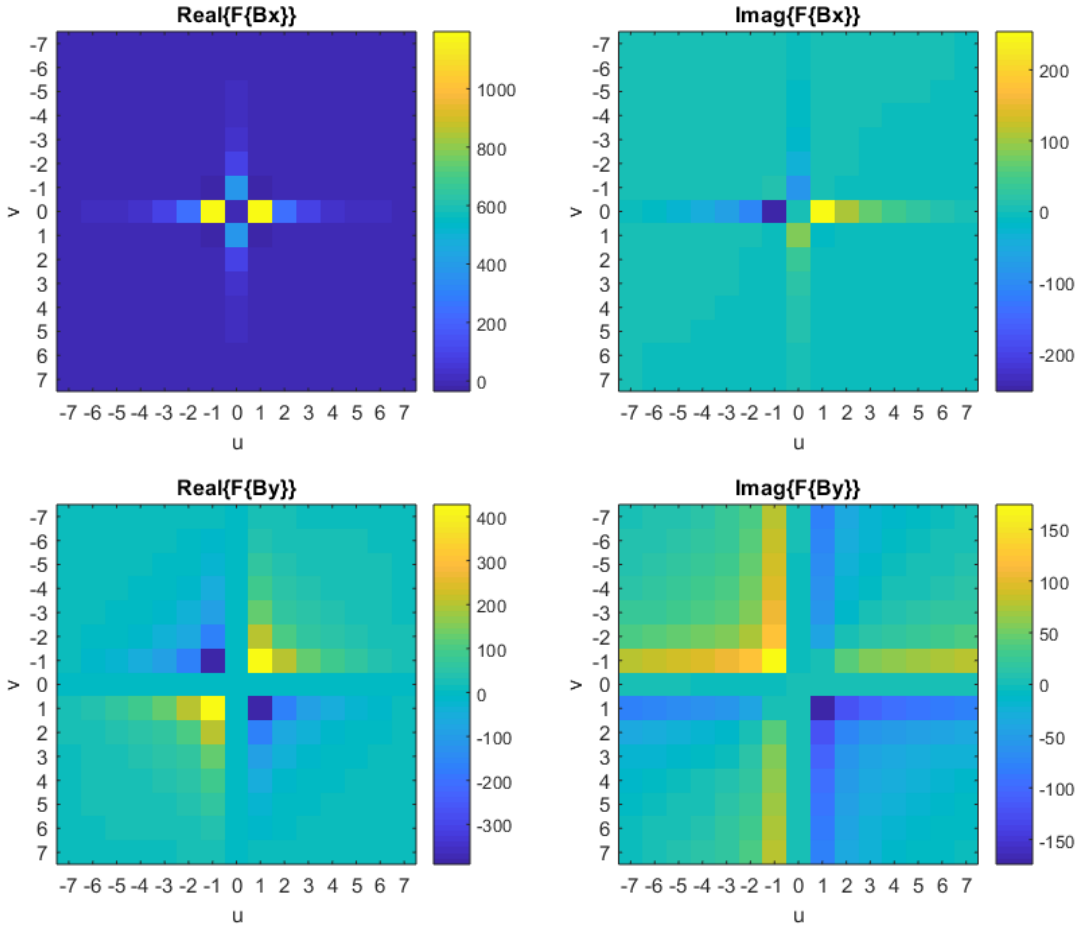


Figure 3.7: Spatial frequency components of 2D-DFT applied separated on $\mathbf{B_x}$ and $\mathbf{B_y}$ data. Frequency components are shifted to the center using the *fftshift() command*. The highest spectral components are located around the homogeneous part at u = v = 0. The homogeneous parts in the pictures are suppressed. Real components exhibit mirror symmetry. Imaginary components possess point symmetry. *Simulation setup: 15 x 15 array, 2 mm z-distance, sphere encoder magnet, 0° angle position.*

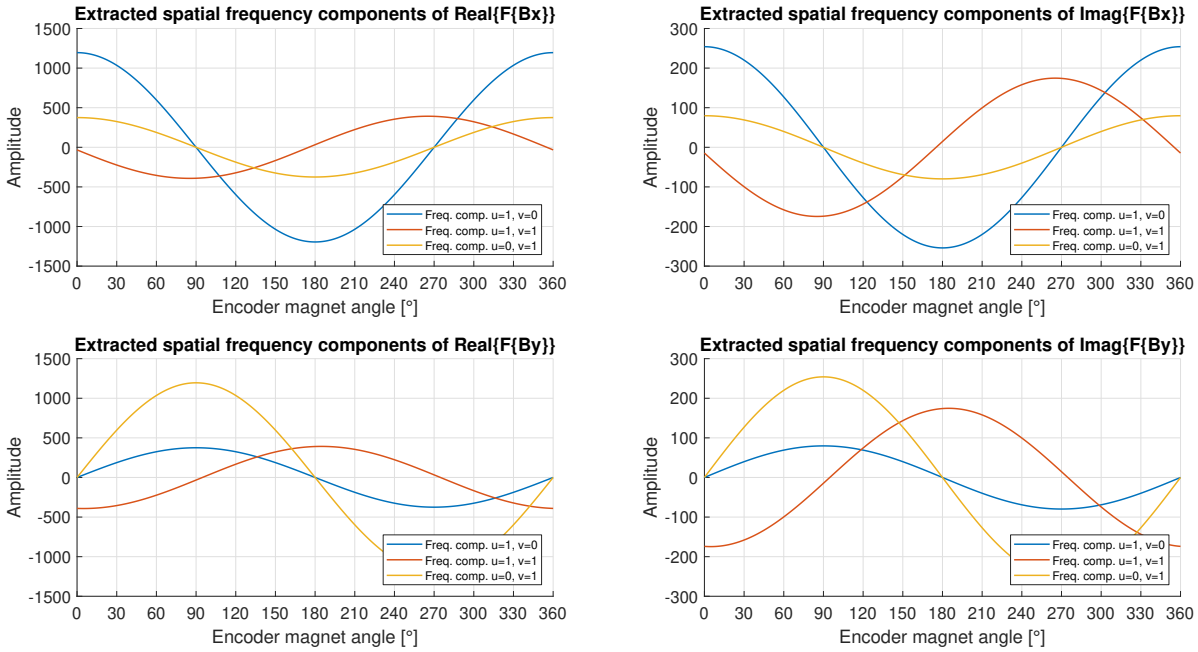


Figure 3.8: Progression of analyzed spatial frequency components over 360° rotation of encoder magnet. All components vary sinusoidally. The spatial component at coordinates pairs
(u = 1, v = 0) and (u = 0, v = 1) exhibits the same phase and no offsets, so in full accordance to ideal sine and cosine wave. The component in (u = 1, v = 1) exhibit a small phase shift and an offset. *Simulation setup: 15 x 15 array, 2 mm z-distance, sphere encoder magnet.*

From the graphs in 3.8 it is evident that the first spatial components in u- and in v-direction have the same phase and exhibit a sinusoidal progression over the full encoder magnet rotation, an ideal cosine wave progression for $F\{\mathbf{B_x}\}$ and a sine wave progression $F\{\mathbf{B_y}\}$. From this observation it follows the angle can be extracted directly by addition of the first pairs of the spatial frequencies in u- and v-direction in the corresponding spectrum images of $F\{\mathbf{B_x}\}$ and $F\{\mathbf{B_y}\}$.

Algorithm:
The signal flow diagram for the 2D-DFT based algorithm for angle calculation including stray field suppression is illustrated in Figure 3.9.

As a conclusion, the formulation for the angle calculation results to:

$$\varphi = arctan\left(\frac{Re\{F_{By}(1,0) + F_{By}(0,1)\} + Im\{F_{By}(1,0) + F_{By}(0,1)\}}{Re\{F_{Bx}(1,0) + F_{Bx}(0,1)\} + Im\{F_{Bx}(1,0) + F_{Bx}(0,1)\}}\right) \tag{3.7}$$

where $F_{By}(1,0)$, $F_{By}(0,1)$, $F_{By}(1,0)$ and $F_{By}(0,1)$ are the corresponding first spatial frequencies in u- and v-direction of measured field components $\mathbf{B_x}$ and $\mathbf{B_y}$.

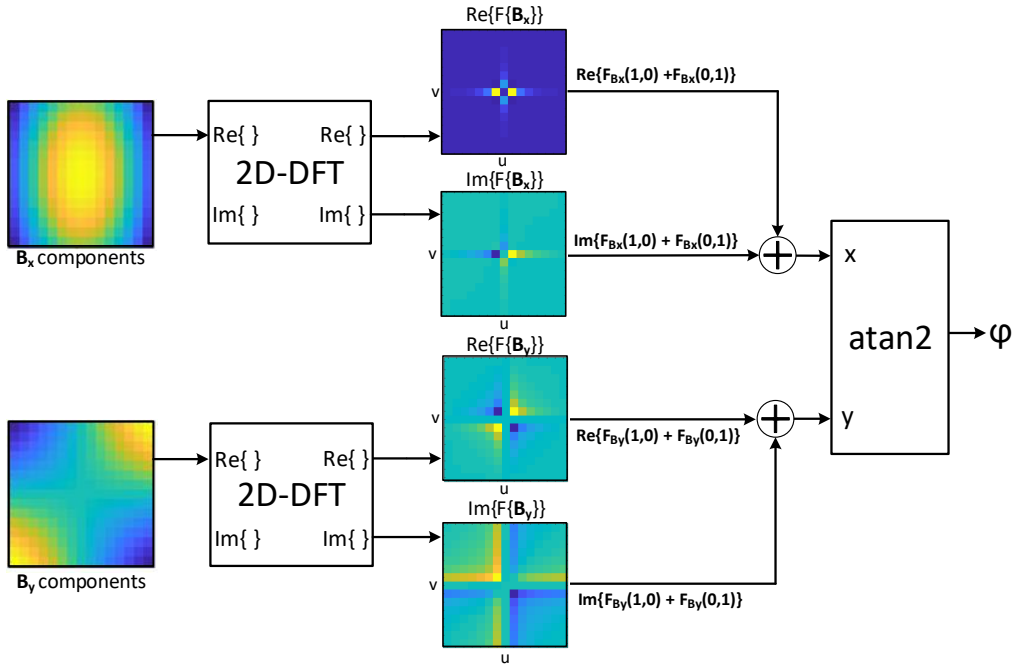


Figure 3.9: Signal flow diagram of 2D-DFT based algorithm for angle calculation including stray field suppression. The angle is obtained by summation of corresponding first spatial frequencies exclusively in u- and v-direction. Preprocessing for outliers and spikes illumination can be performed as a pre-filtering step, see more in Section 3.4.

Although the basis concept for the angle information extraction rely on the calculation of the curvature of the measured components in the xy-plane, the 2D-DFT is still valid for the methodical approach because it includes the total information about the field, inclusive its curvature. It can be understood as a universal tool for the present problem and it may be truth that this effect could be a possible advantage of the 2D-DFT based algorithm.

The application of the 2D-DFT based algorithm for angle information extraction exhibits the following advantages and disadvantages:

- **General advantages:**
  - Application of suitable preprocessing step possible, especially if the amount of sensor elements is sufficient to obtain a computational win using convolution theorem.
  - Explicit suppression of the particular spatial frequencies.
- **Advantages related to the application:**
  - Evaluation of particular spatial frequencies for information extraction.
  - Consideration of total field behavior including all local gradients and the global curvature of the field.

- **General disadvantages:**
  - Required computational effort needed for the calculation of 2D-DFT.
- **Disadvantages related to the application:**
  - No explicitly evaluation of field curvature. Due to it, it may be possible that the mechanical misalignment of encoder magnet will have a significant impact on the angular accuracy of this method. To compensate this possible weakness, the method for realignment of field components can be applied, documented in Section 3.5.2.

3.3.3 Approximation approaches

The main idea used for the algorithms based on approximation is to use a simple mathematical basis model for the approximation of the field components $\mathbf{B_x}$ and $\mathbf{B_y}$. The basis model can be a nonlinear model but it shall behave linear in the parameters [67]. In that case the parameters of the basis model can be estimated by formulation of a linear regression problem[7] and minimization of error between the measured data and the model.
Usually there are more data sets existent as the amount of unknown model parameters, what is practically always favored due to noise interference of the data. In this case the linear equation set becomes overdetermined and can be solved only by application of an optimization procedure. Furthermore, it might be the fact that the formulated linear equation system is not solvable because it does not represent a linear combination of equations, so that there is no exact solution existent. Also in that case one resorts to the optimization methods to obtain the best fit of the model parameters to the measured data.
The most common optimization criteria is the minimization of the sum of the squared errors, called least squares problem. The least squared solution is achieved by minimization of the squared Euclidean norm of the vector $\mathbf{r}(\mathbf{x}) = \mathbf{b} - \mathbf{A}\mathbf{x}$, which is called residual vector, so that [65]:

$$min||\mathbf{r}(\mathbf{x})||_2^2 = min||\mathbf{b} - \mathbf{A}\mathbf{x}||_2^2 \tag{3.8}$$

Matrix[8] $\mathbf{A}$ is an $m\ x\ n$ design matrix[9], $\mathbf{b}$ is a $m\ x\ 1$ vector including the measured data and $\mathbf{x}$ is $n\ x\ 1$ vector which includes the model parameter. The remained values in the residual vector $\mathbf{r}(\mathbf{x})$ give a knowledge about the quality of the model parameters estimation, which should be zero for an ideal fit.
For the solution of the least squares problem, e.g. the optimal values for the model parameters contained in $\mathbf{x}$ either the normal equation system is formulated as solved by an inverse of $(\mathbf{A}^T\mathbf{A})$, what is numerically not suitable due to the sensitivity of its operation or by application of decompositions algorithms such as QR or singular value decomposition SVD. Refer to Appendix B to get more information about solving of least squares problem.

[7] Curve fitting is often used in general linguistic usage.
[8] All elements in matrices and vectors are $\in \mathbb{R}$.
[9] Also denoted as regressor matrix. Common used notation used in parameter estimation is: $\mathbf{y} = \mathbf{X}\boldsymbol{\pi}$, where $\mathbf{X}$ is the regressor, $\boldsymbol{\pi}$ is the model parameter vector and $\mathbf{y}$ the output of the estimation problem [67].

The general advantage of the approximation approach is the noise robustness. Due to the estimation of the coefficient always an optimal solution for the model coefficients is obtained. However, such methods are extremely sensitive against outliers in the measurements due to the quadratically weighting of the resulting error. Therefore, one must make sure that the influence of possible outliers is mitigating or completely suppressed, e.g. in a pre-processing step.

For the mathematical representation of the field components 2D polynomial basis functions of second degree were chosen. This is the minimum required polynomial degree based on the concept, since the curvature of the approximated field shape can be extracted from the coefficients, which remain after the second derivatives in x- and y-direction. The idea regarding the angle calculation by the evaluation of polynomial coefficients is not new and was already proposed in [75], [52]. The followed method here can be understood as a general simplification of previous developed approaches and another evaluation combination of coefficients. The second degree of polynomials correspond also to the mathematical description of a magnetic dipole field present in the xy-plane, described in detail in Section 3.3.3. Higher polynomial degrees could be presumably useful and offer a higher approximation quality for other magnet shapes.

For the approximation based algorithm two methods were considered in this work. The standard 2D polynomial approximation and the modified approximation which includes the information about the physical geometry of the encoder magnet.

2D-polynomial approximation

Based on the assumption made above the following 2D polynomial basis functions for the approximation of the shape of field components measured in the xy-plane were chosen:

$$f_{B_x}(x, y) = p_{x0} + p_{x1}x + p_{x2}y + p_{x3}xy + p_{x4}x^2 + p_{x5}y^2 \tag{3.9}$$

for approximation of the field components measured in x-direction $\mathbf{B_x}$

$$f_{B_y}(x, y) = p_{y0} + p_{y1}x + p_{y2}y + p_{y3}xy + p_{y4}x^2 + p_{y5}y^2 \tag{3.10}$$

for approximation of the field components measured in y-direction $\mathbf{B_y}$. Coefficients p_4 and p_5 are needed for the extraction of the curvature information, coefficient p_3xy to model the xy-dependency, in p_1 and p_2 the information about the global gradients existent in the measured field components $\mathbf{B_x}$ and $\mathbf{B_y}$ can be represented, p_0 represents the homogeneous part of the field.

Figure 3.10 illustrates a result of field approximation by use of chosen 2D polynomial basis functions based on simulation data. The obtained fit by used functions is more than satisfactory.

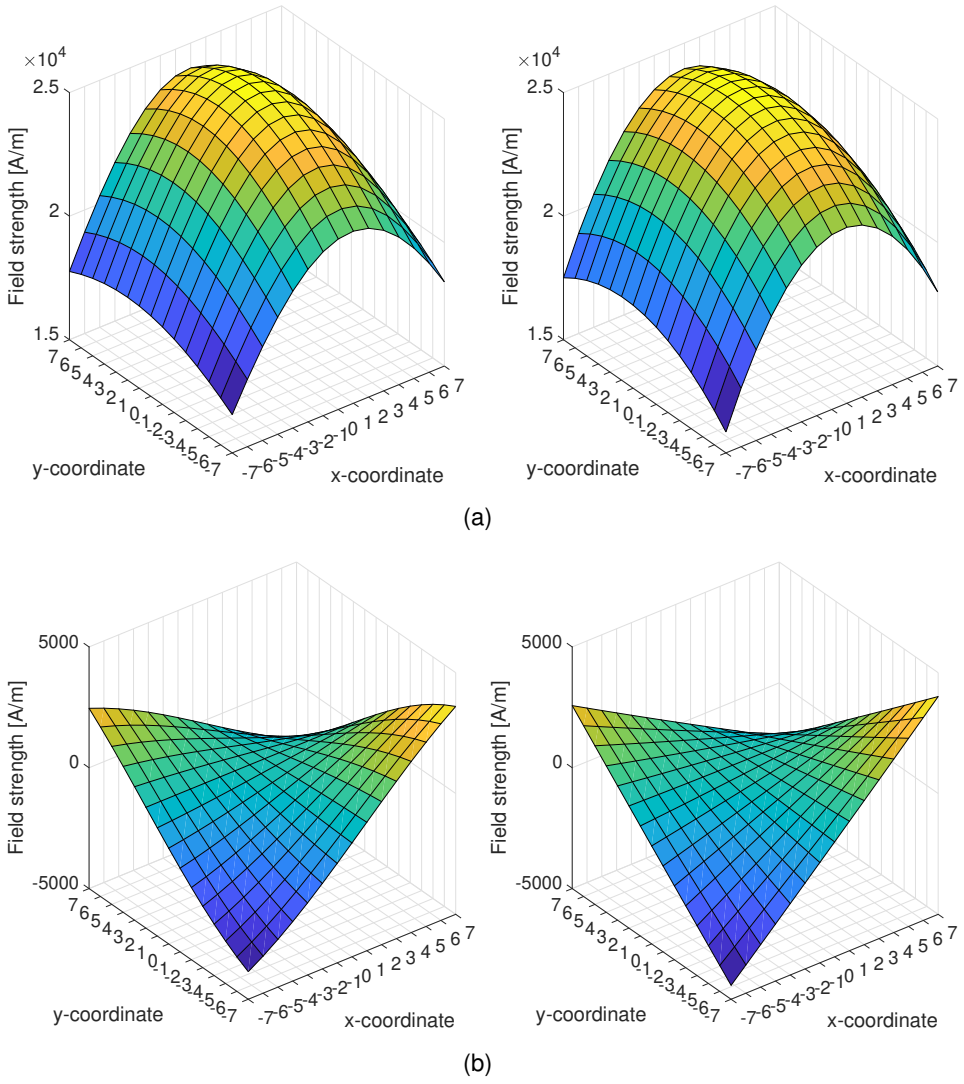


Figure 3.10: Comparison of simulated and approximated field components by use of chosen 2D polynomial basis functions. A tiny misalignment was set for the used encoder magnet in the simulation. (a) Left: Simulated $\mathbf{B_x}$ components. (a) Right: Approximated $\mathbf{B_x}$ components.(b) Left: Simulated $\mathbf{B_y}$ components. (b) Right: Approximated $\mathbf{B_y}$ components. An extremely well fit of the values of functions $f_{B_x}(x, y)$ and $f_{B_x}(x, y)$ is evident. *Misalignment: 0.2666 mm x-translation, 0.1333 mm y-translation, 1° encoder magnet tilt related to x-axis. Estimated coefficients:* p_{x0} *= 24581,* p_{x1} *= 165.2,* p_{x2} *= 129.8,* p_{x3} *= 1.59,* p_{x4} *= -105,* p_{x5} *= -32.6;* p_{y0} *= -73.9,* p_{y1} *= 129.9 ,* p_{y2} *= 27,* p_{y3} *= -67.5,* p_{y4} *= 0.78,* p_{y5} *= 0.38; Simulation setup: 15 x 15 array, 2 mm z-distance, sphere encoder magnet, 0° angle position.*

Figure 3.11 illustrates the progression of analyzed polynomial coefficients p_1 to p_5 of both polynomial functions $f_{B_x}(x, y)$ and $f_{B_y}(x, y)$ over 360° rotation in case of an ideal alignment encoder magnet.

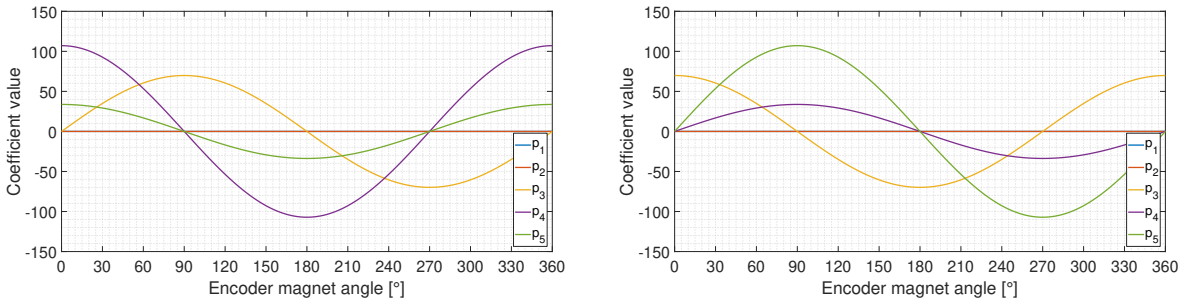


Figure 3.11: Progression of analyzed coefficients of chosen basis functions. (a) Coefficients of $f_{B_x}(x, y)$ function. (b) Coefficients of $f_{B_y}(x, y)$ function. All coefficients vary sinusoidally over the 360° rotation of encoder magnet. *Simulation setup: 15 x 15 array, 2 mm z-distance, sphere encoder magnet.*

From Figure 3.11 the dependence of the coefficients of approximated functions on the angular encoder magnet position becomes clearly. The coefficients p_3 to p_5 exhibit a sinusoidal behavior. The information about the angular position of the encoder magnet can be extracted by summation of curvature coefficients of each function. In addition, the coefficients which describe the xy-dependency could be included into the angle calculation by their mutual consideration from the functions. However, in this way both field functions would be coupled for the subsequent angle calculation. The focus in this thesis was set to explicit curvature evaluation and consideration of separately information from both data arrays $\mathbf{B_x}$ and $\mathbf{B_y}$, in accordance to the classical arctangent principle. Therefore, the xy-dependency was not included in the angle calculation.

Algorithm:
The algorithms for angle information extraction based on magnetic field approximation is represented graphically as a signal flow in figure 3.12.

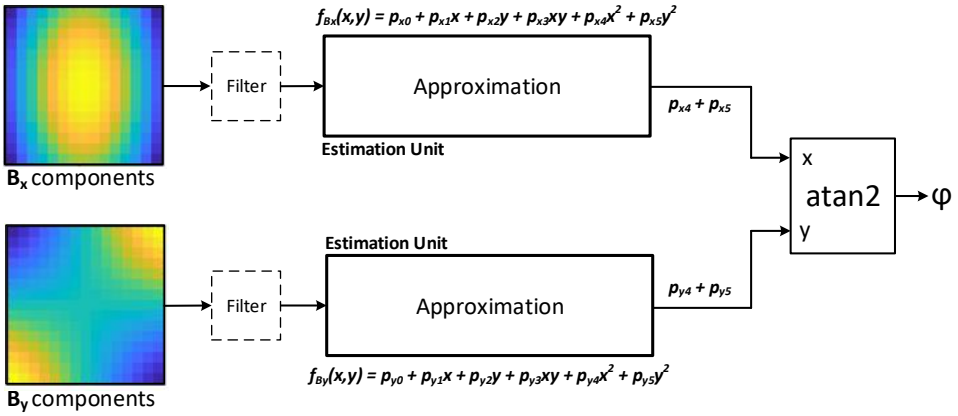


Figure 3.12: Signal flow diagram of approximation based algorithm for angle calculation including stray field suppression. Suitable pre-filtering is advisable in order to remove possible outliers.

In accordance to the information evaluation from the coefficients of chosen basis functions it follows for the calculation of the angle φ:

$$\varphi = arctan\left(\frac{p_{y4} + p_{y5}}{p_{x4} + p_{x5}}\right) \tag{3.11}$$

This algorithm brings the following advantages as well as disadvantages:

- **Advantages**
  - Robustness: Due to the estimation of the function coefficients and the obtain the optimal solution it is assumed that this method exhibits the highest stability against noise during angle calculation in comparison to other approaches. A filtering effect is implicitly present in this approach.
  - Lower sensitivity against misalignment effect: Due to the field approximation and evaluation coefficients which are describing the field behavior it is supposed that this method will exhibit a lower sensitivity against misalignment of the encoder magnet.
  - Suppression of disturbance fields which exhibit a gradient behavior: Not only a homogeneous stray field is suppressed in this approach but also the field components which exhibit a global gradient behavior measured in total xy-plane of the sensor array. Due to the explicit evaluation of the curvature of the field components the global gradient does not influence the angle calculation as well. This might be an additional practical advantage, especially if the distortion source is close to the sensor.
  - Independence: This approach does not require a pre-knowledge about the field shape of the used encoder magnet. It is assumed that the chosen polynomial functions are satisfied for the mathematical description of the measured field in the xy-plane.

- **Disadvantages**
  - Required computational effort: Since an optimization procedure to determine the functions coefficients needs to be performed, this approach leads to a relative high computational as well as implementation effort. This required effort is directly associated with the applied solving method.
  - Approximation error: Due to the simplest assumptions for the field behavior an approximation error is present in this method. This may affect the angle calculation in case of an existent misalignment of the encoder magnet in a negative way. This may directly unmake the second advantage.

Estimation formulation

For the following is is assumed that an quadratic sensor array consisting of N x N elements is used in the application. The amount of sensor elements is uneven. In order to get a symmetrical coordinate the matrices **X** and **Y** which contains the coordinates for each x- and y-coordinate

of the sensor array shall be constructed as follows:

$$\mathbf{Y} = \begin{pmatrix} N/2 & \cdots & N/2 & \cdots & N/2 \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ 2 & \cdots & 2 & \cdots & 2 \\ 1 & \cdots & 1 & \cdots & 1 \\ 0 & \cdots & 0 & \cdots & 0 \\ -1 & \cdots & -1 & \cdots & -1 \\ -2 & \cdots & -2 & \cdots & -2 \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ -N/2 & \cdots & -N/2 & \cdots & -N/2 \end{pmatrix} \tag{3.12}$$

$$\mathbf{X} = \begin{pmatrix} -N/2 & \cdots & -2 & -1 & 0 & 1 & 2 & \cdots & N/2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -N/2 & \cdots & -2 & -1 & 0 & 1 & 2 & \cdots & N/2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -N/2 & \cdots & -2 & -1 & 0 & 1 & 2 & \cdots & N/2 \end{pmatrix} \tag{3.13}$$

$N/2$ implies a floor operation. So, a symmetrical coordinates grid in full accordance to coordinate system is given. For the formulation of the problem the vector $\mathbf{a}_i$ is defined:

$$\mathbf{a}_i = \begin{bmatrix} 1 & x_i & y_i & x_i y_i & x_i^2 & y_i^2 \end{bmatrix}^T \tag{3.14}$$

which includes the sensor coordinates x_i and y_i multiplied in correspondence to the chosen polynomial basis functions. The design matrix $\mathbf{A}$ is constructed of this vectors, transposed to the row vectors. The i represents the index of the sensor plane point and the measured value at this point. Corresponding to it, the value measured in the xy-plane is denoted as b_i.
The vector $\mathbf{x}$ with model parameters, e.g. the coefficients of the 2D polynomial basis functions which need to be estimated is constructed corresponding to structure in $\mathbf{a}_i$ as:

$$\mathbf{x} = \begin{bmatrix} p_0 & p_1 & p_2 & p_3 & p_4 & p_5 \end{bmatrix}^T \tag{3.15}$$

Thus, the linear equation system $\mathbf{Ax} = \mathbf{b}$ is obtained by:

$$\begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \mathbf{a}_3^T \\ \vdots \\ \mathbf{a}_{N^2}^T \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ \vdots \\ b_{N^2} \end{bmatrix} \tag{3.16}$$

and needs to solved using of optimization method, e.g. least squares problem (see more in Appendix B).

Modified 2D polynomial approximation

In the frame of previous work in the ISAR project an algorithm based on the dipol equation and its partial derivatives was developed [75], [51], what laid a foundation for the modification of the approximation approach. The main idea here is to consider the real physical field shape of the encoder magnet by using the mathematical knowledge about its representation. Thus, the sensor is operating on a function which is much more similar to the field than the normal 2D polynomial. It is known by the theory that the mathematical description of the field of an ideal sphere magnet is given by the equation for magnetic dipole [75], [52]. The field of a magnetic dipole is given by Equation [75, extended]:

$$\vec{B}(\vec{r}) = \frac{\mu_0}{4\pi} \frac{3(\vec{r} - \vec{r_0})(\vec{\mu} \cdot (\vec{r} - \vec{r_0})) - \vec{\mu}|\vec{r} - \vec{r_0}|^2}{|\vec{r} - \vec{r_0}|^5} \tag{3.17}$$

where the position vector $\vec{r} = (x, y, z)$ represents distance from the magnetic dipol moment $\vec{\mu}$, i.e. the center of mass of the sphere magnet, positioned at $\vec{r_0} = (x_0, y_0, z_0)$.
From the Equation 3.17 the separate field components of $\vec{B}(\vec{r})$ measured in the sensor plane in x- and y-direction depending on the distance from the center of mass of the sphere magnet located at coordinates $(x_0 = 0, y_0 = 0, z_0 = 0)$ are resulting to [75]:

$$B_x(x, y) = \frac{\mu_0}{4\pi} \frac{3x(\mu_x x + \mu_y y + \mu_z z_d) - \mu_x(x^2 + y^2 + z_d^2)}{(x^2 + y^2 + z_d^2)^{5/2}} \tag{3.18}$$

$$B_y(x, y) = \frac{\mu_0}{4\pi} \frac{3y(\mu_x x + \mu_y y + \mu_z z_d) - \mu_y(x^2 + y^2 + z_d^2)}{(x^2 + y^2 + z_d^2)^{5/2}} \tag{3.19}$$

where z_d represents the z coordinate of the xy-plane where the field is measured. This coordinate can be considered as the z-coordinate of the sensor array plane. The denominator in these equations represents the field weakening behavior which is brought to bear the more the radial distance from the symmetrical central point field is existent, especially important in the near field. Note that this equation, and the exponent in the denominator term equals to $5/2$ is only valid for a sphere magnet.
The numerator terms in Equations 3.18 and 3.18 can be replaced by polynomial equations of corresponding degree. The magnetic moments and z_d are then represented in its coefficients. The exponent of the denominator term is replaced by n. Thus, the approximation equations for field components $B_x(x, y)$ and $B_y(x, y)$ result in:

$$f_{Bm_x}(x, y) = \frac{p_{x0} + p_{x1}x + p_{x2}y + p_{x3}xy + p_{x4}x^2 + p_{x5}y^2}{(x^2 + y^2 + z_d^2)^n} + p_{x6} \tag{3.20}$$

$$f_{Bm_y}(x, y) = \frac{p_{y0} + p_{y1}x + p_{y2}y + p_{y3}xy + p_{y4}x^2 + p_{y5}y^2}{(x^2 + y^2 + z_d^2)^n} + p_{y6} \tag{3.21}$$

The additional coefficients p_{x6} and p_{y6} need to be included into the estimation in order to model the homogeneous part of the field. Otherwise, the homogeneous stray field part will affect the estimation of coefficients for angle information extraction due to the coupling with the field weakening denominator matrix.
So, the idea here is to use the denominator which depends on exponent n and the z-distance z_0 in oder to get a better approximation result of the measured magnetic field by the sensor array. It is assumed that this consideration will affect the angle calculation from the curvature coefficients in a positive way so that the misalignment of the encoder magnet will have a smaller impact on the angle determination than in case of other approaches. For the ideal sphere magnet the exponent n is equal 2.5, according to 3.17. For other magnet shapes the exponent n can be different and needs to be be determined by optimization procedures by use of FEM/FDM magnetic simulation tool data. This represents the main disadvantage of the method because this approach is encoder magnet geometry dependent and implies a pre-optimization of the exponent in the Equation 3.20 and 3.21, if no sphere magnet but another one. e.g. disc, is used in the application.
In the following, the denominator term of functions 3.21 and 3.21 is denoted as field weakening denominator matrix:

$$\mathbf{D} = (\mathbf{X}.^2 + \mathbf{Y}.^2 + z_d^2).^n \tag{3.22}$$

where $\mathbf{X}.^2$ and $\mathbf{Y}.^2$ are point wise squared matrices representing the sensor array coordinates. Exponent n relates element wise to the each element of the matrix. This term needs to be included into the estimation procedure, i.e considered in the matrix $\mathbf{A}$.

To sum up, the modified approximation approach brings the following pros and cons:

- **Advantage:** Field flattening is included in the estimation. Therefore, a better field approximation can be achieved what may lead to a higher angle calculation robustness against misalignment effects, i.e. more stable angle calculation.
- **Disadvantage:** The exponent n and distance z_0 must be known / optimized before the operational phase to be able to use this approach, which is of course depends on shape of used encoder magnet in the application and installation parameters. These parameters / resulting estimation formalisms needs to be stored in the EEPROM of the device individually.

The developed approach can be understood as a simplification of the approximation algorithm presented in [52]. Since the developed approaches in this work imply the shifting possibility of the area for angle information extraction in order to obtain as much field symmetry as possible this approach does not include the insertion of x- and y-coordinates of the rotation axis projection point of the magnet into the field denominator term, what was previously proposed in [52].

3.3.4 Image processing filter based algorithms

As last candidates two approaches based on application of image processing filters were considered as possible methods for angle information extraction.
According to the concept the angle information is extracted from two-times differentiating of data contained in $\mathbf{B_x}$ and $\mathbf{B_y}$. The two-times numerical differentiating generally amplifies the noise in the image always significantly. Therefore, the methods here imply a direct and implicit curvature extraction in combination with the Gaussian low pass filter function. This special filter type is called LoG or Marr-Hildreth Operator. Gaussian low pass filters are applied in spatial domain as well as its pendants in the spatial frequency domain. This filter has e.g. in comparison to a mean filter the advantage of the rotation invariance, more discussed in Section 3.6.3 and the possibility of cut-off frequency control. The Laplacian of Gaussian filter masks are often applied in image processing for a reliable and noise resistant edge detections [78, Chap. 10], due to the filtering operation at first and a more simple detection of the edge by the threshold level equal zero in the second step [39].

The 2D Gaussian function is given by [39]:

$$f(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3.23}$$

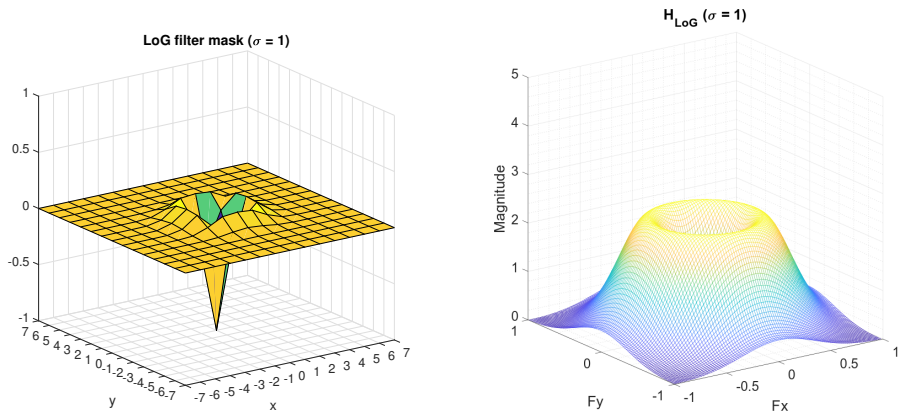
what is the analogy to the normal distribution function. The cut-off frequency of the filter can be adjusted by variation of the parameter σ^2, which represents the variance of the normal distribution function. The second order spatial derivatives of the 2D Gaussian filter function $g(x,y)$ in x- and y-direction are obtained by application of Laplace operator [39]:

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} \tag{3.24}$$

After taking the derivatives, the function by which the LoG filter kernel can be created results in:

$$h_{LoG}(x,y) = -\frac{1}{\pi\sigma^4} \cdot \left(1 - \frac{x^2+y^2}{2\sigma^2}\right) e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \tag{3.25}$$

where σ acts as the setting parameter of bandwidth. Figure 3.13 illustrates the Laplacian of Gaussian filter kernel in the spatial domain as well as its pendant in the spatial frequency domain.
In order to extract the curvature from $\mathbf{B_x}$ and $\mathbf{B_y}$ components either a convolution between the chosen filter mask size and the image or a direct extraction using one mask placed symmetrically on the projection point of the encoder magnet rotation axis in the sensor plane can be performed. In both methods the size of the filter kernel can vary from the minimum required one, consisting of 3 x 3 elements, to the amount of elements equal to the total array size N x N. This is graphically depicted in Figure 3.14.



(a) (b)

Figure 3.13: LoG filter kernel with σ = 1 consisting of 15 x 15 elements and its frequency response. For the calculation of frequency responses the freq2z() function was used, where interpolation and FIR-filtering are used. The amount of elements contained in 2D- frequency response was set to 100 x 100.

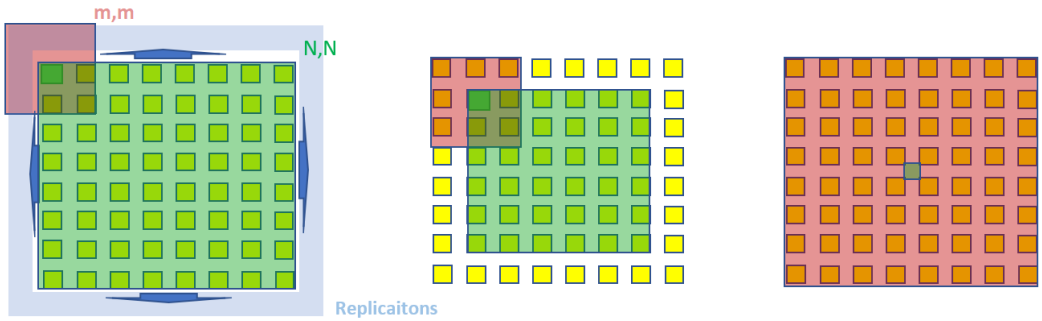


Figure 3.14: Comparison of possible extraction of angle information from a sensor array consisting of 8 x 8 elements based on image processing filtering methods. Red area is the filter mask. Green area is the extracted information. Yellow squares is the pixel information from sensor elements. Left: Convolution with inclusive boundary conditioning. Center: Reduced convolution without boundary conditions. Right: Direct curvature extraction by a single LoG mask of a size equal to the array size.

For the present problem in case of performing of a convolution the handling of boundary conditions[10] does not bring any information win and is therefore not needed. Thus, the convolution operation in this case using the smallest filter mask size would start as shown in Figure 3.14 (center). For the filter mask size equal to the total array size N x N only two values are extracted from the total sensor array.
The last point needs to be mentioned is the bandwidth optimization of the LoG, which can be adapted depending on the application and the filter kernel size. In order to approximate the

[10] However, in case of a suitable preprocessing of $\mathbf{B_x}$ and $\mathbf{B_y}$ components, e.g. for spikes or outliers illumination for the estimation, the application of appropriate boundary conditions are needed. See more in Section 3.4.1.

filter kernel shape a minimum filter mask of 7 x 7 elements is required, what represents one constraint for the LoG approach [39].

Convolution with LoG filter

The convolution between an image and a symmetrical filter mask is defined [39, adapted] as:

$$g'(x,y) = g(x,y) * h(x,y) = \frac{1}{S}\sum_{k=1}^{m}\sum_{l=1}^{m} g\left(x + \frac{m+1}{2} - k, y + \frac{m+1}{2} - l\right) \cdot h(k,l) \quad (3.26)$$

where $g(x,y)$ represents the input image, $h(x,y)$ is a filter kernel consisting of m x m elements and $g'(x,y)$ is the filtered output image. S denotes the scaling factor which is usually equal to the total amount of elements of the filter kernel. Alternatively, to avoid the convolution in spatial domain if the computational effort for the total processing chain *2D-DFT* $\rightarrow$ *filt* $\rightarrow$ *2D-IDFT* is less than the convolution operation, the convolution theorem can be used:

$$g(x,y) * h(x,y) \Leftrightarrow \mathcal{F}\{g(x,y) * h(x,y)\} = G(u,v) \cdot H(u,v) \quad (3.27)$$

where $g(x,y)$ is input image, $h(x,y)$ the filter kernel and $G(u,v)$ and $H(u,v)$ their Fourier transforms.

Algorithm:
The algorithms for angle information extraction based on convolution with LoG filter kernel is is represented graphically as a signal flow in Figure 3.12.

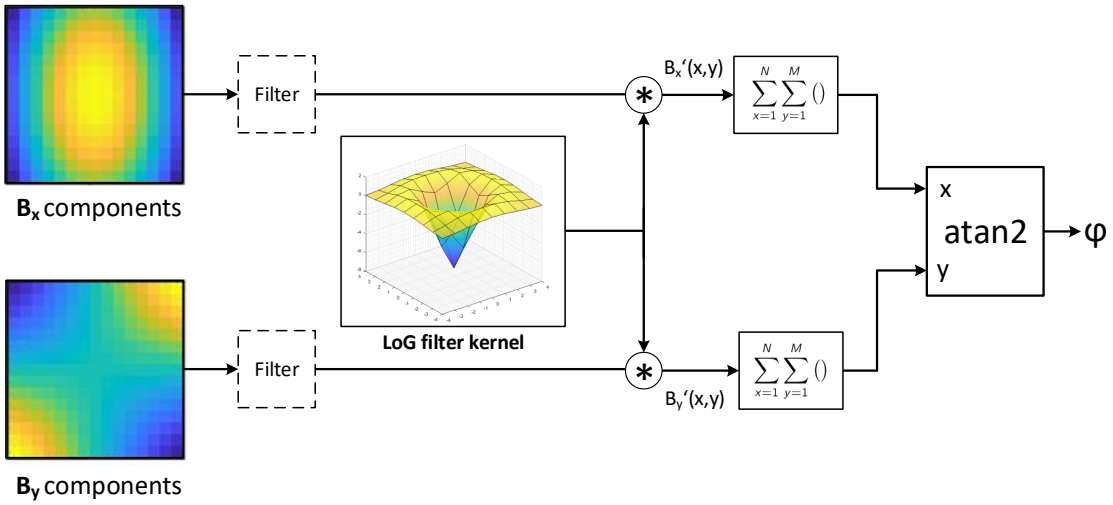


Figure 3.15: Signal flow diagram of LoG convolution based algorithm for angle calculation including stray field suppression. Preprocessing is advisable in order to remove possible outliers.

The angle is calculated by:

$$\varphi = arctan\left(\frac{\sum_{x=1}^{N}\sum_{y=1}^{M} B'_y(x,y)}{\sum_{x=1}^{N}\sum_{y=1}^{M} B'_x(x,y)}\right) \tag{3.28}$$

where $B'_x(x,y)$ and $B'_y(x,y)$ represent the images with extracted local curvatures from the sensor array and NM here is total amount of elements, according to the notation in the signal flow in Figure 3.15.

So, the convolution approach implies a summation of extracted local curvatures depending on the filter mask size contained in $\mathbf{B_x}$ and $\mathbf{B_y}$ data arrays.

This convolution approach brings the following drawbacks and assets:

- **Advantages**
  - Explicit curvature extraction.
  - Suppression of the global gradient as well as the approximation method.
  - Noise suppression due to included Gaussian low pass filter kernel.
  - May probably exhibit a high stability due to extraction of the curvature from multiple areas and their summation.

- **Disadvantages**
  - Arithmetic and implementation effort.
  - Direct curvature extraction: Although a filter kernel is included the extraction of field curvature is still direct. This may lead to a high sensitivity against noise in comparison to the approximation approaches.

Single LoG approach

The single LoG approach implies a direct extraction of the curvature by a „one shot“ operation. Thus, the angle is calculated by:

$$\varphi = arctan\left(\frac{B'_y}{B'_x}\right) \tag{3.29}$$

where B'_x and B'_y represent two values of extracted curvatures from the chosen areas in $\mathbf{B_x}$ and $\mathbf{B_y}$ data arrays.
This approach was developed primarily including the coupling with the tracking algorithm described in Section 3.6 and obtaining a degree of freedom for movement of the area, where the angle information is extracted from.
The single LoG approach includes the first three advantages of the convolution. In addition to it, its advantage and disadvantage is:

- **Advantage:** Simplicity and the lowest computation effort. The required amount of multiplication is the same as the amount of the array elements where the curvature is extracted from.
- **Disadvantage:** This approach may be probably less stable than the convolution method due to „one shot“ curvature extraction.

3.3.5 Effort comparison

To provide an understanding about the complexity of an numerical algorithm and required computation costs the amount of floating point operations[11] needed for the complete algorithmic procedure is used as the complexity measure. For a digital circuit or signal processing procedure in an ASIC the amount of required multiplications gives a satisfied measure for the computation costs [70]. The effort required by the methods for angle information extraction was rated on the required amount of multiplications needed for the calculation of x- and y-component used for the subsequent angle calculation in the CORDIC unit.

2D-DFT
For the computation of required multiplication for the 2D-DFT approach the Equation 3.2 was taken as reference. In accordance to the concept, 4 spectral components of resulting from each data array $\mathbf{B_x}$ and $\mathbf{B_y}$ are necessary for the extraction of angle information. Therefore, the total amount of multiplications required from the 2D-DFT based algorithm amounts to $8\ x\ N^2$, where N^2 is the total amount of sensor array elements.

Approximation
It is assumed that the determination of model parameter vector **x** the least squares method is applied. In case the simplest solving method for the formulated problem the model parameter vector can be obtained by classical normal equation solution:

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} = \mathbf{A_{NS}}\mathbf{b} \tag{3.30}$$

Since the values contained in the matrix **A** represent the coordinates of the sensor array elements, the matrix $\mathbf{A_{NS}}$ is fix. It consists of $m\ x\ N^2$ elements, where m is equal to the amount of model parameters and N^2 is the total amount of sensor array elements. Thus, the minimum required computational effort for the calculation of curvature coefficients of chosen 2D polynomial functions amounts to $4\ x\ N^2$ multiplications.
In case of an application of decomposition algorithms the amount of required multiplications will be larger and is strictly depended on the applied method. The application of such methods becomes especially important if the design matrix **A** is consisting of measurement values, affected by noise. The matrix can become ill-conditioned and one or another method can fail or become numerically unstable. This is especially for the solution of normal equations by use of

[11] Usually denotes in literature as *FLOP*. Performance in computer systems is generally rated on floating operations per second, denoted as *FLOPS*.

the inverse as shown in above equation the case. Therefore, it is not recommended to use this method. The required efforts of decompositions methods such as Cholesky factorization, QR and SVD algorithm for solving least squares problems can be taken from research publication of D. Q Lee [65] (see more in Appendix B).

Convolution

According to the derivation discussed in Section 3.3.4 for the purpose of angle information extraction no extension of boundary areas by padding is necessary. Hence, the resulting convolution procedure is carried out as shown in Figure 3.14 (center). For the calculation of required amount of multiplications the term $2(N-m+1)^2m^2$ was derived, where N is the amount of row or column elements of a quadratic sensor array and m is the amount of row or column elements of a quadratic filter mask. This term is valid for a sensor array and filter mask where the total amount of elements is uneven.
If the filter kernel is symmetrical, what is in the present problem the case, the convolution procedure can be optimized and the resulting amount of multiplication can be reduced. However, optimized convolution is not considered in the comparison.

Single LoG

The single LoG approach implies a „one shot“ multiplication of the filter mask with the elements from an area of interest. If the curvature of field components is extracted from the whole array, the amount of multiplications needed for the extraction of x- and y-signals counts to $2N^2$.

Comparison

The comparison of determined arithmetic effort required by the developed algorithms for angle information extraction is recorded in the Table 3.1. In addition, the implementation effort was also considered in the comparison.

	2D-DFT	**LoG convolution***	**Single LoG**	**Approximation****
Computational effort	$\circ$	$--$	$++$	$+$
Required multiplications	$8N^2$	$2(N-m+1)^2m^2$	$2N^2$	$4N^2$
Implementation effort	$\circ$	$-$	$++$	$+$

Table 3.1: Rating of the algorithms for angle information extraction based on its arithmetic and implementation effort. N^2 is the total amount of sensor elements contained in a quadratic sensor array. m^2 is the size of a quadratic filter kernel mask.

Effort rating scale: $++$ lowest, $+$ low, $\circ$ mean, $-$ high, $--$ highest
***:** The derived equation is valid for an uneven sensor array and uneven filter kernel mask.
****:** In case if the matrix **A** contains measured data the effort is increased significantly.

3.4 Preprocessing

In order to get a better angular accuracy from angle information extraction algorithms it may be useful, depending on chosen approach, to perform a suitable preprocessing of data in order to filter the noise and sporadic occurring high frequent noise components contained in the signal, e.g. eliminate occurring signal spikes[12] which may possibly enter the system by interference of high electromagnetic distortion fields generated by installed electronic components close to the sensor environment. Especially during the on-off time ranges of devices controlled by PWM signals the EMI can be extremely problematic. This impact may distort not only the magnetic signals but also sensor electronics during the data acquisition periods. The latter is especially crucial for the angle calculation algorithm based on approximation approaches due to quadratic weighting of the error caused by outliers existing in the measured data.
For the preprocessing two suitable filter candidates were considered in relation to the present problem: the Butterworth low pass filter and the median filter. Their functionality is described in the following briefly.

3.4.1 Median filter

A median filter is order-statistic nonlinear filter in spatial domain. Its response is based on ordering, i.e. sorting procedure of pixel which are covered by its kernel. The filtering is obtained by replacement the value in the center with the value determined after the sorting [78]. The value by which the central pixel is replaced is the median of the intensity of values covered within the filter mask size. Figure 3.16 illustrates the procedure of the mean filter.

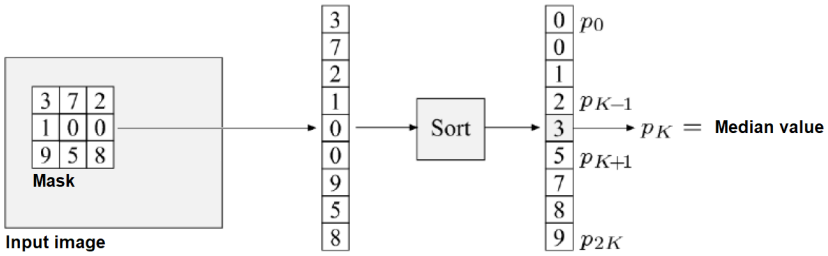


Figure 3.16: Illustration of median filter algorithm [59, modified]. After the ordering of the center value and its neighborhood values covered by filter kernel the central value is replaced by median value p_k.

The median filter provides extraordinary well capabilities for noise reduction of special types and cause considerably less smoothing of the image content than the linear filters of the similar filter kernel sizes. Especially of the presence of impulse noise (e.g. such as salt and pepper in the image processing) [78]. Furthermore, this filter type can be used for a reasonable defect pixel correction.

[12] Short duration electrical transients contained in a signal. Typical sources for spikes are malfunctions of power supply devices, regenerative brake systems, current induction, electromagnetic pulses and general electromagnetic interferences (EMI and EMP).

Figure 3.17 shows an example comparison of median and mean filter noise reduction results.

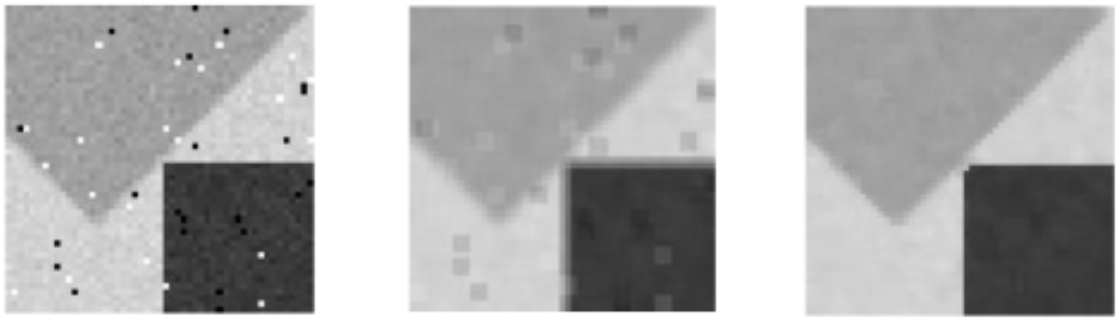


Figure 3.17: Median rank filter compared to mean filter [39]. Left: Input image interfered by salt and pepper noise. Center: Processed input image by application of mean filter of 3 x 3 kernel. Right: Processed input image by application of median filter. It is evident that not only the salt and pepper noise but also the background noise was removed from the image significantly better than by the mean filter. Also the level of information preserve is much higher than in case of mean filter.

As depicted in Figure 3.17 it is obvious that the median filter removes not only the high frequent spikes excellently exhibit also a better low pass result and line preserving in comparison to a mean filter.
The disadvantage of order-statistic nonlinear filter types is the processing speed due to the required sort procedure. There are also adaptive median filter existent [78, Chap. 5], where the intensity range and the filter kernel size is adapted during the filtering.

In general, by application of filters in image processing the problem with regard to the handling of boundaries elements is always existent. This may be not important at all if the input image size is large but in case for the present problem it may play a role if the amount of elements of the sensor array is relative small.The elements needed to be added to the picture behind its boundaries in order to be able to perform a complete multiplication / consider all values from neighborhood of the central point of the filter kernel. Following boundary conditions are applied in the image processing [39]:

- **zero padding:** Simple extension of values at the boundaries by zero.
- **replication:** The image content at boundary rows and columns is pulled outwards.
- **symmetric:** Filling of the required content outwards the image area with the center symmetrical extracted image elements.
- **circular:** The area needed for multiplication with filter kernel elements is filled by the content resulting from concatenation of the same image.

For the present problem sensor array problems either symmetric or replication boundary handlings are the most suitable.

3.4.2 Butterworth low pass filter

The Butterworth low pass filter is a filter in spatial frequency domain. Since the mean filter for the data preprocessing in the spatial domain was already chosen which exhibits also a low pass filter behavior no other spatial filters were considered. Alternatively, the Gaussian spatial filter can be also applied for this purpose. Application of the standard mean filter is not recommended due to its rotation variance which effect is discussed in Sections 3.6.3 and 5.3.1. The Butterworth low pass filter was chosen due to its better adaptability in comparison to other candidates taken into account [78, Chap. 4]:

- **Ideal low pass filter:** Is applicable, but causes ringing artifacts in the spatial domain which is caused by inverse Fourier transform of a rectangular function to sinus cardinalis function (and vice versa).
- **Gaussian low pass filter:** Offers only one degree of freedom what is the cut-off frequency.
- **Butterworth filter:** Offers two degrees of freedom, its order and the cut-off frequency.

The 2D transfer function of Butterworth low pass filter of order n is given by [78]:

$$H(u, v) = \frac{1}{1 + [D(u, v)/D_0]^{2n}} \tag{3.31}$$

where D_0 is the distance from the central point of the frequency domain which represents the cut-off frequency and $D(u, v)$ is the distance from the center of coordinates origin to a point (u,v) in the frequency domain, given by [59]:

$$D(u, v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2} \tag{3.32}$$

In Figure 3.18 the perspective view of 2D Butterworth low pass filter function of different orders n is depicted.

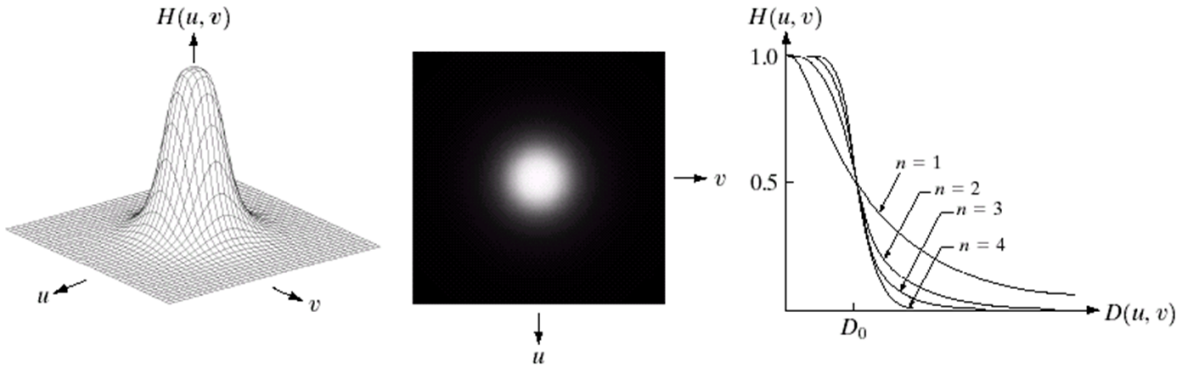


Figure 3.18: Butterworth low pass filter. Left: perspective view on transfer function. Center: View from above. (c) Radial cross section of different filter orders [72].

A 2D-DFT as well as 2D-IDFT needed for the application of frequency filters has already been developed within the ISAR project [64], [60]. In relation to the set specifications it has been proved that the application of this filtering type is relative feasible. The processing speed for the full chain *2D-DFT* $\rightarrow$ *filt* $\rightarrow$ *2D-IDFT* lays within acceptable processing delays [81]. Therefore, the Butterworth filtering is indeed a feasible candidate for the real-time preprocessing purposes.

3.5 Handling of misalignment effects

In the context of algorithm development a misalignment detection method as well as a realignment of field components was derived. These methods can be applied during the operational phase of the system in oder to perform on board diagnostics and monitoring the alignment of encoder magnet. For the basis of methods the knowledge about the ideal field behavior is used. These features are obtained by use of the field approximation method and are explained in this section.

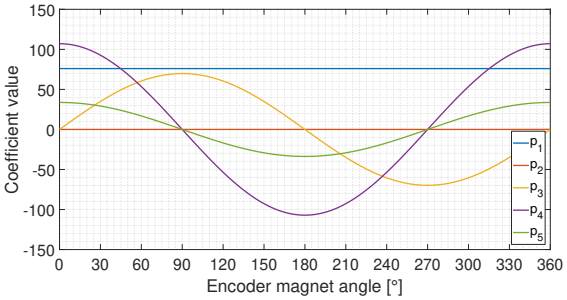
3.5.1 Misalignment detection

The most important property of the field at ideal alignment conditions is the fact that no global gradient is present in $\mathbf{B_x}$ and $\mathbf{B_y}$ field components. The symmetry of the field is perfect and a equilibrium state is existent. This is evident in Figure 3.5. In case of a misalignment of encoder magnet, the global gradient in the field becomes $\neq 0$. As a result of this, the misalignment presence can be detected by analyzing of coefficients of basis functions which represent the global gradient measured in the total xy-plane of the sensor array. Hence, this property can be used for the system analysis. In the following the derived algorithm for misalignment detection is described.

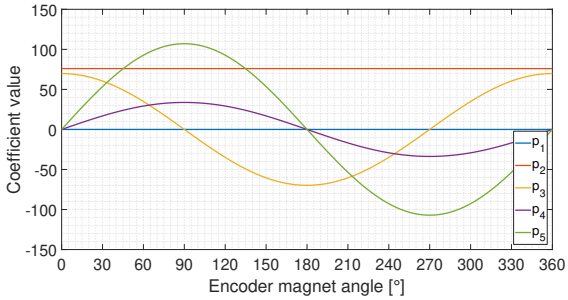
Algorithm:

1. Sample the components $\mathbf{B_x}$ and $\mathbf{B_y}$.
2. Approximate the basic functions
   $f_{B_x}(x, y) = p_0 + p_1 x + p_2 y + p_3 xy + p_4 x^2 + p_5 y^2$ and
   $f_{B_y}(x, y) = p_0 + p_1 x + p_2 y + p_3 xy + p_4 x^2 + p_5 y^2$
   for each data array $\mathbf{B_x}$ and $\mathbf{B_y}$ using the estimation unit.
3. Analyze the coefficients p_1 and p_2 for each data array $\mathbf{B_x}$ and $\mathbf{B_y}$:
   if $p_1 \neq 0$ and $p_2 \neq 0 \Rightarrow$ **Misalignment detected**.

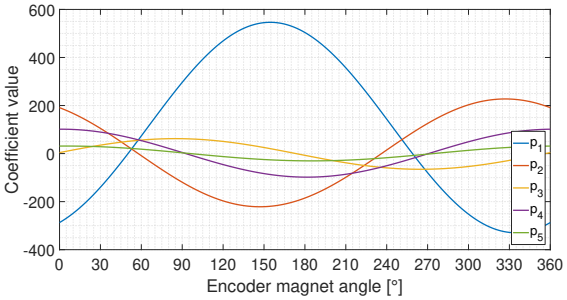
Within the context of the development of this detection algorithm, a pre-analysis in the simulation was carried out in oder to find out the dependencies between misalignment effect the estimated coefficients. Various misalignment cases were applied and the progression of estimated coefficients over the full encoder magnet rotation was analyzed. The results of two misalignment cases are shown in Figure 3.19.



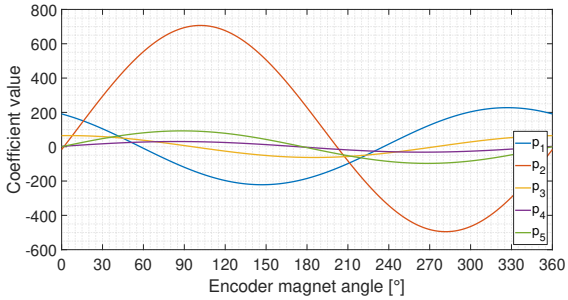
(a) Progression of coefficients of basis function for $\mathbf{B_x}$ field components: tilt misalignment.



(b) Progression of coefficients of basis function for $\mathbf{B_y}$ field components: tilt misalignment.



(c) Progression of coefficients of basis function for $\mathbf{B_x}$ field components: mixed misalignment.



(d) Progression of coefficients of basis function for $\mathbf{B_y}$ field components: mixed misalignment.

Figure 3.19: Misalignment handling pre-analysis. Progression of estimated coefficients. Homogeneous part not shown. (a) and (b): Single encoder tilt misalignment case of 2° x-tilt, no translation. The coefficient p_1 of approximated basis function for $\mathbf{B_x}$ field components as well as coefficient p_1 of approximated basis function for $\mathbf{B_y}$ field components are constant. (c) and (d): Mixed misalignment case: 3° x-tilt, -0.2666 mm y-translation, 0.4 mm x-translation. Besides the standard oscillation of coefficients p_3, p_3 and p_5 the coefficients for description of global gradient behavior exhibit oscillations as well. Their amplitudes are **significantly** higher than that of the curvature and xy-dependency coefficients. *Simulation setup: 15 x 15 array, 2 mm z-distance, sphere encoder magnet*.

The contemplation of progression of polynomial coefficients over the full encoder magnet rotation lead to the following conclusions:

- If the coefficient p_1 is $\neq 0$ or p_2 is $\neq 0$ and do not vary over the encoder magnet rotation, then a simple tilt of encoder magnet is existing.
- If the coefficients p_1 and p_2 vary over the encoder magnet rotation, then either a translation or a mixed case of translation and tilt of encoder magnet is existing.

The encoder misalignment impact on the progression and dependencies between coefficients is complex and was not covered widely within this work due to time limitations. This effects shall be investigated in the context of further work.
In order to reduce the computation effort during the operational phase the degree of polynomial functions could be reduced to the minimum required. It is assumed that this won't effect the

accuracy of detection algorithm much. The developed misalignment detection allows on board diagnostic of sensor array system and real time monitoring of the encoder magnet system condition. In case of occurred misalignment the sensor chip can send a warning signal to the main ECU.

3.5.2 Realignment of field components

The described field property above is also used in this method. The realignment of field components implies the shifting of the measured field components back to the origin. This method is also denoted as dynamic misalignment compensation and can be understood as a global gradient correction of measured field components.
In case of ideal alignment of the encoder magnet system a field symmetry is existent. As described previously in Section 3.3.3, the measured field components can be represented as 2D quadratic function which is absolutely symmetric with respect to the coordinates origin in case of an ideal alignment condition, shown in Figures 3.5. In case of a presence of misalignment, the maximum of components is shifted out of the symmetry point and a global gradient occurs in the measured $\mathbf{B_x}$ and $\mathbf{B_y}$ field components. From this it follows that by an estimation of the coefficient of basis functions which represent a 2D plane, the global gradient containing in $\mathbf{B_x}$ and $\mathbf{B_y}$ can removed by reinsertion of the coordinates of the sensor array into the estimated equations and subtraction of these values from the measured field data. So, a removal of the global gradient from the field is obtained and the field components are shifted back to the center. Figure 3.20 demonstrates the developed method applied on simulated field data carried out in the context of a pre-analysis.

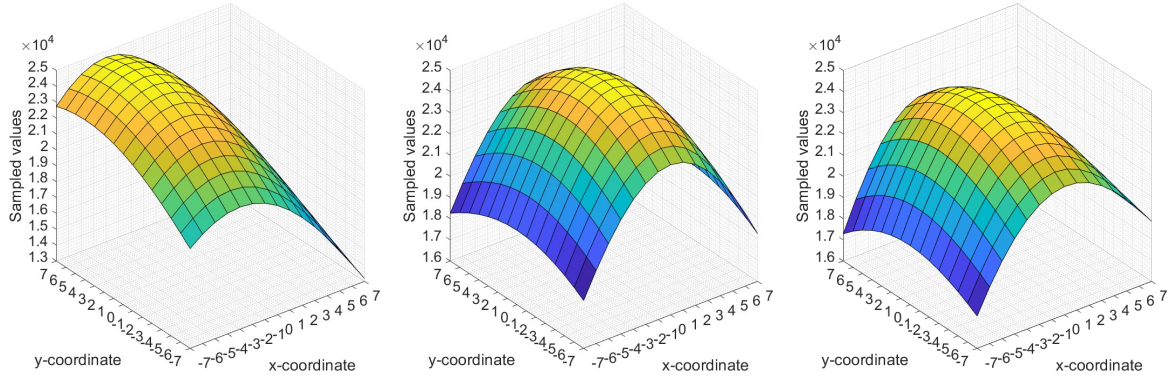


Figure 3.20: Realignment of field components. (a) Sampled $\mathbf{B_x}$ field components by existing misalignment of encoder magnet. (b) $\mathbf{B_x}$ field components by ideal alignment of encoder magnet. (c) Realigned $\mathbf{B_x}$ field components. Realigned field components show a very similar shape to the ideal components. *Misalignment: 0.2666 mm x-translation, -0.5333 mm y-translation. Estimated coefficients:* a_0 *= 21350,* a_1 *= -390,* a_2 *= 247. Simulation setup: 15 x 15 array, 2 mm z-distance, sphere encoder magnet, 0° angle position.*

In order to reduce the computational effort of this algorithm drastically it is suggested to perform the estimation explicitly by use of basis polynomial functions of reduced degree until the minimum required: only the set of linear 2D equations.

The algorithm for realignment of field components is as follows:

1. Sample the components $\mathbf{B_x}$ and $\mathbf{B_y}$.
2. Approximate the reduced basis functions
   $f_{global\ gradient\ Bx}(x, y) = a_{x0} + a_{x1}x + a_{x2}y$ and
   $f_{global\ gradient\ By}(x, y) = a_{y0} + a_{y1}x + a_{y2}y$
   for each data array $\mathbf{B_x}$ and $\mathbf{B_y}$ using the estimation unit.
3. Calculate the global gradient values
   $\mathbf{B_{x\ global\ gradient}} = f_{global\ gradient\ Bx}(\mathbf{X}, \mathbf{Y}) = a_{x0} + a_{x1} \cdot \mathbf{X} + a_{x2} \cdot \mathbf{Y}$ and
   $\mathbf{B_{y\ global\ gradient}} = f_{global\ gradient\ By}(\mathbf{X}, \mathbf{Y}) = a_{y0} + a_{y1} \cdot \mathbf{X} + a_{y2} \cdot \mathbf{Y}$.
4. Subtract the global gradient values from the sampled values
   $\mathbf{B_{x\ realigned}} = \mathbf{B_x} - \mathbf{B_{x\ global\ gradient}}$
   $\mathbf{B_{y\ realigned}} = \mathbf{B_y} - \mathbf{B_{y\ global\ gradient}}$.

The dynamic realignment of field components is only useful applicable for the 2D-DFT angle information extraction approach because this approach does not evaluate the curvature of the field explicitly while the LoG and approximation approaches are operating on the 2nd derivative. In these two algorithms the global gradient is already excluded from the angle calculation.

3.6 Angular error compensation by teach-in phase on spatial domain

According to the proposed functional split the first angular error correction possibility is given by use of degree of freedom coming from the spatial domain of the sensor array plane. The idea behind is the exploitation of field symmetries in case of the presence of encoder magnet misalignment.
The sensor plane area where the angle information is extracted from is reduced and set around the projection point of the rotational axis of the encoder magnet. In this way the area where the angle is extracted from becomes more symmetrical and leads automatically to an increase of angular accuracy.
In oder the use this concept the sensor array coordinates of absolute magnitude maximum point with the highest field strength within the sensor array needs to be localized. This point is denoted in the following as absolute magnitude maximum point (AMM point). The movable mask in the sensor plane where the angle information is extracted from is denoted as extraction mask. The algorithms developed for this purpose are called tracking algorithms and the combination of basis algorithms for angle information exaction is denoted as tracking extension.
The following conditions for the AMM point over the full encoder magnet rotation are valid:

- In case of a simple x- or y-translation of encoder magnet this AMM point is fixed.
- The presence of a single tilt of the encoder magnet related to the x- or y-axis does not affect the angle calculation because both extracted x and y signals are distorted in the same ratio. The AMM point is moving within the sensor plane but its mean value resulting over 360° rotation remains in the center of the sensor area.
- In case of a mixed misalignment consisting of a tilt and a translation of the encoder magnet the AMM point is moving as well. Its mean value vary in this case.
- By extremely large misalignments the AMM point is located outside the array plane. This represents the main accuracy limitation of this algorithmic extension.

Therefore, the projection point needs to be localized by application of an teach-in phase within a full encoder magnet rotation, where the field components $\mathbf{B_x}$ and $\mathbf{B_y}$ are sampled equidistantly with respect to the angle position of encoder magnet.
This approach brings the following assets and drawbacks:

- **Advantage:** Correction of angular error without any loss of processing accuracy. On the contrary, after the teach-in procedure the processing time becomes shorter due to the reduced sensor area for angle information extraction.
- **Disadvantage 1:** The angular error correction possibility is limited due to the amount of sensor array elements.
- **Disadvantage 2:** The reduction of „effective sensor area“ is associated with and increase of noise interfering effects. The robustness of angle calculation against noise reduces compulsorily by decrease of effective sensor amount for angle information extraction.

All developed candidates for angle information extraction described in Section 3.3 are expandable by this feature.
The coordinates of the mean AMM point are calculated by:

$$\mathbf{amm_{mean}} = \begin{bmatrix} round\left(\frac{1}{N}\sum_{n=1}^{N} amm_x(n)\right) \\ round\left(\frac{1}{N}\sum_{n=1}^{N} amm_y(n)\right) \end{bmatrix} = \begin{bmatrix} amm_{mean\,x} \\ amm_{mean\,y} \end{bmatrix} \tag{3.33}$$

where n is the rotational encoder magnet position and N is the total amount of equidistant encoder magnet positions within one full rotation of encoder magnet. $amm_x(n)$ is localized x- and $amm_y(n)$ the localized y-coordinate of an AMM point by applied tracking algorithms at set rotational encoder magnet position n.

The algorithm for the teach-in procedure looks as follows:

1. Set the rotational encoder magnet position to the initial step.
2. Sample the components $\mathbf{B_x}$ and $\mathbf{B_y}$.

3. Import the sampled components $\mathbf{B_x}$ and $\mathbf{B_y}$ into the unit for localization of AMM point coordinates x and y.
4. Store the coordinates in a data register.
5. Change the rotational encoder magnet position to the next step.
6. Repeat step 2 to 5, for all subsequent rotational encoder magnet position until N.
7. Calculate the mean AMM point by use of Formula 3.33.

Operational phase:
After the determination of the sensor array coordinates of the mean AMM point the movable mask for angle information extraction is placed symmetrical on the mean AMM point. The angle information is gained from the area covered by extraction mask.

The working principle of this algorithmic extension and its result based on a pre-analysis is shown in Figure 3.21.

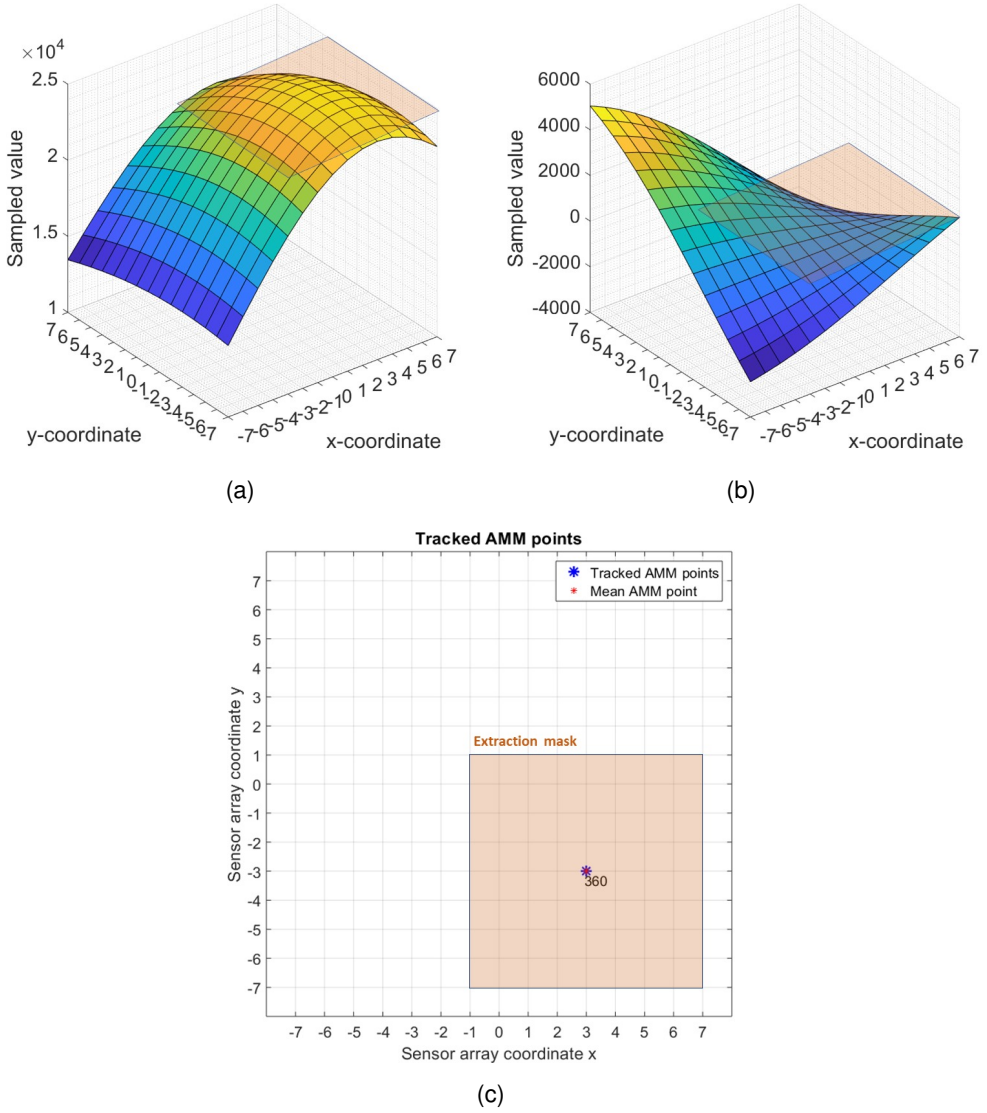


Figure 3.21: Schematically representation of tracking algorithm working principle and its results. Red plane represents the movable extraction mask, increased to maximum possible one in this simulation setup. (a) and (b) Sampled $\mathbf{B_x}$ and $\mathbf{B_y}$ components with set extraction mask on the maximum capture of field symmetry area. (c) Localized AMM points and mean AMM point over one full encoder magnet rotation. *Simulation setup: 15 x 15 array, 2 mm z-distance, -0.4 mm y-translation, 0.4 mm x-translation, sphere encoder magnet, 360 steps encoder magnet rotation.*

In the following a simplification for the determination of the magnitude is done and the candidates for the tracking algorithm, i.e. for calculation of coordinates of AMM points, are presented as well as their advantages and disadvantages discussed.

3.6.1 General simplification for magnitude determination

The basis principles of algorithms for localization of AMM points within the sensor plane is the analysis of gradients contained in the image as well as a direct comparison of magnitudes. At first before the tracking algorithm development a possible simplification for both determinations was assessed. In order to determine the point of maximal field strength measured by a sensor array the largest magnitude from the measured components $\mathbf{B_x}$ and $\mathbf{B_y}$ together needs to be localized within the full array area.
The resulting magnitude picture, in the following denoted as $\mathbf{M_f}$, can be also expressed as magnitude field or magnitude matrix and has to be element-wise calculated by

$$\mathbf{M_f} = \sqrt{\mathbf{B_x}.^2 + \mathbf{B_y}.^2} \tag{3.34}$$

according to the universal vector magnitude calculation. The local magnitudes of measured field components are contained in this picture. This operation is the general required step of the developed algorithms. For the approximation based algorithms this operations need to be performed only once, for gradient filter based algorithms twice.
For the analysis of magnitudes the picture $\mathbf{M_f}$ is treated as a picture where the gradients needs to be extracted from. Here, one can find the association to the gradient analysis in the image processing [78], [59].
In general, the gradient of an image f at its (x, y) coordinates, which points in the direction of the rate of change of f at location (x, y), is defined as the two dimensional column vector [78]:

$$\nabla f(x,y) = grad(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix} \tag{3.35}$$

The direction of the gradient / two dimensional column vector $\nabla f(x,y)$ can be calculated by [59]:

$$\varphi = arctan\left(\frac{g_x}{g_y}\right) \tag{3.36}$$

The length / magnitude of the vector ∇f, which is denoted as $M(x,y)$ is calculated by [78]:

$$M(x,y) = ||\nabla f|| = mag(\nabla f) = \sqrt{g_x^2 + g_y^2} \tag{3.37}$$

In common practice it is referred to this image $M(x,y)$ as the gradient image (note: also the vector norm notation $||\nabla f||$ is often used in image processing).
To reduce computation effort, the following approximation of the square roots operations is used in image processing [59]

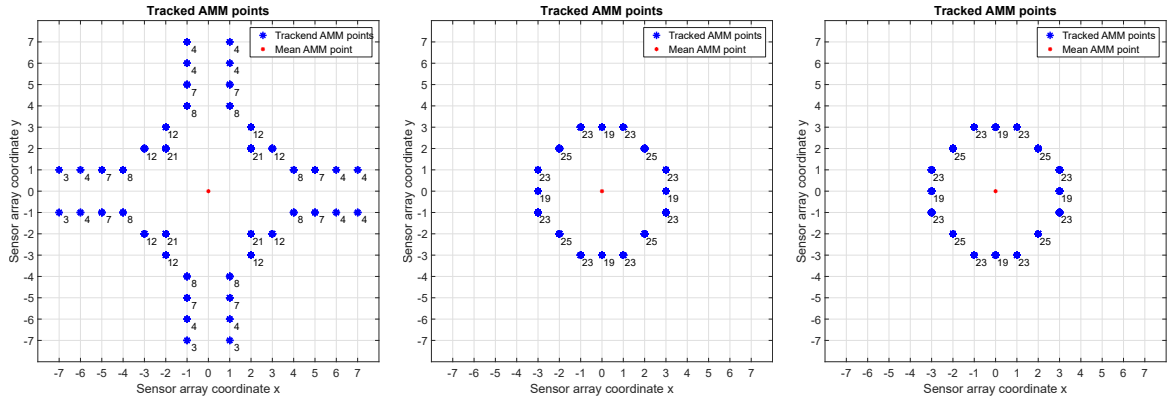
$$M(x,y) \approx g_x^2 + g_y^2 \tag{3.38}$$

or, more simplified, by use of the following approximation of the squares and square roots operations [78]

$$M(x, y) \approx |g_x| + |g_y| \tag{3.39}$$

where the vertical bars denote the absolute values. This approximation operation is computationally of course much more suitable.

Analysis:
Firstly, for the development of the algorithms the following consideration was taken into account: The square root operation is necessary, to come back to the same units scales, defined in euclidean plane after the computation of $g_x^2 + g_y^2$. However, in present case the localization of the maximum magnitude of gradients located in the magnitude picture $\mathbf{M_f}$ is required. Therefore, all local gradients contained in this picture need to be compared with each other, what means that in this respect no square root operation is needed to be performed.
Secondly, it needs to be tested if the simplest approximation of the magnitude is satisfied and applicable for the present problem because it reduces the computational effort significantly.
To determine the accuracy of three possible gradient calculations a preliminary simulation test was carried out, where the encoder magnet got a misalignment in the form of a x-tilt of 10° and was rotated over 360°. The sensor array size was set to 15 x 15 and the magnet placed 5 mm above the sensor plane. The AMM points localized by use of all three gradient calculations methods. Figure 3.22 shows the comparison between localized AMM points with $\mathbf{M_f} \approx |\mathbf{B_x}| + |\mathbf{B_y}|$, $\mathbf{M_f} \approx \mathbf{B_x}.^2 + \mathbf{B_y}.^2$ and $\mathbf{M_f} = .\sqrt{\mathbf{B_x}.^2 + \mathbf{B_y}.^2}$.



(a) (b) (c)

Figure 3.22: Results of preliminary test in frame of tracking algorithms development. (a) Localized AMM points with simplest gradient calculation method. (b) Localized AMM points with method without square root operation. (c) Localized AMM points by using the Formula 3.34. *Simulation setup: 15 x 15 array, x-tilt = 10°, 5 mm z-distance, sphere encoder magnet, full encoder magnet rotation.*

From the obtained results it is evident that while the most simplified method exhibits an insufficient localization accuracy, the method without square root operations shows the same localization accuracy as the square root method. Made consideration about saving of the

root operation is hereby confirmed. In accordance to the results in Figure 3.22(a) the simplified method of gradient calculation is sorted out. Its localization error due to simplification of magnitude determinations is significant and hence the AMM point localization accuracy not satisfied.

3.6.2 Butterworth low pass filter based algorithm

The first candidate for the tracking algorithm is a simple filtering in the frequency domain by application of the Butterworth low pass filter. The filter is applied for suitable noise suppression. Their cut-off and order needs to be identified in relation to the application in order to do not deform the shape resulting from $\mathbf{B_x}$ and $\mathbf{B_y}$ field components significantly and do not distort the localization accuracy of the AMM points hereby. Figure 3.23 illustrates the signal flow diagram of this algorithm.

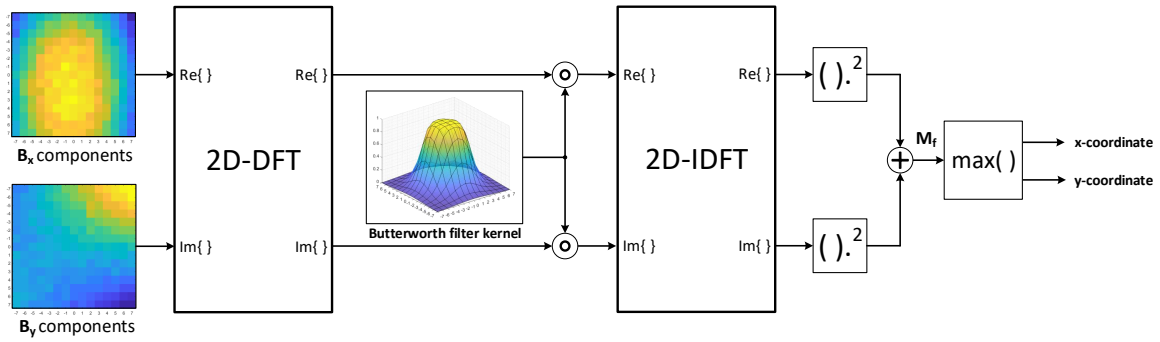


Figure 3.23: Signal flow diagram of Butterworth low pass filter based tracking algorithm.

The input signals are treated as a complex input in to 2D-DFT, so it is performed as $\mathcal{F}\{\mathbf{B_x} + j\mathbf{B_y}\}$. The components are element-wise multiplied by filter kernel coefficients and subsequently inverse transformed. After the filtering of components the magnitude image $\mathbf{M_f}$ is calculated where the maximum value is detected by a comparison of elements included in it. The advantages and disadvantages of this method are:

- **Advantages**
  - Robustness: Due to filtering in frequency domain a significant noise suppression is obtained.
- **Disadvantages**
  - Field deformation: Deformation of the shape resulting from $\mathbf{B_x}$ and $\mathbf{B_y}$ components will occur due to the filtering.
  - Computation effort: High effort due to the total chain *2D-DFT* $\rightarrow$ *filt* $\rightarrow$ *2D-IDFT*.
  - Implementation effort: 2D-DFT and 2D-IDFT units are required.

3.6.3 Gradient filter based algorithms

Classical gradient filter analysis methods from image processing [78, chap. 10], [59], applied in most cases for edge detection, were considered as possible candidates for the tracking algorithms as well. The gradient filter based approaches imply convolution operation:

$$M_f{}'(x, y) = M_f(x, y) * h(x, y) \tag{3.40}$$

where $M_f(x, y)$ is the magnitude image, $h(x, y)$ is applied filter kernel mask and $M_f{}'(x, y)$ is the processed result where the information about gradients is contained in.
The gradient filter based approaches have here a general disadvantage in relation to the detectivity possibility of the AMM points. The range of area where the AMM points is localizable shrinks from all image sides by the half size of the filter kernel mask. Application of boundary conditions described in Section 3.4.1 do not lead to an information win. The second disadvantage of filter based methods is the filter mask size itself. This disadvantage is resulting in case of a combination of derivatives and coupled filter functions. Depending on coupled filter function a particular kernel size is required in order to approximate the filter function shape sufficiently so that no filter kernel rotations variance[13] occurs [39], [59].

Laplacian of Gaussian filter based method

This algorithm is based on calculation of curvatures contained in the image, in similarity to the LoG convolution approach for angle information extraction described in Section 3.3.4. The applied Laplacian of Gaussian filter kernel function is given in Equation 3.25.
The idea in this approach is to evaluate the curvature of the magnitude image $\mathbf{M_f}$ including a low pass filtering by using of the Laplacian filter kernel. By convolution of $\mathbf{M_f}$ with the LoG mask an image of local curvature values is obtained. The coordinates of the image where the highest value is present represents the localized AMM point. The signal flow diagram of the LoG filter based tracking algorithm is shown in Figure 3.24.

[13] Rotation variance describes the property of a filter kernel. There should be no change in the obtained characteristics after the convolution with the filter mask if an object is rotated in the image. If the filter kernel does not exhibit such effect then is said to be rotation invariant. In such case the filter kernel coefficients must become zero at their boundaries. See also a explanation in Section 5.3.1.

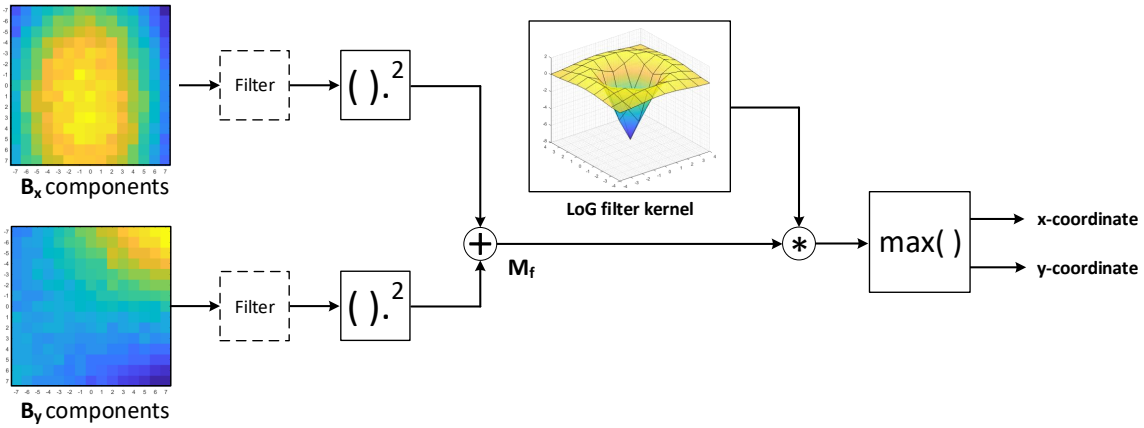


Figure 3.24: Signal flow diagram of Laplacian of Gaussian filter based tracking algorithm.

In addition, before the computation of $\mathbf{M_f}$ the values in $\mathbf{B_x}$ and $\mathbf{B_y}$ could be suitable preprocessed by a low pass filter application.
The pros and cons of this method are as follows:

- **Advantages**
  - Filtering: The Gaussian filter mask is included in the kernel.
  - Rotation invariance: It is possible to obtain the invariance by suitable approximating of the filter kernel.
- **Disadvantages**
  - Field deformation: Deformation of the shape of the components is associated with filtering by included Gaussian filter mask.
  - Minimum required mask size: In order to obtain rotation invariance a minimum kernel size of 7 x 7 elements is required [39].
  - Localization at the boundaries: Due to the image filter based approach the localization area shrinks at the boundaries.

Derivative of Gaussian filter based method

At the next candidate for the tracking algorithm the Derivative of Gaussian filter (DoG). The idea behind is to smooth the image first in order to obtain more noise stability and apply afterwards a direct gradient filter. By partial derivatives of Laplacian filter kernel function $G(x, y)$ given in Equation 3.23 the filter masks for the gradients detection result in:

$$DoG_x = \frac{\partial G(x,y)}{\partial x} = -\frac{x}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3.41}$$

for gradient extraction in x-direction as well as in y-direction:

$$DoG_y = \frac{\partial G(x,y)}{\partial y} = -\frac{y}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3.42}$$

For this gradient filter based algorithm, the square operation needs to be aplied in two steps. In the first step to determine the magnitude image $\mathbf{M_f} = \mathbf{B_x}.^2 + \mathbf{B_y}.^2$ and in the second step to calculate the gradients contained in the magnitude image $\mathbf{M_f}$, after the partial gradient filter application. This approach represents the classical method for viewpoint independent edge detection in the image [59]. But especially in the present case, the interest is in the gradient contained in the image which is equal to zero, in accordance to the localization of maxima by 1st derivative. The signal flow diagram is depicted in Figure 3.25.

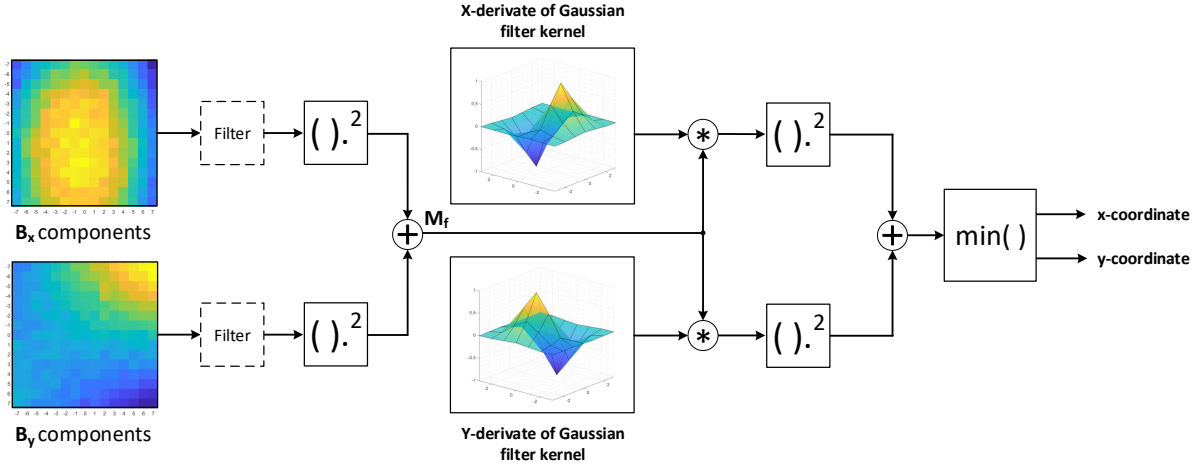


Figure 3.25: Signal flow diagram of derivative of Gaussians filter based tracking algorithm.

The advantages and disadvantages look as follows:

- **Advantages**
  - Filtering: Same as previous.
  - Rotation invariance: Same as previous.
  - Higher accuracy in comparison to LoG: Indeed a better localization of AMM point can be obtained due to considering of all points contained in the mask for each gradient in one direction exclusively.
  - Evaluation: More suitable evaluation is given due to threshold level = 0.
- **Disadvantages**
  - Field deformation: Deformation of the shape is present here as well.
  - Minimum required mask size: Same as previous.
  - Localization at the boundaries: Same as previous.
  - Computational effort: Two additional square operations are required in order to obtain

viewpoint independent localization of AMM point. This is especially important for AMM localizations at 45° encoder magnet position.

Prewitt filter based method

Derived from the previous method the Prewitt gradient filter was taken into the consideration. This filter type comes without a low pass filter kernel and it is rotation variant. This was chosen consciously in order to investigate firstly, the effect if no filter mask is combined with the gradient filter and due to this no field deformations is caused and secondly, how is the impact of rotation variance in the present problem. The alternative filter including a slight low pass behavior would be the Sobel operator.
The processing procedure is the same as for the previous candidate. As a proposal it is suggested to apply a filter kernel mask of 5 x 5 elements what is a good compromise between the minimum required symmetrical mask of 3 x 3 and 7 x 7 elements. The filter kernels of a Prewitt operator consisting of 5 x 5 elements for gradient calculation in x- and y-direction is given by:

$$h_{Px} = \begin{pmatrix} 2 & 1 & 0 & -1 & -2 \\ 2 & 1 & 0 & -1 & -2 \\ 2 & 1 & 0 & -1 & -2 \\ 2 & 1 & 0 & -1 & -2 \\ 2 & 1 & 0 & -1 & -2 \end{pmatrix}, \quad h_{Py} = \begin{pmatrix} 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 \\ -2 & -2 & -2 & -2 & -2 \end{pmatrix} \tag{3.43}$$

The application of Prewitt gradient filter leads to the following assets and drawbacks:

- **Advantages**
  - No explicit field deformation in comparison to LoG and DoG approaches: There is no low pass filter kernel included.
  - Implicit Filtering: A filtering effect is still existent if the filter mask is increased. In this case the gradient extraction is more averaged over the spatial area.
  - Higher accuracy in comparison to LoG and DoG: Due to smaller area of the extraction mask size and no inclusive low pass filtering the smoothing effect is smaller.
  - Evaluation: As well as valid for DoG approach, the threshold level for the detection of the AMM point is equal to zero.
- **Disadvantages**
  - Noise sensitivity: Due to direct gradient extraction a higher noise impact is existent. However, this is compensated by an increase filter kernel size.
  - Rotation variance: Rotation variance is unfortunately ot avoidable for this approach. This may probably affect the localization of AMM points during rotation in a negative way.
  - Localization at the boundaries: Same as previous candidates.
  - Computational effort: As well as all gradient based approaches two additional square operations are required also for this method.

3.6.4 Approximation based algorithms

For the localization of AMM points the methods based on approximation of basis function for the field components were considered as well. The basis function are chosen in correspondence to the approximation based algorithms for angle information extraction described in Section 3.3.3.

Approximation including the reinsertion of sensor plane coordinates

The first candidate is the approximation of basis function given in Equations 3.9 and 3.10 and subsequent calculation of function values by a reinsertion of sensor plane coordinates into the estimated functions. After the calculation of function values the magnitude image is determined. The maximum value contains in it represents the AMM point. The signal flow diagram of this method is shown in Figure 3.26.

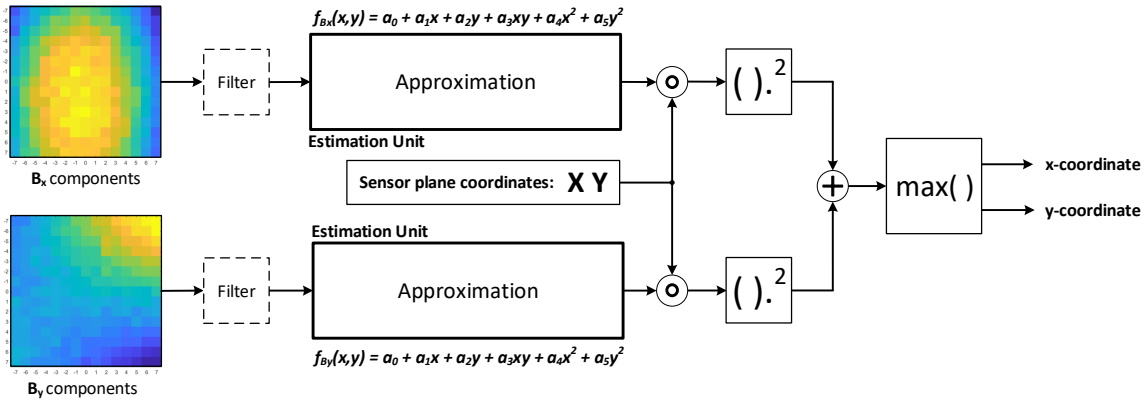


Figure 3.26: Signal flow diagram of approximation based tracking algorithm.

In addition, it would be useful to ensure that $\mathbf{B_x}$ and $\mathbf{B_y}$ data arrays are cleaned of outliers in order to guarantee a robust estimation procedure.
The first main advantage of this method is its robustness. The comparison of values for AMM localization is always done on function values where no noise is existent. The second main advantage is the localization possibility of AMM points also at the boundaries of the sensor plane. However, on the other side, this method requires the highest computational effort due to the multiplication of function coefficients by sensor coordinate values.
To sum up, the pros and cons of this approach are:

- **Advantages**
  - Robustness: Due to optimal solution of estimation and implicitly present filtering effect.
  - No field deformation: No low pass filtering which can deform the field shape is performed.

- Accuracy: In comparison to previous methods the localization procedure is made always on function values which are free of noise.
- Localization at the boundaries.

- **Disadvantages**
  - Approximation error: Due to the chosen basis function an approximation error is existent.
  - Computational effort: A significant computational effort is associated with the reinsertion of sensor array coordinates **X** and **Y**.
  - Implementation effort: An estimation and function values calculation unit is required.

Simplified approximation and analytical method

This method combines the strength of the approximation method described above and simple analysis. Compared with the approximation method in this approach not the shape resulting from $\mathbf{B_x}$ and $\mathbf{B_y}$ components but the magnitude picture $\mathbf{M_f} = \mathbf{B_x}.^2 + \mathbf{B_y}.^2$ is approximated. The AMM points is found analytically, by partial derivatives of the basis function. For the basis function the following polynomial function of second order without xy-dependency is chosen:

$$f_M(x, y) = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 y^2 \tag{3.44}$$

By taking the first derivatives with respect to the x- and y-direction it follows:

$$\frac{\partial f_M(x, y)}{\partial x} = a_1 + 2a_3 x, \quad \frac{\partial f_M(x, y)}{\partial y} = a_2 + 2a_4 y \tag{3.45}$$

To find the global maximum the partial derivatives are set equal to zero and rearranged for x and y. So, the coordinates of the AMM point are localized by:

$$amm_x = round\left(-\frac{a_1}{2a_3}\right) \quad amm_y = round\left(-\frac{a_2}{2a_4}\right) \tag{3.46}$$

Figure 3.27 illustrated the signal flow diagram of this approach. Also here one should make sure that no outliers are present in the $\mathbf{B_x}$ and $\mathbf{B_y}$ data arrays.

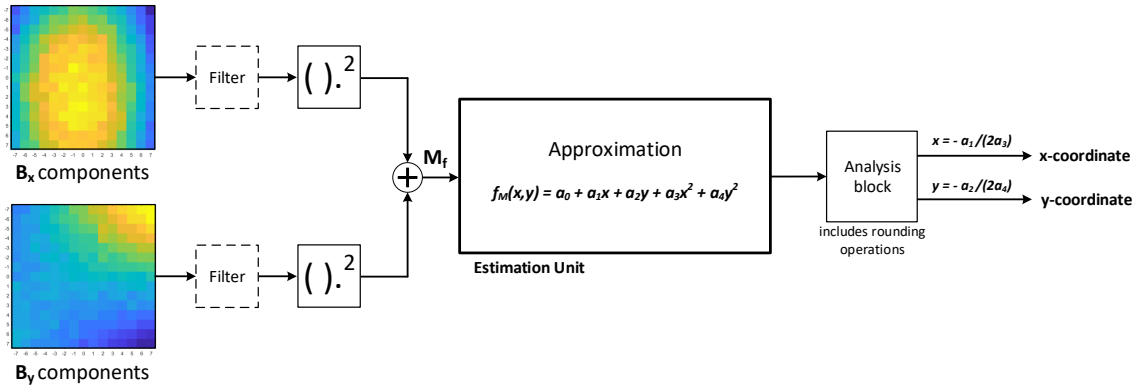


Figure 3.27: Signal flow diagram of simplified approximation and analytical based tracking algorithm.

Of course, the missed degree of freedom in form of the xy-dependency will affect the estimation in a negative way so that a higher approximation error needs to be accepted but on the another side the simplicity of this approach make it a serious and fast candidate for the tracking algorithm.
To sum up, the advantages and disadvantages of this simplified method in comparison to the approximation method above are:

- **Advantages**
  - Simplicity: No reinsertion of sensor array coordinates is required. This leads to a significant reducing of the required computational effort of this method.
- **Disadvantages**
  - Higher approximation error: Due to the missed coefficient for xy-dependency.

3.7 Angular error correction by angular teach-in phase

For the development of angular error correction algorithms the use of knowledge about an ideal system behavior from the point of view for angle evaluation is made. The angular error correction methods described here provides another, spatial domain independent teach-in procedure.

For a perfect aligned and ideal, failure-free encoder magnet - sensor system it is expected that the components of a field vector $\vec{H}_{ext}$ depending on the angular position of the encoder magnet which are measured in the xy-plane of the sensor are:

$$\vec{H}_{ext} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a \cdot cos(\alpha) \\ a \cdot sin(\alpha) \end{bmatrix} \tag{3.47}$$

and correspond normalized to the ideal sine and cosine components of a unit circle depicted in Figure 3.28.

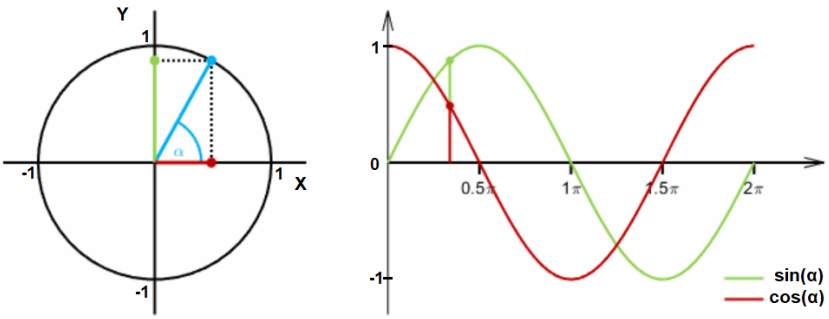


Figure 3.28: Sine and cosine components of a unit circle projected to the angular axis [11, modified].

However, under real conditions due to manufacturing related processes, mounting tolerances and not ideal arrangement of sensor and magnetic encoder the behavior of measured field components in the xy-plane can be represented by [81], [70]:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_x cos(\alpha) + a_k \sum_{k=2}^{n} cos(k\alpha + \Delta\phi_k) + x_{offset} \\ a_y sin(\alpha) + a_l \sum_{l=2}^{n} sin(l\alpha + \Delta\phi_l) + y_{offset} \end{bmatrix} \tag{3.48}$$

where the amplitudes a_x, a_y, a_k and a_l as well the offsets x_{offset} and y_{offset} are not equal. The terms in the summation represent all higher harmonics distortion frequencies included in the x- and y-signal.
So, in the presence of an encoder misalignment or other distortion effects the signals which are measured field components in the xy-plane contain not only the fundamental frequency, which represents the rotational frequency of the encoder magnet, but also an offset, different amplitudes and higher harmonics with various amplitudes and phase shifts [81], which leads to

a nonsinusoidal shape of the output signals of the sensor.
The errors can be summarized into the following three groups:

- **Sensor system side errors:**
  - Positioning deviation of bridge elements in the sensor.
  - Harmonics, due to the geometry of sensing elements (described in Section 2.2).
  - Die displacements on the lead frame.
- **Electronic side errors:**
  - Errors of analog preconditioning electronics, such as gains and offset deviations of OPs due to PVT[14].
  - Linear and nonlinear errors in analog-to-digital converters.
- **Application side errors:**
  - Encoder magnet misalignment effects.
  - Errors due to sensor mounting tolerances.

Not only due to the physical shape of sensing elements in the bridges of the sensor are responsible for the occurrence of higher harmonics in the system, as described in Section 2.2, but also in a non symmetrical sensor to encoder magnet arrangement and of course the hysteresis [57, Chap. 7].
The developed algorithm for angular error correction are applicable for correction of the effects caused by mentioned errors above. The idea of angular error correction algorithms is to match the behavior of sampled x- and y-signals with the ideal sine and cosine wave, i.e to descew the distorted cirle which results from xy-path of the sensor signals x and y over the 360° encoder magnet rotation in such a way that its shape matches to the unit cirle as much as possible and therefore comes closely to the ideal behavior of a perfect aligned encoder magnet - sensor system. Therefor suitable functions which act as rectification functions shall be fitted by optimization procedure during the teach-in phase and used afterwards constantly while the sensor is operating which compensate all the distortion components given in Equation 3.48. The aim here is to determine the coefficients of chosen functions by the use of linear regression approach, where e.g. the least squares method for the estimation of the coefficient of the rectification functions is applied.

The application of angular error correction algorithm by an angular teach-in procedure brings the followings advantages and disadvantages:

- **Advantage 1:** Universal approach. The methods described here are applicable also for one sensor systems. That means that by this method also all sensor array elements can be calibrated individually.
- **Advantage 2:** No reduction of „effective sensor area“. With respect to the sensor array all sensor elements can be still included for angle information extraction.

[14] Process-Voltage-Temperature variations.

- **Advantage 3:** Theoretically a significant angular accuracy improvement is obtainable, if the path resulting from the x- and y-signal is connected and do exhibits any crossings. This is of course depending on the applied correction method.
- **Disadvantage:** Processing delay. The required calculation for the angle corrections has to be included into the total processing chain.

Two approaches were developed in the context of this work: A linear angular correction algorithm based on homography, a projective transformation expressed in homogeneous coordinates, and a nonlinear error correction algorithm based on geometric distortion compensation. Such methods are used in the image processing for camera calibrations and image transformations [35], [88].

3.7.1 Linear angular error correction algorithm based on homography

The projective transformation called homography is a method which comes from the computer vision as well as computer graphics and is used for manipulation, warping and correction of images [45]. Also, used for mapping multiple pictures together, e.g. for panorama or image-scene creations [82]. This is achieved by creation of a projective space by use of homogeneous coordinates [77]. Projective homography is a line preserving (collinear) planar projection of an image from one to another xy-plane within a space of higher dimension and it is always a $P^2 \rightarrow P^2$ mapping, which is invertible [35]. So in general, it allows to manipulate n-dimensional vectors using a n+1 dimensional space. The manipulation is achieved by multiplication of input image coordinates by the homography matrix **H**, which exhibits different degrees of freedom.

Method explanation

The coordinates of a pixel (2D point) in an image denoted as a pair of values in the vector $\mathbf{x} = (x, y) \in R^2$

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} \tag{3.49}$$

can be represented using homogeneous coordinates $\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y}, \tilde{w}) \in P^2$. P^2 is called the two dimensional projective space and is $P^2 = R^3 - (0, 0, 0)$. In this space the vectors are considered to be equivalent which differ only by scale [88]. The conversion of a homogeneous vector $\tilde{\mathbf{x}}$ back to the inhomogeneous **x** is given by:

$$\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y}, \tilde{w}) = \tilde{w}(x, y, 1) = \tilde{w}\overline{\mathbf{x}} \tag{3.50}$$

where the vector $\overline{\mathbf{x}}$ is called augmented vector [88].

The simplest example for an expression of a geometric transformation by application of homo-

geneous coordinate is e.g. a translation operation which looks in the matrix notation as

$$\mathbf{x}' = \mathbf{H}\mathbf{x} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{3.51}$$

where $\mathbf{I}$ is the identity matrix, t_x and t_y are the translation parameters included in vector $\mathbf{t}$, and $\mathbf{0}$ is the zero vector. The x and y are coordinates of a pixel in the origin image and x' and y' the transformed coordinates of the output image, represented here in 3D vectors $\mathbf{x}'$ and $\mathbf{x}$.
The line preserving projection of homography can be easily comprehended by consideration of a 3D coordinate space where the input and output images are represented as two arbitrary planes in it, illustrated in Figure 3.29. The point in the origin plane left is projected onto the other plane right. The points on the planes are connected by a line from the coordinate origin.

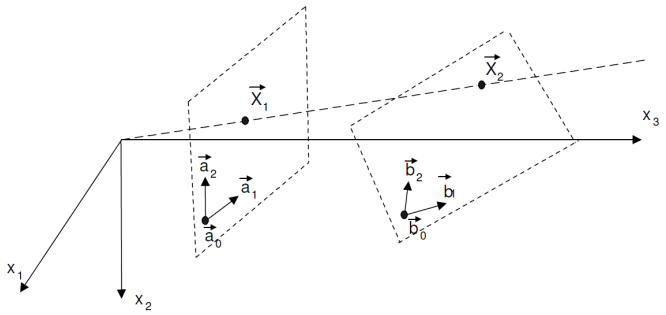


Figure 3.29: Line preserving projection of information contents of two arbitrary planes in a 3D coordinate frame [82, see definition]. The content included in the pixel denoted as $\vec{x_1}$ in the left plane is projected to the point $\vec{x_2}$ in the right plane.

In order to obtain a desired projection of the picture and the associated image manipulation a particular coordinates set must be chosen in the origin picture and defined as the expected coordinates in the output image (correspondences). Figure 3.30 depicts an example for this procedure.

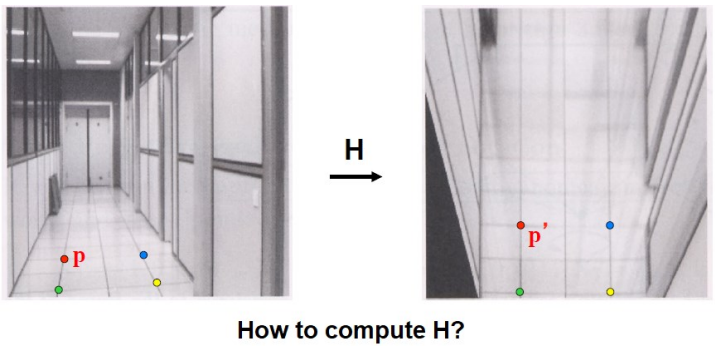


Figure 3.30: Chosen coordinates sets p and p' and obtained planar image projection [45, modified].

So, in order to get a homography of P^2 the following equations set of linear mapping is defined

$$\mathbf{p}' = \mathbf{H} \cdot \mathbf{p} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \mathbf{p} \tag{3.52}$$

where $\mathbf{p}$ is a 3D vector including the chosen x- and y-coordinate of the original image

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ w \end{bmatrix} \tag{3.53}$$

and $\mathbf{p}'$ is 3D vector including the chosen x'- and y'-coordinate of a point in the output image [45]

$$\mathbf{p}' = \begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} \tag{3.54}$$

The variable w in $\mathbf{p}$ is existent due to the formalism and is 1 for representation of points in homogeneous coordinates. Contained variable w' in $\mathbf{p}'$ is either 1 or $\neq 1$, depending on applied homography degree of freedom.

For the present problem, the samples of distorted sensor signals x and y are represented in the vector $\mathbf{p}$ and their correspondences x' and y', which represent the corrected signals, in the vector $\mathbf{p}'$ calculated by multiplication of the homography matrix $\mathbf{H}$ by $\mathbf{p}$. So, the matrix $\mathbf{H}$ acts as a signal correction matrix. The coefficients in this matrix are estimated during the teach-in phase. During the estimation of the $\mathbf{H}$ matrix the values in vector $\mathbf{p}'$ are the coordinates of the unit circle and values in vector $\mathbf{p}$ the sampled x- and y- distorted input signals which needs to match as good as possible with the corresponding values in $\mathbf{p}'$. The correspondence $\mathbf{p}$ to $\mathbf{p}'$ is denoted in the following as matching pair. The ideal coordinates x' and y' of a point $\mathbf{p}'$ on the unit circle are denoted as matching point what is an logical equivalence to the dataset $\mathbf{p} \rightarrow \mathbf{p}'$. The latter notation is used constantly during the investigations performed in Chapter 5.

There are in total 4 different degrees of freedom in homography existent [35]: 3, 4, 6 and 8, denoted in the following as DOF. Their geometrical transformation possibility is illustrated in Figure 3.31.

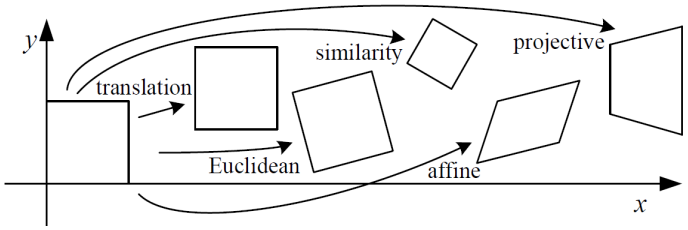


Figure 3.31: The set of possible 2D planar transformations [88].

The homography matrix of 3 DOF includes a translation vector $\mathbf{t} = [t_x \; t_y]^T$ and rotation matrix $\mathbf{R}(\theta)$. It is a isometric transformation. Only the offset and the initial angle (circle rotation) of x- and y- signals are correctable by these amount of DOF. Therefore, it is less interest for the present problem.
The homography matrix of 4 DOF the equitation set is [35]:

$$\mathbf{p}' = \mathbf{H}_{4DOF} \cdot \mathbf{p} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s \cdot cos(\theta) & -s \cdot sin(\theta) & t_x \\ s \cdot sin(\theta) & s \cdot cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s\mathbf{R}(\theta) & \mathbf{t} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3.55)$$

(1 scales, 1 rotations, 2 translations) and it is a similarity transformation. By use of this matrix the offsets, the initial angle and a common gain error of x- and y-signals can be corrected. However, common gain correction won't affect the angle calculation in the CORDIC and hence, 4 DOF are also of less interest.
For the case of 6 DOF the equation set is [35]:

$$\mathbf{p}' = \mathbf{H}_{6DOF} \cdot \mathbf{p} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3.56)$$

Here, a rotation and deformation of the image can be performed (2 scales, 2 rotations, 2 translations). This is an affine transformation. Two independent gain, offsets and initial angle error corrections are possible here.
The homography of 8 DOF the the equitation can be represented by [63], [35][15]:

$$\mathbf{p}' = \mathbf{H}_{8DOF} \cdot \mathbf{p} = \begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{v}^T & v \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3.57)$$

This is a projective transformation. It includes 2 scales, 2 rotations, 2 translations and 2 lines at infinity and acts non-homogeneous over the plane. It makes allowable to observe the points that vanish, e.g. striving to the horizon [35]. By use of this transformation not only the standard error mentioned above but also high order errors such as harmonics distortions contained in the x- and y- signal can be reduced. For this transformation the homogeneous coordinate changes in case if after the estimation process the variables in the last row of the $\mathbf{H}$ matrix $[h_{31} \; h_{32} \; h_{33}] = [\mathbf{v}^T \; v]$ becomes $\neq [0 \; 0 \; 1]$. As a result out of it, in order to get the coordinates x' and y' from the output image the homogeneous coordinate needs to be restored. The first two elements in vector $\mathbf{p}'$ needs to be divided through by the scaling factor $w = h_{31}x + h_{32}y + h_{33}$ in an extra subsequent step, which is in this case depending on

[15] Attention: Description of p' at 8 DOF is not standardized and various conventions exists. In [35] multiplication factor λ is used, in [45] only a w' and a constant w in p. Unfortunately, no suitable denotation could be found in [88]. For simplicity, the notation combined with convention from Larry Zitnick is used here [63]. The last element in the vector $\mathbf{p}$ is set to 1.

the input vector **p**. Due to the needed operation

$$\mathbf{p}'' = \mathbf{p}'/w = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \tag{3.58}$$

the transformation method becomes nonlinear. But, especially in the present case, the division by w is not required because it acts as a constant scaling factor for each component, which does not affecting the angle calculation in the arctangent function $\varphi = arctan(wy'/wx')$. Consequently, in case of DOF 8 the coefficients h_{31}, h_{32}, and h_{33} needs to be estimated but afterwards not considered in the operational phase. This represents an important advantage of the applicability of this method on the present specific problem.

Estimation formulation

The **H** matrix can be calculated directly by use of matching points whose amount starts with minimum required depending on DOF. In case of 6 coefficients of **H** matrix the amount of minimum required matching pairs counts to 3. For 8 coefficients 4 matching pairs are required at least. For the estimation of all h coefficients on the **H** matrix 4 matching pairs and an additional points set either of x to x' or y to y' is needed.
But in order to get a robust calculation of the **H** matrix coefficients it is necessary to consider much more samples of the signals than the minimum required[16]. so that the linear equation system becomes overdetermined. In this case, the **H** matrix is obtained by estimation procedure, formulated as a linear regression problem:

$$\mathbf{Ax} = \mathbf{b} \tag{3.59}$$

where vector **b** is consisting of a set of ideal coordinates $\mathbf{p}'$, so expected x-/y-pairs on the unit circle, vector **x** includes the coefficients of the **H** matrix and **A** represents the design matrix which is consisting of particular compositions of input values x and y as well as the desired values x' to y'. The solution for **b** is obtained either by solving of normal equations system or using methods such as QR decomposition or SVD algorithm, described more in detail in Appendix B.

The corrected signals , represented in vector $\mathbf{p}'$ in the Equation 3.52 can be written as follows:

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \quad \Rightarrow \quad (h_{31}x + h_{32}y + h_{33})x' = h_{11}x + h_{12}y + h_{13} \tag{3.60}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}} \quad \Rightarrow \quad (h_{31}x + h_{32}y + h_{33})y' = h_{21}x + h_{22}y + h_{23} \tag{3.61}$$

[16] It might also here happen that the formulated equation system becomes is not solvable due to not linear combination of equations set. Therefore, in such a case the solution is only obtainable by a estimation procedure as well.

The terms multiplied through by denominator $(h_{31}x + h_{32}y + h_{33})$ at the right side represents the basic terms needed for the design matrix construction. Depending on degree of freedom of homography matrix the Terms 3.60 and 3.61 are rearranged for x' and y' and afterwards constructed as $\mathbf{Ax} = \mathbf{b}$, arranged pair wise one by another in rows. In the following the notation $\mathbf{A}_i$ and $\mathbf{b}_i$ is used. $\mathbf{A}_i$ represents the submatrices consisting of two rows of the design matrix $\mathbf{A}$

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{a}_{xi}^T \\ \mathbf{a}_{yi}^T \end{bmatrix} \tag{3.62}$$

where $\mathbf{a}_{xi}$ and $\mathbf{b}_{xi}$ are the corresponding vectors placed under each other. In $\mathbf{b}_i$ the corresponding coordinates of the desired point are included:

$$\mathbf{p}_i' = \mathbf{b}_i = \begin{bmatrix} x_i' \\ y_i' \end{bmatrix} \tag{3.63}$$

The i is the index of the current correspondence, i.e the current matching pair, and l is the total amount of matching pairs.

6 DOF formulation:
For 6 degrees of freedom the denominator $(h_{31}x + h_{32}y + h_{33})$ becomes 1. In this case the submatrix $\mathbf{A}_i$ results to:

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{a}_{xi}^T \\ \mathbf{a}_{yi}^T \end{bmatrix} = \begin{bmatrix} -x_i & -y_i & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -x_i & -y_i & 1 \end{bmatrix} \tag{3.64}$$

The equitation system $\mathbf{Ax} = \mathbf{b}$ constructed of matching pairs of the amount n results to the following equations set:

$$\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_l \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \vdots \\ \mathbf{b}_l \end{bmatrix} \tag{3.65}$$

8 DOF formulation with 8 coefficients:
There are two methods for the determination of homography of 8 DOF. First implies the constraint, where the coefficient h_{33} is kept equal to 1 [37]. In this case 8 coefficients need to be estimated. This leads to the submatrix $\mathbf{A}_i$:

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{a}_{xi}^T \\ \mathbf{a}_{yi}^T \end{bmatrix} = \begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x_i x_i' & -y_i x_i' \\ 0 & 0 & 0 & x_i & y_i & 1 & -x_i y_i' & -y_i y_i' \end{bmatrix} \tag{3.66}$$

$\mathbf{Ax} = \mathbf{b}$ results in the following equations sets:

$$\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_l \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \vdots \\ \mathbf{b}_l \end{bmatrix} \tag{3.67}$$

8 DOF formulation with 9 coefficients:
For 8 degrees of freedom where all 9 coefficients need to be found the submatrix $\mathbf{A}_i$ is given by

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{a}_{xi}^T \\ \mathbf{a}_{yi}^T \end{bmatrix} = \begin{bmatrix} -x_i & -y_i & -1 & 0 & 0 & 0 & x_i x_i' & y_i x_i' & x_i' \\ 0 & 0 & 0 & -x_i & -y_i & -1 & x_i y_i' & y_i y_i' & y_i' \end{bmatrix} \tag{3.68}$$

$\mathbf{Ax} = \mathbf{b}$ results in the following equations sets:

$$\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_l \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = 0 \tag{3.69}$$

In case of estimation of 9 coefficients in the homography matrix, the linear system of equations becomes equal to zero. In this case the normal least squares algorithm cannot be applied here and the equations systems can only be solved by using SVD algorithm [62]. A short introduction and explanation about SVD algorithm can be found in [71]. To get deep understanding about SVD and the practical background and application in image processing of it is suggested to consider [31]. More explanations about formulation of design matrices for homography estimation is given in [37] and [62].

Algorithm

The teach-in procedure for the angular error correction was developed in a way so that the procedure is kept as simple as possible. This can be achieved by application of time controlled state machine while the encoder magnet rotates at a fixed rotation speed. The teach-in phase is illustrated in Figure 3.32.

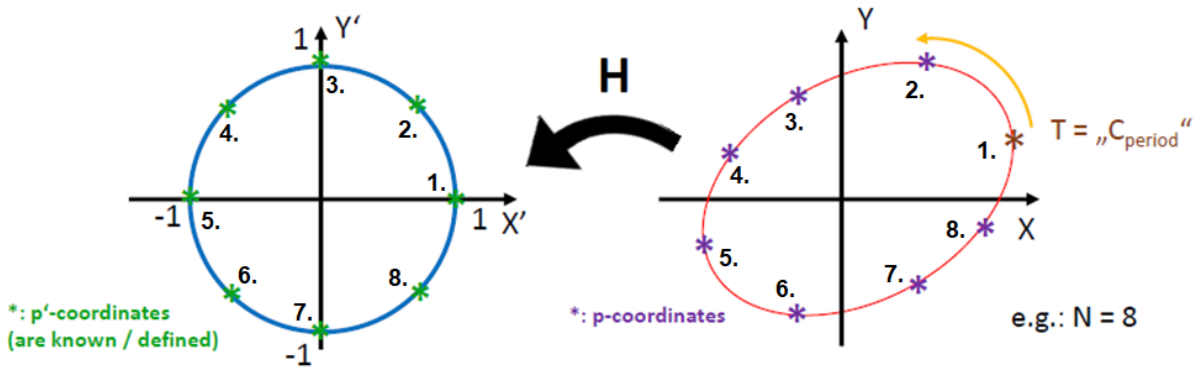


Figure 3.32: Teach-in procedure of the angular error correction algorithm. Left: xy-path resulting from sensor signals x and y of the ideal align and perfect system over the full encoder magnet rotation. The ideal coordinates of the points p' on the circle are the matching points, needed for the estimation. Right: xy-path resulting from distorted x- and y-signals of a sensor over the full encoder magnet rotation. The correspondent matching pairs are enumerated.

The matching procedure requires equidistant sampling of data points **p** with respect to the angular positions of the encoder magnet. Under the assumption that the sensor system is built in a system where a full 360° rotation of the encoder magnet is evaluated, the teach-in procedure can be easily realized by a time controlled procedure.

The teach-in procedure looks as follows:

0. Sensor is mounted into the application or calibration system at EOL and the encoder magnet is rotating with a constant speed $\omega = const.$
1. Activate the sampling of signals (e.g. operational phase timer interrupt for signal acquisition) by sending a flag from the ECU or μC to the sensor system.
2. Compare the sampled x value with the previous one and store each time the maximum value $x = x_{max}$. Perform this operation within a set time-out value $t_{out} > t_{period}$.
3. Continue to compare the values after the reaching of time-out value.
4. If the sampled value $x = x_{max}$ (or within its tolerance range): start a counter.
5. With a new sample value $x = x_{max}$ (or within its tolerance range): read the counter value.
6. Divide the counter value by the amount of set matching points N and set counter = 0.

7. Store the sampled x- and y- values at counter values [1,2, ..., N] $\cdot\, C_{peroid}/N$ in a data register.
8. Import the sampled values and the matching points into the estimation unit for determination of the **H** matrix $\Rightarrow$ h **parameters determined**.

So, after at least 3 full rotations of the encoder magnet the coefficients of the **H** matrix are estimated.
As the last step, if the proposed teach-in procedure is used, the angular position of the encoder magnet needs to be set and the initial angle φ_0 needs to be calculated once:

1. Drive the encoder magnet to its reference position and send a second flag to the sensor system if the encoder magnet reference position is reached.
2. Sample the values x and y by sending the second flag from the ECU or μC to the sensor system.
3. Calculate the corrected signals x' and y' using estimated **H** matrix
4. Calculate the initial angle φ_0 using CORDIC algorithm.
5. Store the initial angle φ_0.

After the calculation of the initial angle the system is ready to go into the operational phase, where the estimated **H** matrix is used and the initial angle φ_0 is subtracted constantly from the calculated one.
For summing up, the processing chain of the input signals in the operational phase is

$$\begin{bmatrix} x''(n) \\ y''(n) \\ 1 \end{bmatrix} = \begin{bmatrix} wx'(n) \\ wy'(n) \\ w \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x(n) \\ y(n) \\ 1 \end{bmatrix} \tag{3.70}$$

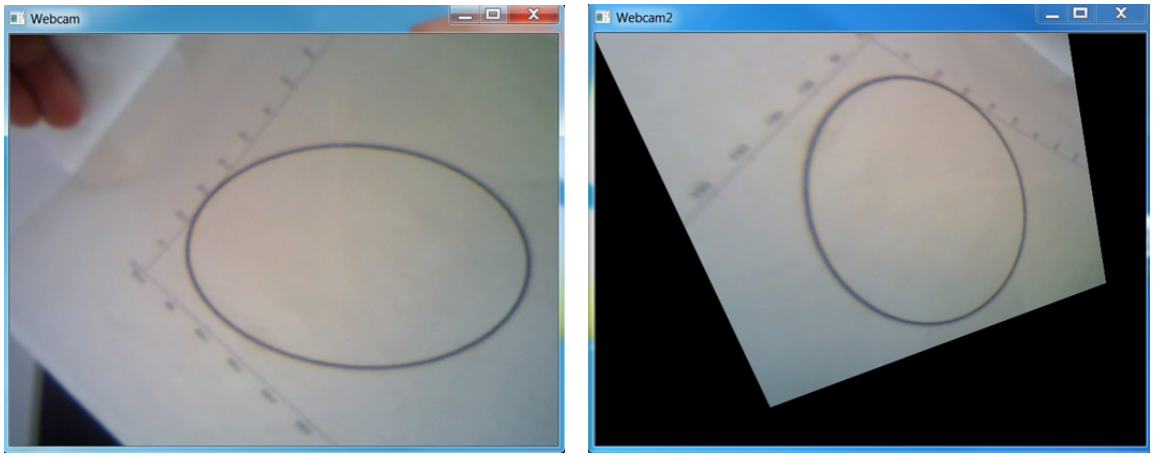
and for the angle calculation:

$$\varphi_{corrected}(n) = arctan\left(\frac{y''(n)}{x''(n)}\right) - \varphi_0 \tag{3.71}$$

where $x(n)$ and $y(n)$ are the input signals, $x'(n)$ and $y'(n)$ the corrected signals, φ_0 the initial and $\varphi_{corrected}(n)$ the corrected angle. Note that the calculation of w is not needed if only the information about the angle is of interest.

Alternatively or in case of non full rotation system, the teach-in procedure and the angle equidistant sampling of input signals can be performed each step controlled. In this case, no subtraction of initial angle in the operational phase but more interaction between the sensor and ECU will be needed, what leads to more complex communication between ECU and the device as well as more encoder magnet control, e.g. angle equidistant movements of the magnet and sampling the data at each position.

For demonstration of the method in Figure 3.33 a calibration experiment of a xy-path resulting from distorted x and y signals to the xy-path of the unit circle by using OpenCV library in python is shown.



(a) (b)

Figure 3.33: Homography camera experiment. Matching of x-/y-signal path to the unit circle by using OpenCV library in python. (a) Input frame: distorted x-/y-signal path. (b) Output frame: corrected x-/y-signal path. Correction matrix H = [0.33, -1.24, 536.4; 0.69, -0.09, -80.3; 0.0004, -0.001, 1]

Preconditioning recommendation

Estimation of parameters for projective transformations is often afflicted in practice with weak conditioning of regression matrices [37]. To get the best results from an estimation procedure it is recommended to follow the proposals given by R. Hartley and perform suitable preconditions of matrices values by adherence to the following rules [48]:

- always translate the center of mass to the origin of coordinate system
- performing of properly scaling so that the **average distance** of points from origin becomes $\sqrt{2}$
- independently normalizations to each point set (for multiple homographies)

3.7.2 Nonlinear angular error correction algorithm based on geometric distortion cancelation

This algorithm was developed with the goal to get the highest angular error correction possibility. The basis of this algorithm is the correction of geometric distortion, also denoted as optical aberrations that results from not ideal optical systems in image acquisition devices. These are the lens distortions from category of monochromatic aberrations [39], [Chap. 2][88]. A similar effect to such a distortion was observed within the context of pre-test of developed linear angular error correction algorithm as a remained error after its application. Therefore, this nonlinear additional algorithm was developed. This algorithm can applied in the total processing chain either as a subsequent correction after the linear correction via homography or also directly without the linear angular error correction.
Geometric distortion occurs in case of magnification which is location-dependent. The parallel and perpendicular lines in the origin image gets bowled and its elements are of focus in the border area. A so-called radial distortion is given if the strength of the aberration depends exclusively on the distance of the object point to the coordinate origin of the optical axes [39]. The radial distortion and its correction of an example picture if shown in Figure 3.34.

Figure 3.34: Two geometric distortion types and their correction [30]. Left: pincushion distortion. Center: corrected distortion. Right: barrel distortion.

To correct the radial distortion a geometric transformation of usually low order polynomials is applied, which are depending on the radial distance r^2 from the distortion center and are valid for the whole image area [Chap. 2][88], [39], [89]. Also combined models are applied, e.g Brown-Conrady model, which are able to correct not only a radial but also a tangential distortion [94].

The mathematical formalism in the developed algorithm is similar but also differs in its realization from the considered. The developed approach implies a split of the image plane, each for the x- and y-component in two parts. So, in the developed approach four 2D polynomial functions in total are used, which creates correction areas denoted as rectification planes. This

allows to correct not only the geometric distortion errors described above but also to get translations, shearings, warpings of the planes. It represents a mixture correction approach and gives a possibility to achieve a much higher angular error correction. But on the other side, this approach requires a controlling procedure while the sensor operation and implies hereby a discontinuity weakness. Summarized, the application of this algorithm brings the following advantages and disadvantages:

- **Advantage 1:** The nonlinear 2D polynomial functions allows a possibility to bent the xy-plane where the sensor signals are present. As a result of it, it is expected that this method will exhibit a higher angular error correction capability than the considered projective transformation approach.
- **Advantage 2:** Due to the splitting of xy-plane into 4 parts the areas covered by the polynomials can be warped independently from each other. This splitting method enhances the correction possibility as well. Besides the radial distortion correction it allows to perform independent concave / convex operations, scalings and warpings in the areas which are covered from polynomials.
- **Disadvantage 1:** The main disadvantage of the developed algorithm are the discontinuities at the transitions between the used 2D polynomial functions. In the transitions areas an error may occur which will decrease the angular accuracy of the system.
- **Disadvantage 2:** Since in the polynomial functions the variable input values need to be multiplied by each other the processing delay a higher processing delay in comparison to the linear error correction algorithm will be caused.
- **Disadvantage 3:** Since the xy-plane is split, a case statement which goes through several conditions is needed during the operational phase and due to it a complex state machine.

Method explanation

As well as in the previous algorithm, the samples of distorted sensor signals x and y are represented as components of the vector **p** and their correspondences, i.e. the corrected signals x' and y', in vector **p**$'$. For the rectification functions from which the corrected x' and y' signals are calculated the following 2D polynomial basis function was chosen:

$$f(x, y) = p_0 + p_1 x + p_2 y + p_3 xy + \sum_{n=1}^{N} (p_{2n+2} x^{2n} + p_{2n+3} y^{2n}) \tag{3.72}$$

where x and y are the sensor signals, $N \geq 1$ represents the degree multiplied by two of the polynomial rectification functions and x' as well as y' the corrected sensor signals. The even exponents of the coefficients in the sum terms are needed to obtain the re-bending of the xy-path, offsets p_0 allows the translation of the coordinates, linear function terms $p_1 x + p_2 y$ the scalings and skews of the plane and $p_3 xy$ allows to describe a plane twist.

Figure 3.35 illustrates the distortion compensation procedure of the developed algorithm for an axis symmetrical error.

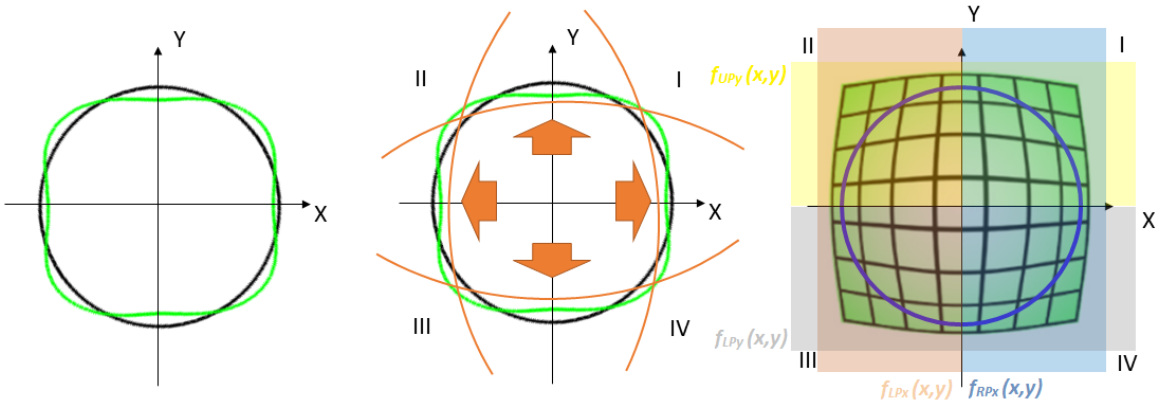


Figure 3.35: Functionality of geometric distortion cancelation algorithm. Left: observed remained error after homography correction of simulated data of a preliminary investigation based on real measurement data. Center: visualization of the distortion correction procedure. The lines represent profiles in the corresponding xy-plane obtained by 2D polynomial functions. Right: Perfectly corrected xy-plane by reverse buckling procedure (green barrel grid is taken from [14]). The colored areas represents the area covered by 2D polynomial functions in the operational phase. The transition between the usage of polynomial functions happens only for one function.

The algorithm requires a split of the operational xy-plane. The split of the plane is schematically illustrated in Figure 3.35(right), including the notation for the planes covered by rectification functions. Note that the alignment of the split in this illustration is valid only if a symmetrical error is existing, as shown in the figure.

For each component two rectification functions are required. The plane for the x-component is split to a left and a right rectification plane. The corresponding correction polynomials are denoted as

$$f_{RPx}(x, y) \tag{3.73}$$

for the right area of the x-coordinate and

$$f_{LPx}(x, y) \tag{3.74}$$

for the left area of the x-coordinate. The correction polynomials for the y-component are denoted as

$$f_{UPy}(x, y) \tag{3.75}$$

for the upper area of the y-coordinate and

$$f_{LPy}(x, y) \tag{3.76}$$

for the lower area of the y-coordinate.
However, in the real case it is not guaranteed that the required split of the operational xy-plane is symmetrical, especially if the algorithm is applied directly without a linear pre-correction. This is in Figure 3.36 the case. In the figure the resulting transition points and the sections of virtual planes needed to be created during the splitting of the operational plane for the teach-in and the operational phase are marked.

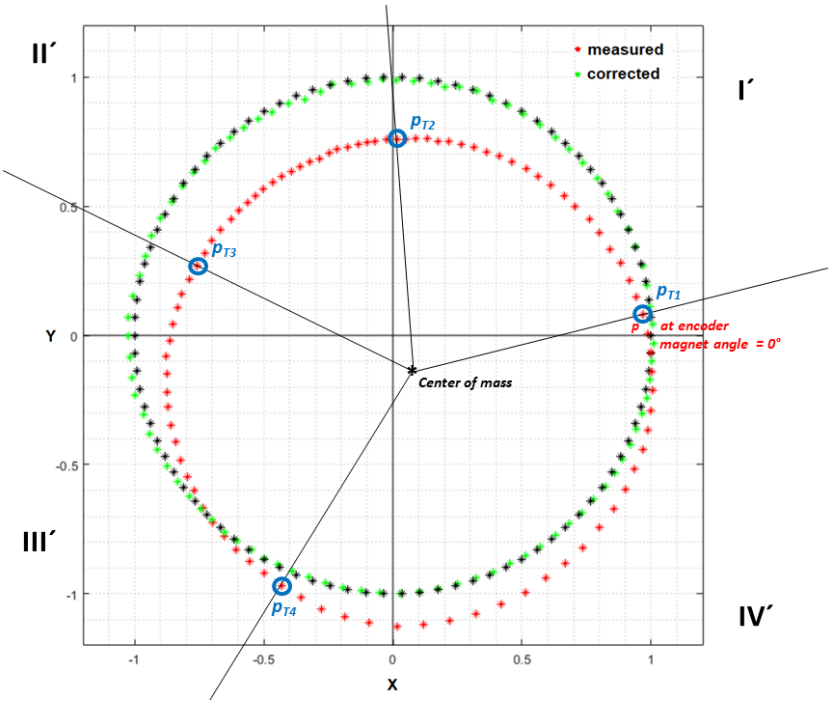


Figure 3.36: Split of the operational xy-area required by nonlinear angular error correction algorithm based on a measurement pre-test of the linear homography correction method. Black points: x- and y-coordinates of on the unit circle. Red points: sampled x-/y-signals of the KMZ60 sensor after offset compensation. Green points: Corrected x-/y-signals via 8 DOF homography. The key message here is the violation of the equidistance in the xy-plane resulting by the encoder misalignment although the sampling of data was performed encoder magnet angle equidistantly. Therefore, it is strictly required to determine the corresponding points of the distorted x- and y-signal samples $\mathbf{p_{T1}}$ to $\mathbf{p_{T4}}$ first and match the content included in it with the points from the ideal of the unit circle in accordance to the algorithm. Furthermore it illustrates the weakness of the splitting of the area in general. The case statement of the algorithm could fail if 3 of the 4 points comes relative close to each other. However it is assumed that this won't be the case in the practice. *Measurement setup: 180° encoder magnet rotation (AMR is double periodic). Magnet field strength above minimum saturation limit of 25kA/m. Extremely large x- and y-translation misalignment, approximately - 8 mm in y- and 16 mm in x-direction.*

The operational xy-plane needs to be split into 4 equivalent parts. Each part covers the system behavior represented in corresponding quadrant of the plane.

For the split of the operational area 4 points are required, denoted as:

$$\mathbf{p_{T1}} = [x_{T1} \quad y_{T1}]^T \tag{3.77}$$

$$\mathbf{p_{T2}} = [x_{T2} \quad y_{T2}]^T \tag{3.78}$$

$$\mathbf{p_{T3}} = [x_{T3} \quad y_{T3}]^T \tag{3.79}$$

$$\mathbf{p_{T4}} = [x_{T4} \quad y_{T4}]^T \tag{3.80}$$

These points are obtained by equidistant encoder magnet angular position sampling of the x- and y- sensor signal. The points between $\mathbf{p_{T1}}$ and $\mathbf{p_{T2}}$ covered in the warped quadrant $\mathbf{I}'$ has to be matched with the unit circle points in the first quadrant, points between $\mathbf{p_{T2}}$ and $\mathbf{p_{T3}}$ contained in the second distorted quadrant $\mathbf{II}'$ with the points in the second quadrant and so on. The coordinates contained in these points need to be stored for the teach-in phase and used during the operational phase for the automatic signaling of transitions between the used rectification functions $f_{RPx}(x, y)$, $f_{LPx}(x, y)$, $f_{UPy}(x, y)$ and $f_{LPy}(x, y)$.

Algorithm

The sampling and estimation procedure during the teach-in phase has to be performed as follows:

1. Sample the $x(n)$ and $y(n)$ signals angle equidistantly, either by use of a counter and time controlled state machine as proposed in Section 3.7.1 or manually by encoder magnet <- ECU / μC <-> sensor system interaction and store the values.

2. Extract and store the quadrant transition points $\mathbf{p_{T1}}$, $\mathbf{p_{T2}}$, $\mathbf{p_{T3}}$, $\mathbf{p_{T4}}$ from the sampled $x(n)$ and $y(n)$ sequences.

3. Split each of the $x(n)$ and $y(n)$ samples sequences into 2 parts: $x(n)$ sequence depending on x_{PT2} and x_{PT4} and the $y(n)$ sequence depending on y_{PT1} and y_{PT3}.

4. Prepare the sampled data by its concatenation to the sequences:
   For *Seq. 1 distorted* the $x(n)$ values covered from x_{PT4} to x_{PT2} and the $y(n)$ values covered from x_{PT4} to x_{PT2}, for the right side of the plane.
   For *Seq. 2 distorted* the remained $x(n)$ values from x_{PT2} to x_{PT4} and $y(n)$ values from x_{PT2} to x_{PT4}, for the left side of the plane.
   For *Seq. 3 distorted* the $y(n)$ values covered from x_{PT1} to x_{PT3} and $x(n)$ values covered from x_{PT1} to x_{PT3}, for the upper side of the plane.
   For *Seq. 4 distorted* the remained $y(n)$ values from x_{PT3} to x_{PT1} and $x(n)$ values from x_{PT3} to x_{PT1}, for the lower side of the plane.

5. Use the corresponding ideal sine and cosine values sequences:
   Seq. 1 ideal from $[cos(-\pi/2), cos(\pi/2))$ for the x-components on the right side of the plane.
   Seq. 2 ideal from $[cos(\pi/2), cos(3/2\pi))$ for the x-components on the left side of the plane.
   Seq. 3 ideal from $[sin(0), sin(\pi))$ for the y-components on the upper side of the plane.
   Seq. 4 ideal from $[sin(\pi), sin(2\pi))$ for the y-components on the upper side of the plane.

6. Perform the estimation of the coefficients of the 2D polynomial rectification functions by matching the following sequences:
   Seq. 1 ideal with *Seq. 1 distorted* for estimation of the coefficients of f_{RPx}
   Seq. 2 ideal with *Seq. 2 distorted* for estimation of the coefficients of f_{LPx}
   Seq. 3 ideal with *Seq. 3 distorted* for estimation of the coefficients of f_{UPy}
   Seq. 4 ideal with *Seq. 4 distorted* for estimation of the coefficients of f_{LPy}

In the operational phase the corrected signals $x'(n)$ and $y'(n)$ are obtained by the required case statement depending on the stored transition points p_{T1}, p_{T2}, p_{T3} and p_{T4}. By consideration of the results shown in Figure 3.35 it becomes clear that the multiple assignments are needed. Unambiguous assignments if the sampled x- or y- signal in the point **p** is behind or greater than both correspondent components of its transition points, or joint assignments depending on both coordinates for each x- or y-components, what makes the case statement procedure slightly complex. The formulated conditions are as follows:

$$x'(n) = \begin{cases} f_{RPx}(x(n), y(n)) & if \quad x(n) \geq x_{PT4} \quad and \quad x(n) \geq x_{PT2} \\ & else \quad if \quad x_{T2} > x(n) \geq x_{T4} \quad and \quad y(n) \leq y_{T1} \\ & else \quad if \quad x_{T2} < x(n) < x_{T4} \quad and \quad y(n) > y_{T1} \\ f_{LPx}(x(n), y(n)) & otherwise. \end{cases} \tag{3.81}$$

$$y'(n) = \begin{cases} f_{UPy}(x(n), y(n)) & if \quad y(n) \geq y_{PT3} \quad and \quad y(n) \geq y_{PT1} \\ & else \quad if \quad y_{T3} > y(n) \geq y_{T1} \quad and \quad x(n) \geq x_{T2} \\ & else \quad if \quad y_{T3} < y(n) < y_{T1} \quad and \quad x(n) < x_{T2} \\ f_{LPy}(x(n), y(n)) & otherwise. \end{cases} \tag{3.82}$$

The described case statement procedure is valid only under the assumption if the x-/y-signal path do not exhibits a initial rotation > 45° and the coordinates of transition points vary only marginally with respect to their opposite neighbors. To decrease the weakness of this method, additional points could be considered for the quadrant assignments.
Alternatively, instead of using the coordinates containing in $\mathbf{p_{T1}}$ to $\mathbf{p_{T4}}$ their corresponding angles φ_{T1} to φ_{T4} could be used for the transitions between the rectification polynomials. However, for this transition an additional cancelation of the offsets of x and y in a pre-correction step is required, but in this way the unambiguous assignments will be obtained completely.

Estimation formulation

The linear equation set formulated as a linear regression problem

$$\mathbf{Ax} = \mathbf{b} \tag{3.83}$$

is constructed as follows:
Vector **b** is consisting of a set of ideal values of the corresponding half of the sine or cosine component of the unit circle. Vector **x** includes the coefficients of 2D polynomial correction

function. Matrix **A** represents the design matrix which is consisting of compositions of input values x and y in accordance to the sequence of the chosen polynomial constructed as vector $\mathbf{a}_i$:

$$\mathbf{a}_i = [1 \quad x_i \quad y_i \quad x_i y_i \quad x_i^{2n} \quad y_i^{2n} \quad \cdots \quad x_i^{2N} \quad y_i^{2N}]^T \tag{3.84}$$

where index i is the index of the current correspondence between $\mathbf{a}_i$ and ideal value in b_i, and is $[1, \ldots, I]$. Index n is the index representing the polynomial degree according to the Equation 3.72. After the construction of vectors $\mathbf{a}_i$ the linear equation set $\mathbf{Ax} = \mathbf{b}$, which needs to be solved by an optimization method in the final step, results in:

$$\begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \mathbf{a}_3^T \\ \vdots \\ \mathbf{a}_I^T \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ \vdots \\ p_{2N+2} \\ p_{2N+3} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_I \end{bmatrix} \tag{3.85}$$

where I is the total amount of the row vectors $\mathbf{a}_i$ contained the sampled values as well as matching points of sine or cosine halves. N is equal to the maximal degree of the applied 2D polynomial rectification functions. During the teach-in procedure four of such linear equation constructions for the determination of coefficients of the 2D polynomial functions $f_{RPx}(x, y)$, $f_{LPx}(x, y)$, $f_{UPy}(x, y)$ and $f_{LPy}(x, y)$ need to be solved.

3.8 Stray field detection

As discussed in Section 2.7 it would be reasonable to operate on gradient field and evaluate their behavior for angle calculation only in case is a stray field is existent. For this purpose the algorithm described in this section is applicable.
Here, as well as for angular error correction algorithms the knowledge of ideal system behavior is used, described in the previous Section 3.7. In case of linear, cross field independent sensor array system, the interference of a homogeneous stray field component can be simply detected by summation of all corresponding $\mathbf{B_x}$ and $\mathbf{B_y}$ components measured by the sensor array which sums represent the sine and cosine component of the external field $\vec{H}$, calculation of magnitude and comparison of its value with the previous computed magnitude value. The interference with a stray field is detected as occurred, if the magnitude value is not the same as the previous one. Figure 3.37(b) illustrates the used property.

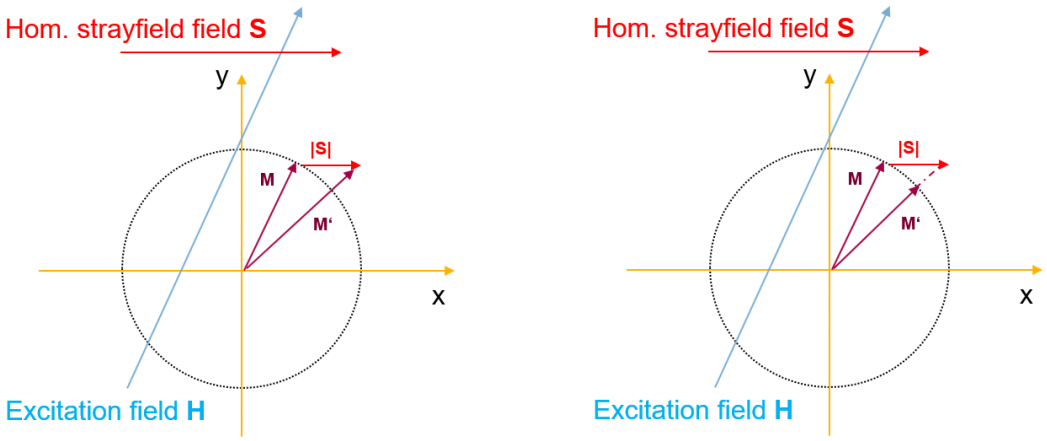


Figure 3.37: Schematically representation of stray field component interaction during the sensor operation of a saturated and non-saturated system. (a) x-/y-signal path of non-saturated sensor system: Amplitude and angle information are available. In presence of stray field magnitude M is changing, M $\neq$ M'. (b) x-/y-signal path of saturated sensor system. Only angle information is available. In presence of stray field S the magnitude M stay always equal M' (or can be only smaller of if the stray field vectors in the completely opposite direction).

Algorithm:

1. Sample the components $\mathbf{B_x}$ and $\mathbf{B_y}$.
2. Accumulate the elements contained in $\mathbf{B_x}$ to $B_{x\,sum}$ and $\mathbf{B_y}$ to $B_{y\,sum}$
3. Calculate the magnitude value $M = B^2_{x\,sum} + B^2_{y\,sum}$.
4. Store the magnitude value M in a data register.
5. Repeat the procedure 1 to 3 on anew sampling cycle.
6. Compare the actual magnitude value with the previously calculated one:
   if actual value > previous value or actual value < previous value $\Rightarrow$ **stray field detected**.

Since a comparison with physical units is for this purpose not required, neither a division by the amount of total sensor pixels after the summations in step 2 nor the square root operation for the magnitude calculation in step 3 needs to be performed. A suitable threshold value shall be added to the comparison step 6 in order to avoid the impact of mechanical tolerances and quantization on detection, so decrease slightly the detection sensitivity.
By application of this algorithm one cannot distinguish if the stray field is present in case of an existing misalignment. Therefore, the algorithm for stray field detection is only applicable if the encoder magnet aligned perfectly or its misalignment is negligible small.

3.9 Total algorithmic and mathematical chain

The description of the total chain for angle information extraction, algorithmic extensions as well as angular error correction is illustrated in Figure 3.38.
To summing up, the following two key properties laid a foundation for the developed algorithm:

- Knowledge about ideal system behavior in the form as an axis symmetrical equilibrium magnetic state of the field components with respect to its spatial representation.
- Knowledge about an ideal system behavior on the evaluation side for angle calculation based on trigonometry. The operation of a perfect aligned encoder magnet - sensor system is represented on evaluation side by a unit circle.

The mathematical chain after the teach-in phases needed to be calculated while the system is operating looks as follows:

1. Extraction of field curvature either from the whole sensor array or symmetrically around the AMM point by:

$$x(n) \Leftarrow \frac{\partial^2 B_x}{\partial x^2} + \frac{\partial^2 B_x}{\partial y^2} \tag{3.86}$$

$$y(n) \Leftarrow \frac{\partial^2 B_y}{\partial x^2} + \frac{\partial^2 B_y}{\partial y^2} \tag{3.87}$$

2. Propagation of the signals $x(n)$ and $y(n)$ which represents the curvatures extracted from $\mathbf{B_x}$ and $\mathbf{B_y}$ data arrays through the **H** matrix by:

$$\begin{bmatrix} x'(n) \\ y'(n) \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \end{bmatrix} \begin{bmatrix} x(n) \\ y(n) \\ 1 \end{bmatrix} \tag{3.88}$$

3. By choise (if needed), propagation of corrected signal $x'(n)$ and $y'(n)$ to the nonlinear correction stage

$$x''(n) = \begin{cases} f_{RPx}(x'(n), y'(n)) & or \quad f_{LPx}(x'(n), y'(n)) \end{cases} \tag{3.89}$$

$$y''(n) = \begin{cases} f_{UPy}(x'(n), y'(n)) & or \quad f_{LPy}(x'(n), y'(n)) \end{cases} \tag{3.90}$$

   depending on the defined cases. This correction can also be used directly without pre-correction step 2.

4. Angle calculation by use of arctangent function.

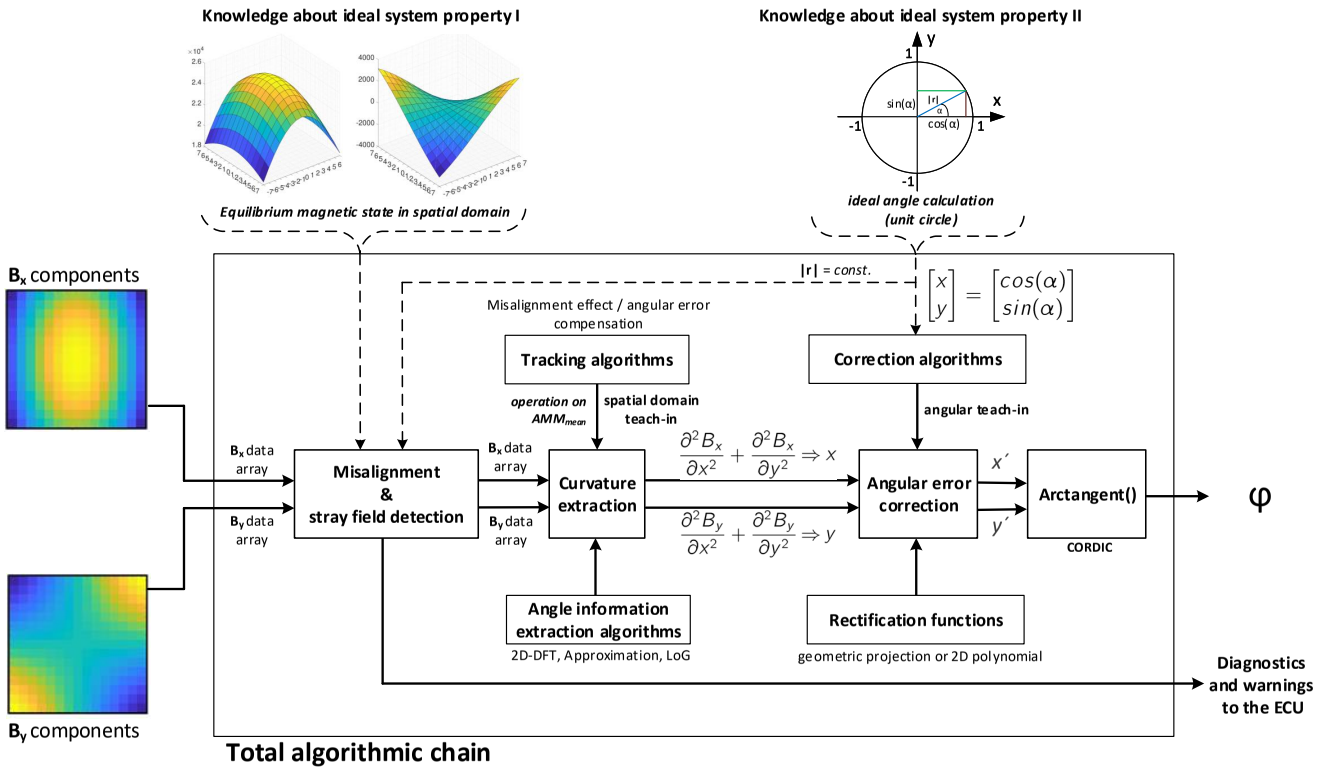


Figure 3.38: Signal flow chart of the whole algorithmic chain. Knowledge about ideal system properties is used in the developed concept completely, on the spatial domain and on the angle evaluation side. Misalignment and stray field detection during the operational phase can be achieved by pre-analysis of field components. Algorithms for Angle calculation can be coupled with tracking algorithm in order to compensate the misalignment effect on spatial domain. For this purpose a teach-in phase before the transition into the operation is necessary. To get the maximum possible angular precision a teach-in phase on the angle evaluation side can be applied, called as angular teach-in. Rectifications functions are used to match the x-/y-signals with ideal sine and cosine waves. Afterwards, the rectification functions are applied constantly during the operational phase.

4 Simulation and evaluation

This chapter consists of three parts. As the first, the simulation and its setup which was performed within the framework of investigation of algorithms is described. In the second part of this chapter the evaluation software created for modeling of digital acquisition and processing units on the ASIC, the analysis of system behavior as well as for tests of developed algorithms is described. In the last part the mathematical formalisms for evaluation of algorithmic results as well as rating of their accuracy and performance are documented.

4.1 Simulation setup

In order to obtain a knowledge about the practicability of the developed algorithmic concepts a 1:1 replicated real encoder magnet - sensor system was created in magnetic simulation. Realistic simulation scenarios were applied with the full correspondence to the specified operating area of the sensor. For this purpose the magnetic simulation software which was developed by H. Pape as part of the preliminary work in the ISAR project was used. The development of this simulator in documented in [75].

The functionality of magnetic simulator is based on the equation for the magnetic field of a magnetic dipole

$$\vec{B}(\vec{r}) = \frac{\mu_0}{4\pi} \frac{3\vec{r}(\vec{\mu} \cdot \vec{r}) - \vec{\mu}|\vec{r}|^2}{|\vec{r}|^5} \tag{4.1}$$

which is described in detail in Section 3.3.3.
In general, the following two degrees of freedom can be used in the simulation environment:

- Position of dipoles in three-dimensional Cartesian coordinate system
- Specification of the direction of magnetic moment $\vec{\mu}$ of dipoles

By placing of multiple dipoles close to each other a certain magnetic body can be constructed. Due to the superposition of dipole fields an approximation of the field which is generated from this body is obtained. By application of suitable geometric transformations operations the position and orientation of constructed magnetic body can be varied within the simulation area.
In the simulation, the central of mass of created magnetic body is located at the coordinate origin x = y = z = 0. The magnetic field which is generated by inserted magnetic body is calculated in the xy-plane of a particular length, width and at a set z-distance in relation to z = 0, specified

by the user. The xy-plane is represented by a grid of equidistant distributed points where the vector field which represents the magnetic data is extracted from, in the following denoted as meshgrid. All points in the meshgrid represents the sensor array elements, also denoted in this thesis as „sensor pixels“. Hence, from each sensor element the measured field components in x- and y-direction are obtained.

4.1.1 Encoder magnets

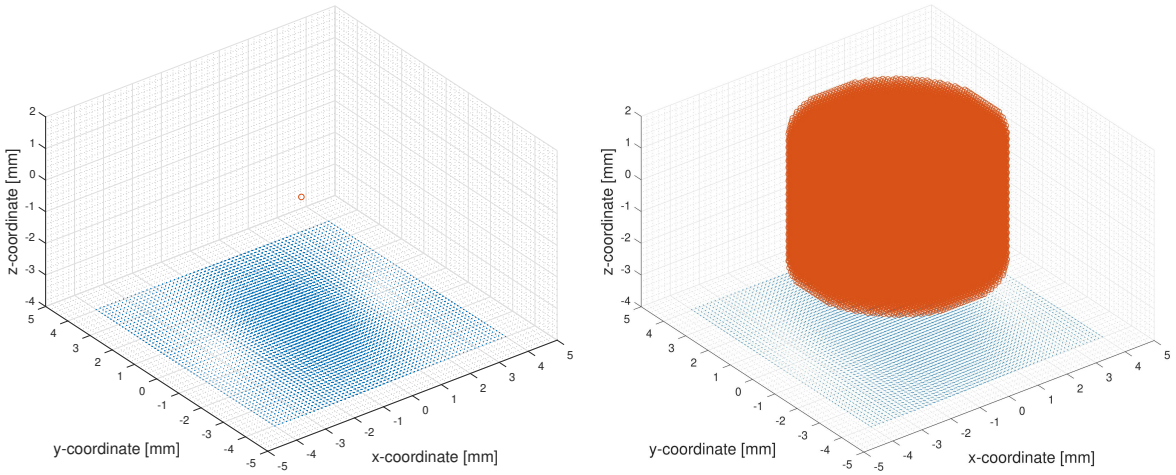
Within the context of this work a practical feasibility of the developed algorithms also for commercially used encoder systems shall be examined. Therefore, two magnet systems were built in the magnetic simulator: one dipole which represents the field generated by an ideal fictive sphere magnet and an in series present disc encoder magnet consisting of multiple dipoles within a cylindric body. The disc magnet was constructed with the smallest dimensions which is applied in series in some applications [68].
The dimensions and amount of dipoles of simulated encoder magnets are recorded in Table 4.1.

Encoder magnet	Length	Width	Height	Amount of dipols
Spherical magnet	4 mm	4 mm	4 mm	1
Disc magnet	6 mm	6 mm	4 mm	72,000

Table 4.1: Dimensions and amount of dipoles of simulated encoder magnets.

The fictive sphere magnet has the diameter of 4 mm. The amount of dipoles for the approximation of the disc magnet body was chosen in accordance to [75, P. 29] in order to keep the error in field magnet data sufficiently small[1]. The created encoder magnets and inserted xy-planes at the initial z-distance are illustrated in Figure 4.1.

[1] As a result of this setup the simulation for the generation of field data for the disc magnet within the framework set in 4.1.3 was relative time consuming. A simulation runtime of approximately of one week was necessary to generate the data with applied discretization of input parameters (specification used PC: 4x Intel Core i5-4690 CPU @ 3.5 GHz, 8 MB cache, 8 GB DDR3 RAM @ 1600 MHz).



(a) (b)

Figure 4.1: Created encoder magnets in the simulation and the xy-plane with sensor elements. (a) Infinitesimal small dipole which represents the ideal spherical shaped encoder magnet and the measured field components. (b) Created disc encoder magnet consisting of 72,000 dipoles. At each point within the meshgrid the superposition of field components generated by all inserted dipoles is calculated.

4.1.2 Coordinate system, units and initial values

For clarification reasons all the initial values, setup of coordinates and meshgrid units are documented in this section. All the definitions and specifications described in this section are equally valid for the fictional sphere magnet in the same matter.
Figure 4.2 illustrates the initial values with the correspondence to applied simulation setup and results evaluation.

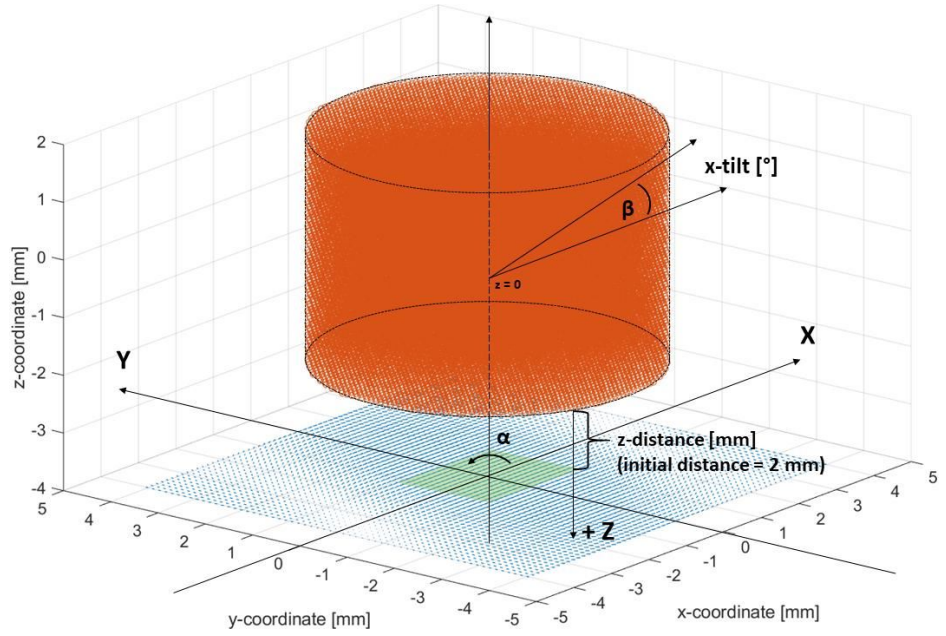


Figure 4.2: Coordinate system, units and initial values corresponding to the simulation setup and evaluation of results. Drawing for the disc magnet.

The initial orientation of the encoder magnet points to the x-direction. That means that the average vector of all field vectors present in the xy-plane is in this case:

$$\vec{H}_{average} = \begin{bmatrix} max \\ \\ 0 \end{bmatrix} \tag{4.2}$$

So, the initial angle at this orientation is zero. In the simulation the magnet rotates anti-clockwise.
Applied tilt variation described in Section 4.1.3 is related to the x-axis. The tilt axis applied in this setup is thus in parallel to the direction of magnetization of the encoder magnet (yaw axis). The location of the tilt-axis is the same as the standard x-axis. So, the central of mass of the encoder magnets is located exactly at the point where all axes are crossing each other. The notation *z-distance*, which is used in all following sections and chapters, is herewith defined as the distance from the central point on the lower surface of the magnet. According to the coordinate system used in the simulation this point has the coordinates (x = 0, y = 0, z = -2). The initial value for the z-distance is 2 mm. This is the closest physical distance to the encoder magnet which is adhered in the applications [68] and is that's why used as the initial z-distance value in this work.

Figure 4.3 illustrates the way how the simulation data of interest is extracted from the total simulated xy-plane.

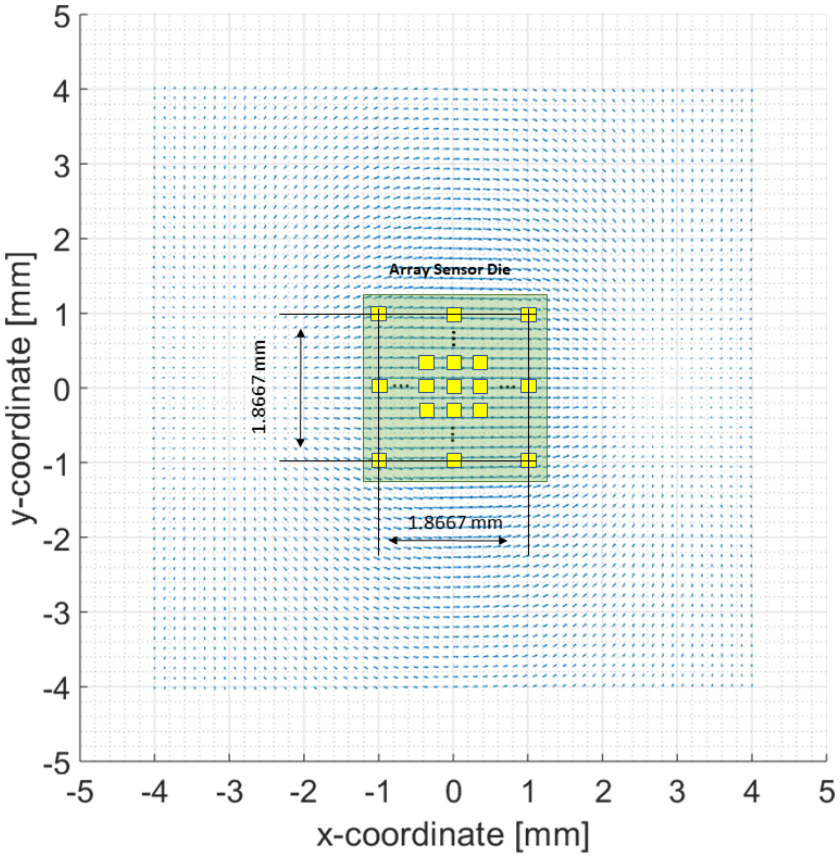


Figure 4.3: Simulated vector field in the created xy-plane and schematically illustrated 2 mm x 2 mm sensor array die inclusive placed sensor pixels on it. Real x- and y-distances between the edge sensor pixels are 1.8666 mm. The green area is movable within simulated plane limits. Amount of sensor pixels is variable. Amount of sensor points equidistant placed on the sensor die is also variable. This is achieved by interpolation, described in Section 4.2.4.

In the simulation an uneven meshgrid consisting of 61 x 61 elements was created in the xy-plane which has a physical size of 8 mm x 8 mm. According to it, the physical distance between the sensor points counts to 8 mm / (61 - 1) = $0.1\overline{3}$ mm. The used silicon sensor die area is 2 mm x 2 mm (see Table 2.3). Thus, by applied simulation setup a sensor array consisting of 15 x 15 elements is located within this physical sensor die area. By use of this setup the sensor die area is movable within the total simulated xy-plane. The movement step size corresponds to the physical distance between the sensor points, logically. In this way a mechanical misalignment in the form of x-/y-translation of the encoder magnet is equivalently represented without any need to perform additional simulations for this purpose.
The physical distance in between the first and the last sensor array columns or rows counts due to the applied setup to $1.8\overline{6}$ mm and is a little less than the maximum possible distance of 2 mm. However, a particular distance between the ends of the sensor die and xMR elements located at the boundaries needs to be maintained. Currently, a distance of approximately not less than 80 μm has to be maintained, which is almost covered by the applied discretization

size of the meshgrid created in the simulation. This is of course technology dependent and it is expected that by further miniaturization process in near future this boundary becomes smaller. Hence, the applied simulation setup was rated as suitable [68].

4.1.3 Simulation limits and parameter variation

Within the framework of investigations certain limitations for the simulation and results evaluation of algorithms applied on simulation data were set. These limitations are defined by the Safe Operating Area of the sensor system and are summarized in 2.6. According to it, an appropriate parameters variations space and their variational step size were defined. The whole parameter space is divided into the following groups:

- Array size and the corresponding extraction mask size for tracking algorithm extension
- Z-distance and mechanical misalignment parameters of the encoder magnet
- Signal quality parameters, also denoted as magnetic noise or magnetic interferences
- Electronic interferences (Gain mismatches and offset deviations of OPs)
- Parameters for data acquisition and arithmetic accuracy

In the following subsections the limits and variational step sizes for each parameter group are documented.

Array and extraction mask size

Based on to the theory described in Section 3.3 it is known that an uneven sensor array size of 3 x 3 elements is the minimum required one for the angle information extraction. As already mentioned in Section 4.1.2 the amount of sensor elements located within this physical sensor die area of 2 mm x 2 mm counts to 15 x 15. This represents the maximum sensor array size and it was chosen more or less pragmatically within the framework of the ISAR project in association with existing sensor hardware and possible interpolation techniques. In order to keep a central sensor element at the origin of coordinate system, what is more suitable for the arithmetic operations in case of field approximation, and be able to apply a symmetrical filtering in the spatial frequency domain, no even arrays were included within the framework of simulation and investigations. Therefore, the variational step size was set to 2 x 2 elements. The extraction mask size vary in the same step size within the limits of the array size, logically.

Z-distance and mechanical misalignment of encoder magnet

The limitations and variational step size of parameters for mechanical encoder magnet misalignment representation and the z-distance are listed in Table 4.2. The simulation was executed with a much wider parameter variation range as specified in 2.6 to be able to investigate

the system behavior and algorithms performance also outside the specifications, see values in brackets and * in the Table.

Parameter	Minimum	Maximum	Step size
Z-distance from the encoder magnet [mm]:	2	5 (10)	1
X-translation of encoder magnet [mm]:	-1 *	1 *	$1.\overline{3}$
Y-translation of encoder magnet [mm]:	-1 *	1 *	$1.\overline{3}$
X-tilt [°]:	0	5 (10)	0.5

Table 4.2: Limitations of simulation parameters and their variation step sizes: Z-distance and mechanical misalignment of encoder magnet. Values in brackets are the total limits of the simulation. *: the total limits for the x-/y-translation are limited by the total physical size of simulated xy-plane and counts to $\pm 3.8\overline{6}$ mm to an array size of 3 x 3 elements.

The limits for the z-distance and translation of encoder magnet in x- and y-direction corresponds to specified operation area in Table 2.2. Due to applied simulation setup described in the variational step size for representation of translation of encoder magnet in x- and y-direction is $1.\overline{3}$ mm. As already mentioned in section 4.1.2, applied tilt variation is related to the x-axis is in parallel to the direction of magnetization of the encoder magnet. According to it, this parameter is denoted in the following as x-tilt (yaw). Only one variation for the tilt was set within the parameter space since it is assumed that the tilt variation for the disc magnet will affect the system in similar matter. In case of a sphere magnet a tilt perpendicular to the magnetization axis (pitch axis) does not play any role at all. Also, the tilt of the sensor die in relation to the z-axis listed in Table 2.3 was neglected in the parameter space because its effect is represented in x-tilt variation. Radial shift could not be included into the parameter space due to functional limitation in the simulator.

Signal quality parameters

To model the variation of signal quality and interfering of signals by magnetic noise effects the interfering of the signals by the additive white Gaussian noise was applied within the simulation, where the input parameter is the signal-to-noise ratio measured in dB. The limits were chosen in the way that the SNR range of practical relevance as well as extrema are included into the parameter space. The variational step size of SNR was chosen as a trade-off between associated simulation time and visualization.

Parameter	Minimum	Maximum	Step size
Signal-to-noise ratio [dB]:	100	20	0.1

Table 4.3: Limitations of simulation parameters and their variation step sizes: SNR.

The setup for the application of white Gaussian noise interference is described in detail in Section 4.2.4.

Electronic interferences

Within the context of performed test only one real based case was applied for the examination of electronic interferences. Hence, no variations of the corresponding parameters were defined within this scope. The applied methods for the test are described in Section 4.2.4.

Parameters for data acquisition and arithmetic accuracy

In order to represent the impact of electronic data acquisition units on the angular accuracy an ideal analog-to-digital converter unit was created as well as methods for representation of limited binary arithmetics were applied in the software. The functionality of the methods is explained in detail in Sections 4.2.4 and in D.3.4. The set parameter variation space for data acquisition and arithmetical accuracy investigations is listed in Table 4.4.

Parameter	Minimum	Maximum	Step size
ADC resolution [bit]:	8	16	2
Arithmetic accuracy (s1Qn) [bit]:	12	22	1

Table 4.4: Limitations of simulation parameters and their variation step sizes: ADC resolution and arithmetic accuracy.

It is assumed that an ADC resolution below 8 bit won't be sufficient for aspired angular accuracy of $\varphi_{err} \leq 1^{\circ}$ due to a enormous quantization effects of the analog-to-digital converters at this resolution. The maximum value for the ADC resolution was set to 16 bit. This value was assumed in relation to the maximum possible acceptable data acquisition delay associated with sampling and data conversion time. The minimum value for arithmetic accuracy was set to 12 bit with regard to the hardware conditions in the project what is discussed in D.3.4. The maximum value for the numerical accuracy was set pertinently to the expected maximum applicable hardware limitations.

4.2 Evaluation software

Within the framework of development and testing of algorithms an evaluation software consisting of a graphical user interface as front-end and a centralized program with subordinates routines and functions as a back-end was constructed. The goal of software creation was a centralization of data control, to obtain a possibility for fast & direct comparison of all algorithmic results as well as a better overview in general. The GUI as well as the back-end was created in MATLAB® by using UI design environment called GUIDE (GUI development environment). The created evaluation software is fully compatible to applied magnetic simulator [75] so that the generated data can be directly accessed, processed and evaluated.
The created software is consisting of more than 60 functions in total. The complete software functionality can be divided into the following groups:

- Import of simulation data and handling with it.
- Filter masks creations.
- Application of interferences on „measured“ field data.
- „Digitalization“ procedures (sampling and conversion of data to fixed-point).
- Application of filtering procedures.
- Implementations of algorithms for angle information extraction.
- Implementation of tracking algorithms.
- Implementations of algorithms for angular error correction.
- Functions for mathematical formulations needed for applied estimations procedures.
- Evaluation of angular errors.
- Visualizations.

For the main tests executed in 5.7 additional script files were created, coupled on software data structure and its controlling procedures via flags etc.

4.2.1 Import, interference and acquisition of simulation data

To provide a clear understanding about the way how the data is interfered by electronic and magnetic disturbances a data flow illustration was created. Figure 4.4 illustrates the data flow from the import until the evaluation step.

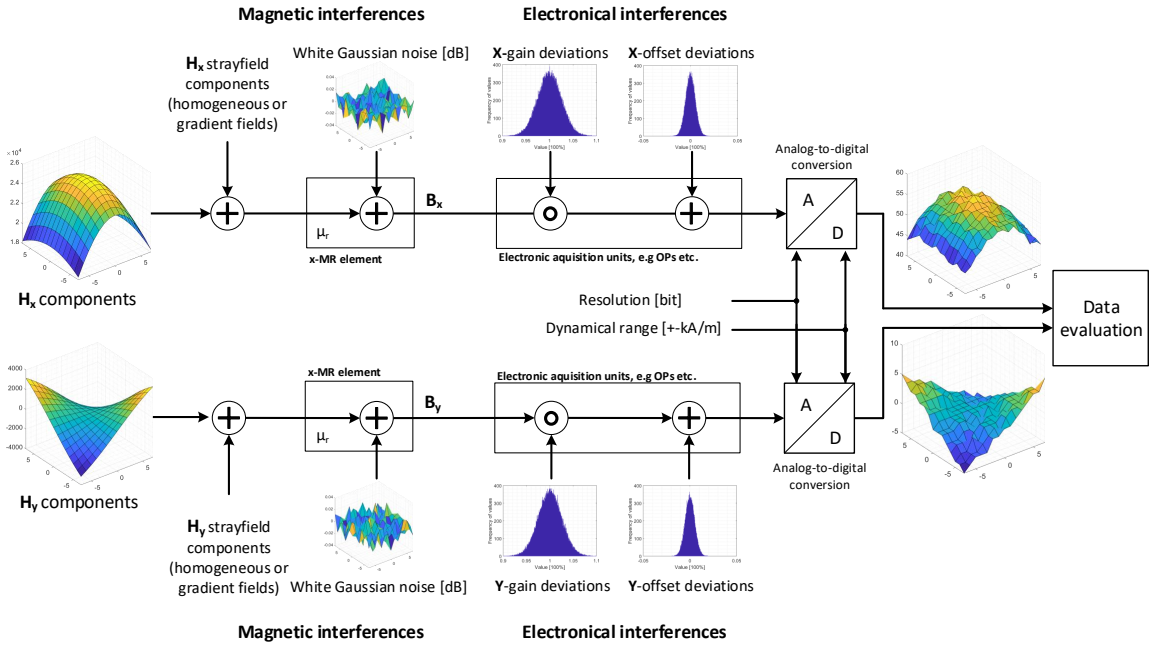


Figure 4.4: Illustration of data flow from the loading step, magnetic and electronic interferences until the data acquisition and evaluation step. The step where the magnetic data is interfered by white Gaussian noise can be considered as interaction of field data with xMR elements. Here additional function can be included in order to represent the hysteresis and non ideal behavior of the magnetic elements of the sensor. Afterwards, the impact of gain and offset deviations of the amplifiers are modeled. Finally, the data is sampled by created ADC functions.

In the beginning the simulated data is loaded into the program environment. The magnetic data is firstly interfered by a stray field and secondly by magnetic noise, modeled as white Gaussian noise. In this way, the quality of signals which are propagated to the analog electronic units can be affected. The interaction with magnetic material is occurring exactly at this step. Hence, after this step the magnetic components are considered as $\mathbf{B_x}$ and $\mathbf{B_y}$. The representation of the physical behavior of xMR-sensor elements is not covered in the simulation since the sensor physics is not focused in this thesis. Hence, it is assumed that all the sensors array elements exhibit the ideal magnetic behavior with neglected hysteresis, so an 1:1 transmission on response of magnetic body to excitation by external magnetic field $\vec{H}$, with the permeability parameter

$$\mu = \mu_0 \mu_r = 1 \tag{4.3}$$

Thus, the strength magnetic flux density $\vec{B}$ which is caused by excitation field interaction with magnetic materials is equal to the strength of magnetic field $\vec{H}$. Due to this, the effects of hysteresis and non linearities associated with it are not covered in this work, but exactly on this step the software can be extended by coupling of additional functions into the chain before the electronic interfering effects. At the end, the data is sampled by the ADC units and is

propagated further where a possible pre-processing and algorithms are applied on it. The sampling of field components by use of created ADC functions is of course done by choice.

4.2.2 Data and program flow

At first, with executing the file *ISARGUI.m* the graphical user interface is constructed which is described graphically in *ISARGUI.fig*-file and all important flags and pre-initializations of variables, filter creation as well as storage reservation, is done. Also the scaling of field strength of data is performed here once to obtain a field strength of 25 kA/m at the central array element at a z-distance from the encoder magnet of 2 mm. Scaling factors for both magnets are calculated here and used subsequently, each time if a new data is loaded. All the data are stored in a data struct called *handles*. This data struct must be passed to the functions as a transfer parameter in order to allow to access the data stored in it.
All objects included in the GUI are associated with corresponding callback-functions, where suitable handling of data and / or variables is performed, e.g. reallocation of the data-storage in case of variation of array size and / or extraction mask size, setting and re-setting of flags for algorithm extensions etc. These callback functions are executed if an event on one of GUI object is caused by user. The main part of the program represents the update-function *UPDATE_BUTTON_Callback* where the at first the chosen data is loaded and an area of interest is extracted of it, interferences magnetic and electronic interferences takes place, preprocessing as well as sampling of data are performed, algorithm for angle information and angular error correction are applied and the obtained results visualized. The functions for angle information extraction / calculation of corresponding x-/y-signal are self consistent what means that all statement and flag queries for e.g. extensions or are done internally.
After the processing and data modification the handles struct must by updated by executing the command *guidata(hObject,handles)*.
The functions are operating on the data separated as follows:

- *SOAR_DATA_string* - Data string created in dependence of variables for z-distance and x-tilt. Used for opening of data from magnetic simulator.
- *SIMDATA* - Storage for the total simulation data loaded (includes also encoder magnet information etc.).
- *MAGFIELD_BX_BUFFER / MAGFIELD_BY_BUFFER* - Loaded field data.
- *MAGFIELD_BX / MAGFIELD_BY* - Storage for field data after interpolation step.
- *MAGFIELD_BX_operated / MAGFIELD_BY_operated* - Main data for angle information extraction and for operation on it.
- *MAGFIELD_BX_operated_AMM / MAGFIELD_BY_operated_AMM* - Data for angle information extraction and operation on it in case of tracking algorithm extension.

For clarification of dependencies between the functions and in order to get a understanding about the total processing flow sequence a UML activity diagram of the main callback procedure in the function *UPDATE_BUTTON_Callback*, which represents the main program execution was

created. The UML diagram is included in Appendix D.2.5 in Figures D.3 and D.4 inclusive color coding and legend notation.

4.2.3 Algorithms implementation

The implemented algorithms for angle information extraction as well as the main update function which calls all implemented methods can be found in Appendix D.2. The meaning of transfer and output parameters of the functions can be taken from the comments in the codes. The functions with the algorithm implementations are divided into three groups and are in the following briefly introduced.

1. **Angle information extraction algorithms**
   The angle information extraction algorithms including stray field suppression are implemented in the following functions[2]:
   - *DFT_xy_calculation()* - Covers the 2D-DFT based algorithm. Extension for global gradient correction is coupled into this function directly and is implemented in function *gradient_correction()*. This functions uses a sub function *field_approximation_middle_shifted_glob_gradients_only()* for the estimation of coefficients of 2D linear equations.
   - *Approx_xy_calculation()* - Includes the approximation based algorithm. Uses the function *field_approximation_middle_shifted()* for the estimation of 2D polynomial coefficients.
   - *Flat_approx_xy_calculation()* - Includes the modified approximation based algorithm. The estimation of the basis function coefficients is done directly in this function. The function operates directly on the spatial area covered by the extraction mask.
   - *Convolution_xy_calculation()* - Covers the application of convolution with the LoG filter mask.
   - *LOG_xy_calculation()* - Covers the single LoG algorithm.
2. **Tracking algorithms**
   All the methods for the localization of the AMM points are implemented in the function *Track_AMM_point()*[3]. The selection of the tracking method is controlled by the GUI flag *handles.Tracking_method*.
3. **Angular error correction algorithms**
   The algorithm for angular error correction associated with the angular teach-in phase are implemented in the following functions[4]:

[2] See Appendix D.2.2.
[3] See Appendix D.2.3.
[4] See Appendix D.2.4.

- *Homography_correction()* - Function for application of linear angular error correction algorithm on the extracted x- and y-signals. Performs the estimation of **H** matrix coefficients once and corrects the signals by using the **H** matrix afterwards. The estimation of H matrix is performed in the function *homography()*.
- *nonlinear_correction_geometric_distortion()* - Function for application of nonlinear angular error correction algorithm on the extracted x- and y-signals. This functions splits the virtual operational area into 4 parts, executes the estimation of correction function coefficients, corrects the signal sequences and concatenates the corrected signal sequences together. The estimation of the 2D polynomial rectification functions is carried out in one of three implemented sub functions *polynomial_approximation_2D_2nd/4th/6th_ord()* (2nd for second, 4th for fourth and 6th for sixth polynomial degree).

4.2.4 Relevant functions for investigations

In the following, only the most important functions within the scope of performed investigation in Chapter 4 are explained.

Array size variation

In the context of investigations it is necessary to examine the effect on angular accuracy and stability enhancement by an increase of the total amount of sensor elements within the specified silicon die area. Based on simulation setup described in Section 4.1.2 the area within the simulated xy-plane where the field components are extracted from is $1.8\overline{6}$ x $1.8\overline{6}$ mm^2. On this area 15 x 15 equidistantly distributed „sensor pixels“ are located. Only an array of 3 x 3 elements can be directly created out of this sensor field. Hence, for creations of all other equidistantly placed arrays of sizes between 5 x 5 and 13 x 13 elements within the specified die area another suitable solution must be found.
To be able to vary the amount of sensor elements placed as an rectangular array in the form of a matrix on the specified die area the function *decimate_values()* was created, where interpolation technique is used. The function requires in addition to passing of standard handles data struct a transfer parameter called *Interpolation_steps_inbetween*, which represents the step size for the interpolation of field data.
On one hand one could simulate multiple xy-planes of various equidistantly distributed „sensor pixels“ in order to ensure a direct extraction of field data in equally distributed points according to the set array size. But this would require on the other hand more complicated procedures during data loading periods and in any way a violation of set x-/y-misalignment translation steps, if the array size needs to be varied in case of presence of a fixed x-/y-translation. Also, to following of this approach would bring several of GBs of additional data. Another solution for it would be a simulation of field data on the fly during the test of algorithms. However, this is adequate only for a sphere magnet but not for the generation of the disc magnet data. Due to this consideration the array size variation problem was solved by interpolation, where the

spline interpolation method was chosen. Of course, this is directly associated with an error, which occurs due to interpolation itself (which is assumed to be insignificant) and the error related to the equidistance of points where the data is extracted from, which is a much more concern issue. The equidistance problem is associated with the limitation of amount of interpolation points, that represent the resolution of interpolated array data. A *linspace()* command could be applied in this step, but in this case it would be necessary to round the sensor coordinates between the first to middle and middle to the last element, what cause a violation of the equidistance of points in total. In oder to avoid this problem, a special technique was applied in *decimate_values()* function, what is described in the following:

1. Creation of an equidistant meshgrid within the limits (-7,7) of a chosen step size between set limits (denoted as *Interpolation_steps_inbetween*).
2. Creation of two coordinates vectors for the middle element of the sensor array pointing to the last and to the first element of created meshgrid, with a step size which is calculated by: $\textit{step size} = \textit{rounding toward zero}\left(\frac{\textit{length of created mesh}}{\textit{desired array size} - 1}\right)$.
3. Flipping the first coordinate vector and subsequent concatenation of created vectors.
4. Performing of interpolation of components stored in $\mathbf{B_x}$ and $\mathbf{B_y}$ data arrays.
5. Extraction of values from interpolated field by use of concatenated coordinates vectors.

By application of this method a symmetric equidistance of points in relation to the central element within the sensor die area of $1.8\overline{6}$ x $1.8\overline{6}$ mm^2 is always guarantied. Only a physical array size deviation with relation to the specified absolute boundaries of the die area will occur, represented as remaining amount of interpolation steps which do not fit into the capturing area of coordinates vectors.
Within the framework of tests of algorithms in fact a small impact of physical array size deviation was observed since certain tendencies did not exhibit a behavior which was normally expected. However, this error was not observable by an decrease of interpolation steps factor to 0.01. Hence, it is strictly recommended to keep the decimation factor $\leq$ 0.01.

Analog-to-Digital Converter

The most important function for the applicability study and practical feasibility test of the algorithms is the created function *sample_components_signed* which represent the analog-to-digital converter unit. The function returns field components values represented as signed integer. The functionality of an ADC is covered in Appendix D.3.1. Applied adjustment of the ADC unit in the created software is explained in the following.
In the evaluation software none of the mentioned additional errors described in Appendix D.3.1 were modeled because of their manufacturer dependency. So, the created function represents an ideal ADC.

In accordance to the application specification the physical capture which represents the dynamic range of the ADC was set within the framework of the evaluation setup to $\pm 30 kA/m$, in oder to:

- Be able to resolve the maximum field range of $\pm$ 25 kA/m of the encoder magnet at minimum z-distance of 2 mm.
- Do not cause a climbing in case of presence of a stray field, which maximum field strength is specified up to 4 kA/m .
- Obtain an additional reserve of 1 kA/m, before the climbing limit of the ADC.

Figure 4.5 illustrates the obtained ADC transfer curve of a test case for the created function *sample_components_signed*. In the test case the ADC resolution was adjusted to 6 bits and its dynamic range to $\pm$ 30 kA/m. As a stimuli signal a constant field variation from -35 kA/m to 35 kA/m was used as parameter input value.

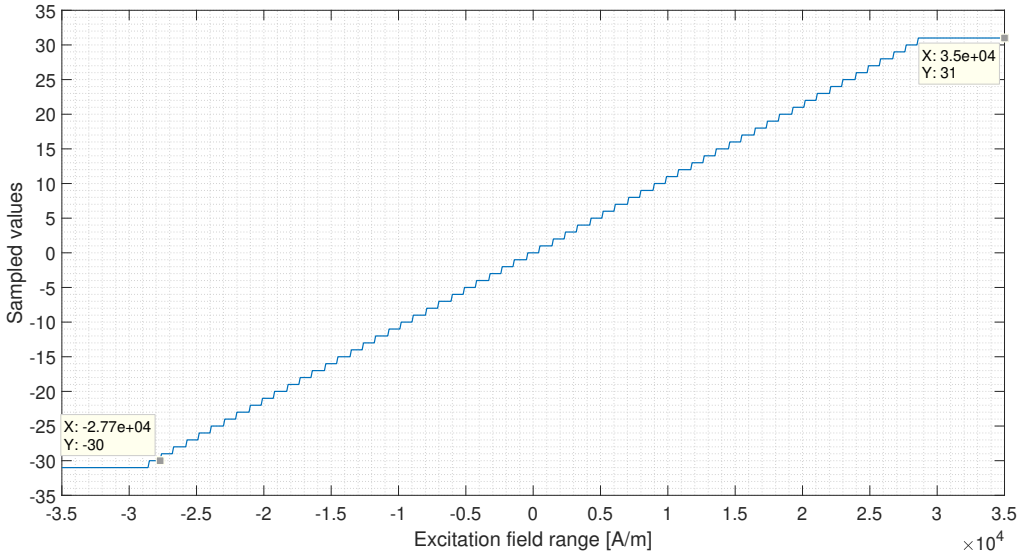


Figure 4.5: Proof of ADC function functionality *sample_components_signed*. Conversion of a constant field variation from -35 kA/m to 35 kA/m. Values outside the covered signal range by the ADC cause a climbing on the last possible digital value.

From Figure 4.5 the correct functionality of created function is evident. The conversion curve is shifted to LSB/2 and a climbing on the last digital value is caused if the input value is out of the specified dynamic range. Due to direct conversion to signed integer without the pre-conversion to binary representation in two's complement, the values margin of created ADC function amounts to $[-2^n/2 - 1, 2^n/2 - 1]$.

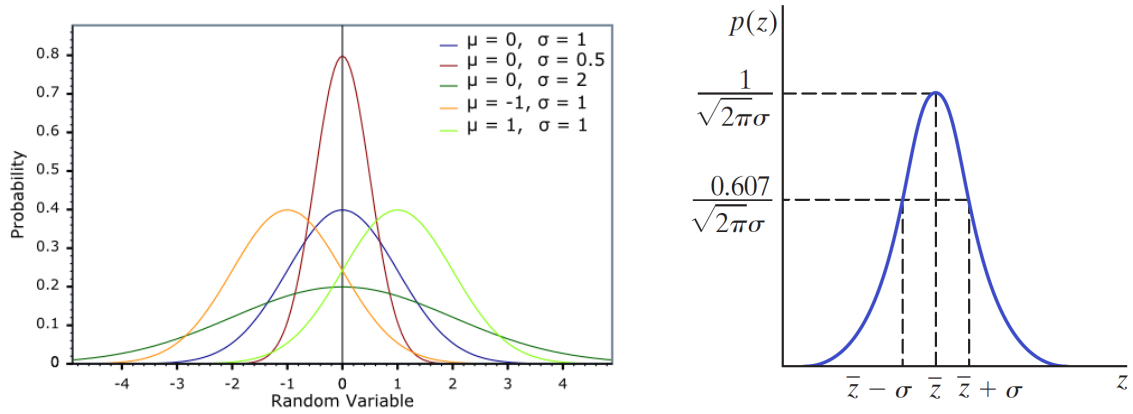
Signal-to-noise ratio

In reality the quality of measured physical signals is always interfered by noise. Its effect is extremely crucial. Especially for magnetic sensors, the noise interference is depending on manufacturing processes, material properties as well as the sensing technology. Last but not least, the electrical devices and power electronics next to the sensor environment cause electromagnetic interferences[5] and acts as distortion sources which cause magnetic noise as well. The most important noise kinds were considered within the framework of this thesis and are introduced in Appendix D.3.2. Since the simulation investigations serve for a general prove of developed methods, the application of the additive white Gaussian noise model is considered as the most suitable method for noise effects representation. Its usability has been proved in information technology and due to its mathematical tractability the Gaussian noise models are used frequently in practice [78]. In this type of noise, the issues associated with Johnson-Nyquist are covered well due to its identity to the spectral power density and the Gaussian distribution. Moreover it is assumed, that the main effect arising from electronic components in the ASIC as well as noise interference due to electromagnetic interferences of electronic devices can be modeled by an additive white Gaussian noise as well. The mathematical description for white Gaussian noise is its symmetric, bell-shaped probability density function.
The probability density function of a Gaussian random variable z, is defined by the following equation [78, extended]:

$$f(z \mid \overline{z}, \sigma^2) = p(z) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{(z-\overline{z})^2}{2\sigma^2}} \qquad -\infty \leq z \leq \infty \tag{4.4}$$

where z represents the intensity of the sampled gray value, $\overline{z}$ is the mean or average value of z, also denoted as μ, and σ its standard deviation. In the mathematics the Gaussian distribution is also called normal distribution. The variance what represents the „measure“ of dispersion is σ^2. If a random variable with a Gaussian distribution exhibits a mean value of 0 and σ = 1 then it is said to be standard normal distributed. The illustration in Figure 4.6 shows the probability density function of Gaussian distributed variables.

[5] EMI and EMP



(a) (b)

Figure 4.6: Gaussian probability density functions. (a) PDFs of various mean and deviation values σ [49, modified]. The larger the σ value the higher is the peak of the function. The „coverage limits“ come closer to the coordinates origin. (b) Gaussian random variable and its probability density values [78, modified].

For a Gaussian random variable, the probability that the values which deviates from the mean z are in the range of $z \pm\sigma$ is approximately 0.68. The probability is 0.95 that the values of $\overline{z}$ are in the range $z \pm 2\sigma$ [78].
Figure 4.7 depicts the effect of a white Gaussian noise and the corresponding histogram of a value. The distribution of sample values contained in the signal posses a bell-shaped curve.

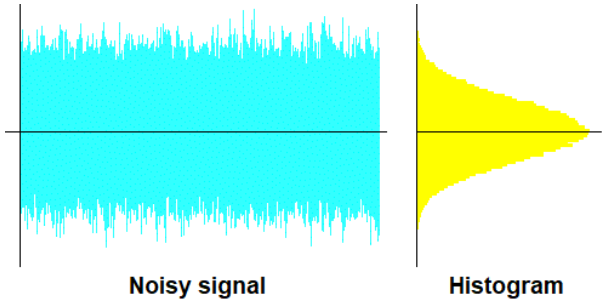


Figure 4.7: Effect of a white Gaussian noise. Left: A signal interfered by white Gaussian noise. Right: Frequency of values in the signal depicted in a histogram [9, modified].

The values near the mean occurs more frequently in the total signal sequence, 4.7 (left). If the noise level is increased the values deviation from the mean value, which represents the failure-free value for each rotational encoder magnet position in the simulation, becomes larger. Thus, the bell-shaped histogram curve becomes wider, so a wider margin is „covered“ within the PDF.

In the evaluation software the function called *add_noise* was created for interference of field components from simulation by additive white Gaussian noise. A use of MATLAB® *awgn*-function is made. It is assumed that the data has a signal power level measured in dBW of 0, what is 1 Watt. The interference of magnetic data $\mathbf{H_x}$ and $\mathbf{H_y}$ by white Gaussian noise is performed as follows:

1. Calculation of magnitudes of field components $\mathbf{M_f} = \sqrt{\mathbf{H_x}.^2 + \mathbf{H_y}.^2}$.
2. Calculation of mean signal magnitude H_{mean} by accumulation of all elements contained in $\mathbf{M_f}$ and division by their total amount.
3. Creation of two independent white Gaussian noise matrices by using *awgn()*-function of the signal power level of 0 dBW, mean value of 0 and desired noise level specified in dB.
4. Multiplication of created white Gaussian noise matrices by mean signal magnitude M_{mean}.
5. Addition of modified white Gaussian noise matrices to the field components $\mathbf{H_x}$ and $\mathbf{H_y}$.

The procedure is done for each rotational position of the encoder magnet.

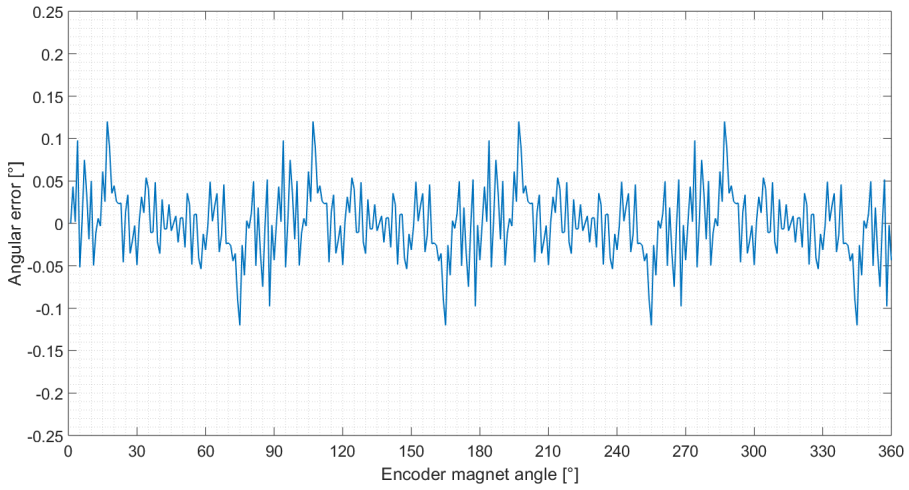
The question in relation to the implementation of the *add_noise* and made assumptions in which need to be answered as first is: How to proof the correctness of made assumptions and noise effects in relation to the reality? The question which is resulting by consideration of the theory is: How is the amount of sample size to set, what represents in the present case the amount of repetitions of noise interfering within the full encoder magnet rotation, in order to reach an appropriate confidence level for the conclusions about maximal angular error $\varphi_{err_{max}}$ caused by noise?
By consideration of Formula D.2 it follows to the conclusion that the impact of an ideal ADC in dependence of its resolution can be also modeled by a simple noise function which is expecting a SNR specification as transfer parameter in dB, what is the evaluation the case. Within the context of functionality proof of the created method for noise interference can be examined by a comparison of noises caused by quantization and Gaussian PDF. This can be considered as the validation procedure of the method.
As one option for the answer to the second question, the required sample size could be calculated by rearranging the equation for confidence interval calculation[6], calculation of standard deviation and an assumption for the margin error[7], only under the assumption that the angular error caused by noise interference is normally distributed (what is not necessary the case, due to nonlinearity of arctangent function). The second option is more suitable due to its practicability related to simulation possibilities and was therefore applied in for determination of repetitions needed for reliable conclusion about the angular accuracy with satisfied confidence level.
For the proof of functionality of created noise function an ADC resolution of 10 bit was chosen. In correspondence to the Equation D.2 the noise level caused by the chosen ADC resolution is

[6] See [4].
[7] See [4] and [6].

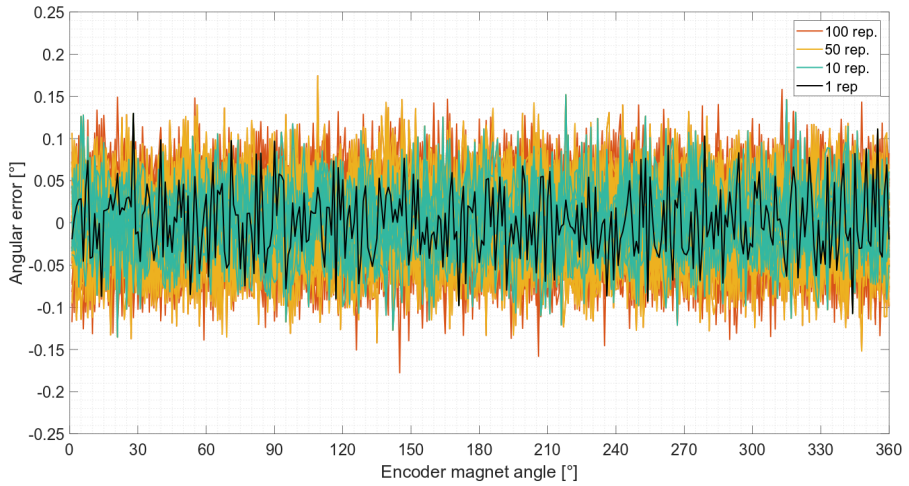
61.96 dB. Hence, it is expected that the quantization noise caused by an ADC with 10 bits resolution will have an approximately identical impact on angular accuracy at the closest z-distance of 2 mm where the $\pm$ range of field data corresponds approximately to the dynamic range of the ADC.
In order to determine the suitable amount of repetitions for all investigation associated with SNR change 4 variations of repetitions were applied: 1, 10, 50 and 100. The angular error results were analyzed optically.

In Figures 4.8 and 4.9 two comparisons of quantization and additive white Gaussian noise impacts on angular accuracy at z-distances 2 mm and 5 mm are shown. Array size was set to 15 x 15 elements. The sphere encoder magnet was chosen. No misalignment was applied in this setup. For the calculation of the angle the approximation approach described in Section 3.3.3 was applied.
From the comparison of results in Figures 4.8(a) and 4.8(b) the practical suitability created SNR function is evident. The quantization and white Gaussian noise levels are equivalent. Figures 4.8(b) and 4.9(b) show that a satisfied coverage of angular error values is obtained at 50 repetitions. Nevertheless, a small enlargement of confidence intervals for maximal angular error is observable by increase of repetitions from 50 to 100.

Conclusions: The practical suitability of created noise function based on Gaussian PDF was confirmed. As a result out of the performed pre-test for confidence levels it is proposed to keep the amount of repetitions $\geq$ 100 to get a confident conclusion about the maximum angular error caused by magnetic noise interference, modeled as additive white Gaussian noise.

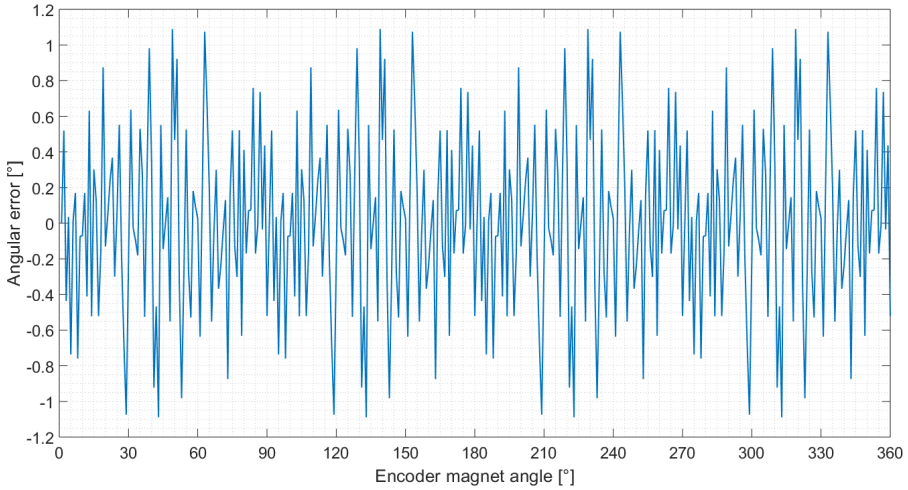


(a)

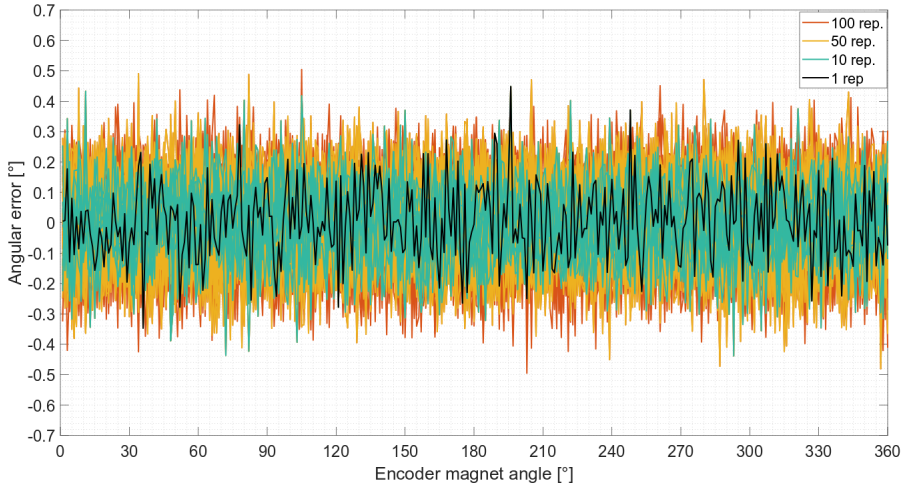


(b)

Figure 4.8: Comparison of quantization and white Gaussian noise impacts on angular accuracy at 2 mm distance from the encoder magnet. (a) Angular error due to quantization noise of applied ADC with 10 bit resolution. The angular error values caused by quantization noise of the ADC are within the same range as the error caused by created noise function shown in Figure (b). (b) Angular error due to interference of additive white Gaussian noise of a SNR of 61.96 dB. 4 different amounts of repetitions are shown in this Figure. A significant enhancement in coverage of the angular error values in observed by in increase of repetitions from 1 to 10. Coverage increase is also between 10 and 50 repetitions visible. An increase of repetitions from 50 to 100 is marginal, only a tiny enhancement is observable. The spikes that represents the maximum possible angular errors are located almost at the same limits. *Simulation setup: 15x15 sensor array, 2 mm z-distance, sphere encoder magnet. Approximation based algorithm results.*



(a)



(b)

Figure 4.9: Comparison of quantization and white Gaussian noise impacts on angular accuracy at 5 mm distance from the encoder magnet. (a) Angular error due to quantization noise of applied ADC with 10 bit resolution. The range of angular error values caused by quantization noise of the ADC is higher than in (b). This is due to significant loss of usable effective dynamic range of the ADC because of the field weakening. (b) Angular error due to interference of additive white Gaussian noise of a SNR of 61.96 dB. 4 different amounts of repetitions are shown in this Figure. Identical effects as in Figure 4.8 of coverage of angular error values associated with an increase of repetitions are observable. *Simulation setup: 15x15 sensor array, 5 mm z-distance, sphere encoder magnet. Approximation based algorithm results.*

Device specific SNR representation

In general, the signal-to-noise ratio of a measured signal is calculated either by the ratio of signal power to power of noise or by ratio of its voltage RMS value to the noise voltage RMS value by the equation:

$$SNR = 10log_{10}\left(\frac{P_{measured\ signal}}{P_{noise}}\right)dB = 20log_{10}\left(\frac{U_{RMS\ measured\ signal}}{U_{RMS\ noise}}\right)dB \tag{4.5}$$

The RMS value of a signal is calculated by the Formula 4.22 denoted in Section 4.3.5. Also here the assumption of sinusoidal input signal waveform leads to a SNR calculation derivation. Thus, the SNR of a signal becomes determinable by equation:

$$SNR = 20log_{10}\left(\frac{\frac{1}{\sqrt{2}}\cdot\hat{u}}{U_{RMS\ noise}}\right)dB \tag{4.6}$$

where $\hat{u}$ is the peak amplitude of the signal and $U_{RMS\ noise}$ is the indication of the noise level of the sensor in the data sheet. In the data the amplitude level is often given as a peak to peak amplitude value or as a [max, typ, min]-value in mV/V, so in relation to the supply voltage V_{cc} (also often denoted as V_{dd}). The RMS value of the noise is usually specified in mV, so independent from the voltage level of the measured output signal. Therefore, a more convenient way for the SNR calculation equation based on data given in a data sheet is:

$$SNR = 20log_{10}\left(\frac{\frac{1}{\sqrt{2}}\cdot\frac{1}{2}pp_{value}\cdot Vcc}{U_{RMS\ noise}}\right)dB \tag{4.7}$$

where pp_{value} is the typical peak to peak factor which needs to be multiplied by voltage supply level Vcc. This formula can be directly applied for a calculation of signal-to-noise ratio of a sensor specified in its data sheet.

Gain and offset deviations

Besides the ADC quantization and magnetic interference effects it is also necessary to consider the impacts on the sensor signals caused by electronic signal conditioning units in order to replicate the system behavior in the simulation closely to a real one. Therefore, the possibility for representation of impacts of gain and offset deviations of operational amplifiers was created in the software.

Operational amplifiers are needed for analog preprocessing of measured signals and the adaption of impedance in a sensor system. Due to their manufacturing tolerances and inequalities in the resistor networks the nominal peak to peak values and offsets levels of the output signals may vary.

For the purpose of representation of these effect the function *add_offset_variations()* and

add_gain_variations() were created. The applied references for the setup of gain mismatches and offset deviations are explained in Appendix D.3.3.
In both functions the *normrnd*-function is used, which generates random numbers from the normal distribution with mean parameter μ and standard deviation parameter σ. In corresponding μ and σ parameters are treated as follows:

- **Gain mismatch**
  The parameter for the gain variation is related to the percentage deviation of the gain factor of 1 what is the mean parameter μ. Correspondingly to derivations above, the value for 1 σ is obtained by division of passed gain variation parameter by 6, in accordance to automotive standard described in Appendix D.3.3. The „measured sensor array“ values are multiplied by values contained in the created gain deviation matrix once for the whole encoder magnet rotation.
- **Offset deviation**
  The adjustment for the offset level is related to the full operational range specified in Section 4.2.4 of 60 kA/m, what corresponds to the full dynamic range of the ADC. Correspondingly, the mean parameter μ is 0. The offset level is hence represented as an A/m value. As well as in the function for the gain variations the value for 1 σ is obtained by division of passed offset variation parameter by 6. To the „measured sensor array“ values the values contained in created offset deviation matrix are added once for the whole encoder magnet rotation.

Fixed-point arithmetic

Within the framework of further implementation of algorithms on the ASIC the limitation for arithmetic operations by reason of limited register bit width for data representation as well as for multiplications computation plays also a role of a particular importance. In order to be able to investigate the impact of arithmetic accuracy limitation on the angle calculation the evaluation software was extended by functions included in MATLAB® Fixed-Point Designer Toolbox™. By use of the methods contained Fixed-Point Designer Toolbox[8] the numbers can be converted to fixed-point numeric objects of particular word length and user specified fixed-point settings for arithmetics can be applied.
In order to provide an understanding about performed fixed point arithmetic investigations a briefly explanation about the binary number formats was written and can be found in Appendix D.3.4.
Within the framework of the project the s1Q10 fixed-point data format was specified as an appropriate format for the digital representation of values. This is the standard format defined for hardware-specific arithmetic operations in the project group. Hence, the significance of values represented in this format is as follows:

- The bit width for the signed integer value is set to $n = 1$.

[8] https://de.mathworks.com/products/fixed-point-designer.html

- The bit width for the representation of decimal value behind the radix point is set to $m = 10$. Thus, the overall accuracy of the system is depending on this bit width.
- As a result of format specification the full coverage of the numbers which can be represented in s1Q10 format is [-2, 2).

The following basic attributes needs to be specified by conversion of standard double precision number to a fixed-point number by using the *fi()*-function:

- **Signedness:** (1 or 0) *(Denotes the bit presence for sign representation of the value.)*
- **WordLength:** (1 or 0) + m + n *(Total length of the register where the value incl. sign bit is stored. The m represents the bit width for the integer and n the fractional part.)*
- **FractionLength:** n *(Fractional part length of the register. In applied case the accuracy of arithmetic operations is depending on this parameter.)*

After the conversion procedure the numbers becomes objects of embedded.fi-class. For an adequate carry out of computations it is proposed to set the following properties for the arithmetic control[9] of fixed-point numeric objects:

- **RoundingMethod:** 'Floor' *(Specification for rounding method)*
- **OverflowAction:** 'Saturate' *(Specified to avoid a wrap-around)*
- **ProductMode:** 'SpecifyPrecision' *(Indication of user dependent product precision spec.)*
- **ProductWordLength:** ((1 or 0) + m + n) * 2 *(needs to have the double sized bit width as the total word length due to multiplication operation)*
- **ProductFractionLength:** n * 2 *(needs to have the double sized bit width as the fraction length due to multiplication operation)*
- **SumMode:** 'SpecifyPrecision' *(Indication of user dependent addition precision spec.)*
- **SumWordLength:** (1 or 0) + m + n + 1 *(needs to be one bit more due to addition operation)*
- **SumFractionLength:** m *(Stays the same as the standard fraction part in applied case)*
- **CastBeforeSum:** 'false' *(Indication of user controlled casting)*

For the control of arithmetic operations either *fimath()* or *globalfimath()* can be used.
In the software, the parameters **ProductWordLength** and **ProductFractionLength** are coupled and adjustable direct from the GUI. If the control flag *FI_FLAG* for fixed-point arithmetic operations is set, the field data is at first appropriately preprocessed by scaling operation and then converted to the specified format, using the *fi()*-function. The arithmetic operations are controlled globally using the *globalfimath()* function. If the control flag is set, the functions for angle calculation are operating directly on the fixed-point converted data. The following procedure is done in the function *normalize_data_to_S1QN*:

[9] https://de.mathworks.com/help/fixedpoint/gs/fixed-point-arithmetic_bt25flf-1.html (rounding control)
https://de.mathworks.com/help/fixedpoint/ug/precision-and-range.html (fixed-point arithmetic procedure)
https://de.mathworks.com/help/fixedpoint/ug/using-fimath-productmode-and-summode.html (modes explanation)

- Calculation of common scaling factor in order to maximize the coverage of sampled components within the bit width of used registers.
- Up-scaling of sampled values to the defined maximum number limit ($\pm$ 2 limit for the present case in accordance to preliminary definition within the project group) in order to obtain the maximum possible numerical resolution. The lsb which do not pass to the positive range is considered as an acceptable numerical quantization error.
- Conversion of scaled sampled values to S1Qn-format by use of *fi()* function.

4.3 Mathematical descriptions for evaluations

In the following, all the mathematical descriptions, applied and derived formulas for evaluation of algorithmic results and its performance rating are explained.

4.3.1 Angular accuracy

The angular accuracy of algorithms was rated on two criteria: the maximum and the mean angular error. For the determination of absolute limits of angular accuracy, e.g. in case of investigations of quantization noise effects or maximum misalignment errors, the maximum angular error represents is a necessary criterion. For the purpose of tendency identification the quantization and noise effects play a subordinate role and therefore the mean angular error is a more meaningful criterion.
The maximum angular error is determined as the maximum absolute difference between the expected and calculated angle over the full encoder magnet rotation, by the following formula:

$$\varphi_{err_{max}} = max\big(|\varphi_{expected}(n) - \varphi_{calculated}(n)|\big) \tag{4.8}$$

where n represents the encoder magnet position in range of 0° to 359°, discretized in this simulation setup in 1° steps, so, equal to the real encoder magnet angle.
The mean angular error is determined as the mean value of absolute differences between the expected and calculated angles

$$\varphi_{err_{mean}} = \frac{1}{N}\sum_{n=0}^{N-1}\big(|\varphi_{expected}(n) - \varphi_{calculated}(n)|\big) \tag{4.9}$$

where N represents the total amount of rotational encoder magnet positions within a full rotation and counts to 360, respectively.

4.3.2 SNR sweep dependent angular accuracy

All calculation where the SNR parameter is varied were repeated. The amount of repetition was set to $R = 100$, according to the empirically determined sufficient level of confidence in Section 4.2.4. In correspondence to it, the formulas for maximum and mean angular error were

extended by the SNR repetition parameter. As a consequence for using the SNR parameter as an input parameter for the tests, the worst angular error which occurs within the whole set amount of repetitions must be evaluated, in order to reach the absolute maximum limit for angular error covered by normal distribution curve. Hence, in both formulas the maximum error is of interest. It therefore follows for the maximum angular error

$$\varphi_{err_{max}} = max\big(max\big(|\varphi_{expected}(n,r) - \varphi_{calculated}(n,r)|\big)\big) \tag{4.10}$$

and for the mean angular error

$$\varphi_{err_{mean}} = max\big(\frac{1}{N}\sum_{n=0}^{N-1}\big(|\varphi_{expected}(n,r) - \varphi_{calculated}(n,r)|\big)\big) \tag{4.11}$$

where r is the actual repetition of the current SNR sweep step.

4.3.3 AMM localization accuracy

The accuracy of tracking algorithms for localization of the absolute magnitude maximum point within the sensor array is assessed by the calculation of mean distance deviation (MDD) based on euclidean distance calculation. In analogy to the SNR dependent angular determination, the MDD calculation is depending on SNR and amount of repetitions of the current SNR sweep step as well. The MDD is calculated by the formula:

$$MDD = \frac{1}{R}\sum_{r=1}^{R}\frac{1}{N}\sum_{n=0}^{N-1}\sqrt{\Big(x_{coord\ exp.}(n,r) - x_{coord\ loc.}(n,r)\Big)^2 + \Big((y_{coord\ loc.}(n,r) - y_{coord\ loc.}(n,r)\Big)^2} \tag{4.12}$$

The n represents the rotational encoder magnet position in range of 0 ° to 359 ° and N its total amount. $x/y_{coord\ exp.}(n)$ are the expected, ideally localized x-/y-coordinates of AMM point at angle n without noise interfering and $x/y_{coord\ loc.}(n)$ are the localized x-/y-coordinates of AMM point at angle n by investigated tracking algorithms. This values are dependent on the actual SNR sweep step. The r is the actual repetition of the current SNR sweep step and R its total amount.
The ideally localized x-/y-coordinates of AMM points $x/y_{coord\ exp.}(n)$ are determined by extraction of x- and y- coordinate of the maximum magnitude value contained in $\mathbf{M_f}$:

$$max(\mathbf{M_f}) = max(\sqrt{\mathbf{Bx}.^2 + \mathbf{By}.^2}) \tag{4.13}$$

The deviation from mean AMM point is calculated in analogy to 4.12 but without the summation over N:

$$MDD_{mean\ AMM} = \frac{1}{R}\sum_{r=1}^{R}\sqrt{\Big(x_{coord\ exp.}(r) - x_{coord\ exp.d}(r)\Big)^2 + \Big((y_{coord\ loc.}(r) - y_{coord\ loc.}(r)\Big)^2} \tag{4.14}$$

All the deviations from expected AMM points are measured in sensor-to-sensor distance units, always in relation to the investigated array size.

4.3.4 Optimization and field deformation indication

Within the context of this work the optimization of parameters needed for the modified approximation approach described in Section 3.3.3 and examination about shape deformation of measured field data associated with filter applications was carried out. For these investigations, formulas described in the following are introduced.
The deviation of field components can be expressed by determination of magnitude differences between ideal and calculated field components considered in a xy-plane. The difference of each field component is included into the euclidean distance calculation in order to consider the direction deviation of components. The magnitude differences between ideal components $\mathbf{H_{x\_ideal}}$ and $\mathbf{H_{y\_ideal}}$ and calculated components $\mathbf{H_{x\_calc.}}$ and $\mathbf{H_{y\_calc.}}$, e.g. by approximation, is determined by:
$H_{magnitudes\_error}(n, m, l) =$

$$\sqrt{\left(H_{x\_ideal}(n, m, l) - H_{x\_calc.}(n, m, l)\right)^2 + \left(H_{y\_ideal}(n, m, l) - H_{y\_calc.}(n, m, l)\right)^2} \tag{4.15}$$

To get the representation of the error in total, the mean value of the magnitudes errors is calculated, by the following equation:

$$\mathrm{M}_{\text{mean error}} = \frac{1}{L}\sum_{l=0}^{L-1}\left(\frac{1}{NM}\sum_{n=1}^{N}\sum_{m=1}^{M} H_{magnitudes\_error}(n, m, l)\right) \tag{4.16}$$

The introduced mean value of the magnitudes errors in Equation 4.16 is denoted in following as mean error of field components error MEFC:

$$\mathrm{MEFC} = \mathrm{M}_{\text{mean error}} \tag{4.17}$$

The mean magnitude of expected, ideal field components is determined alternatively to 4.16 and 4.15 without subtraction of approximated components, logically:

$$H_{magnitudes\_ideal}(n, m, l) = \sqrt{H_{x\_ideal}(n, m, l)^2 + H_{y\_ideal}(n, m, l)^2} \tag{4.18}$$

$$\mathrm{M}_{\text{mean ideal}} = \frac{1}{L}\sum_{l=0}^{L-1}\left(\frac{1}{NM}\sum_{n=1}^{N}\sum_{m=1}^{M} H_{magnitudes\_ideal}(n, m, l)\right) \tag{4.19}$$

In all described formulations the variable l represents the encoder magnet position in range of 0° to 359°, discretized in 1° steps. L is the amount of angular encoder magnet positions equal to 360. Variables n and m are the coordinates within the considered xy-plane and NM is the

amount of considered points in the xy-plane.
To obtain a relative indication about the error of field components, the MEFC is calculated in percentage relation to the ideal component:

$$\mathrm{MEFC}_{\%} = \frac{100\%}{\mathrm{M}_{\text{mean ideal}}} \cdot \mathrm{MEFC} \tag{4.20}$$

The Formula 4.20 was applied also for the indication for deformation of the field shape caused by application of filters within the framework of pre-processing investigations and filter optimizations in Section 5.3.

4.3.5 Signal quality analysis

For evaluation of rating of results of angular error correction algorithms a formalism which is described in the following was derived.
The quality of output signals while the sensor is being exciting with a rotating field $\vec{H}_{ext}$ is analyzed by calculation of the total harmonic distortion factor THD [61]:

$$THD = \sqrt{\sum_{h=2}^{H} \left(\frac{Q_h}{Q_1}\right)^2} = \frac{\sqrt{\sum_{h=2}^{H} Q_h^2}}{Q_1} \tag{4.21}$$

in accordance to DIN EN 61000-2-4 [18] and identical distortion measurement THD in IEEE 519-1992 [17]. Q_1 is the RMS-value of the fundamental rotation frequency of encoder magnet and Q_h is the RMS-value of each harmonic, i.e integer multiples of the fundamental rotation frequency. H represents the amount of included harmonic frequencies.
The RMS-value of a signal, e.g. voltage, is calculated by the following formula [84]:

$$Q = U_{RMS} = \sqrt{\frac{1}{T}\int_0^T u(t)^2 dt} \tag{4.22}$$

where T is the one period time and $u(t)$ the measured signal. Corresponding to it, the RMS-value Q of a time discrete signal $q(n)$ is calculated by:

$$Q = \sqrt{\frac{1}{N}\sum_{k=0}^{N-1} q(k)^2}, \quad 0 \leq k \leq N-1 \tag{4.23}$$

where N is the amount of samples within one period and $q(k)$ the sampled value of measured signal. For the special case of ideal sinusoidal signals their RMS value is determined by [84]:

$$Q_{RMS} = \frac{1}{\sqrt{2}} \cdot \hat{u} \tag{4.24}$$

where $\hat{u}$ is the peak amplitude of the signal.
The amount of frequencies and their phases contained in a finite-length angle equidistant measured sensor signal $x(k)$ can be calculated by application of one dimensional discrete Fourier transform [83]:

$$X(n) = \sum_{k=0}^{N-1} x(k) \cdot e^{-j2\pi nk/N} = \sum_{k=0}^{N-1} x(k) \cdot W_N^{nk}, \quad 0 \leq n \leq N-1 \tag{4.25}$$

The harmonic frequencies as well as the fundamental frequency can be extracted by the spectral analysis of obtained DFT results. Such techniques are typically applied for conditions diagnostics of magnetic sensors, such as speed sensors in anti-lock braking systems [61].
Based on the property of Fourier transform [79] for sinusoidal signals the discrete Fourier transform of a sine and cosine signal leads to

$$cos(2\pi \frac{n_0}{N} k) \circ\!\!-\!\!\bullet \frac{N}{2}(\delta(n+n_0)_{modN} + \delta(n-n_0)_{modN}) \tag{4.26}$$

$$sin(2\pi \frac{n_0}{N} k) \circ\!\!-\!\!\bullet \frac{N}{2} j(\delta(n+n_0)_{modN} - \delta(n-n_0)_{modN}) \tag{4.27}$$

where $\delta(n)$ is a discrete Dirac impulse of the value of 1.
If the magnitudes of spatial frequencies have been calculated, the THD factor can be calculated directly by extension of Equation 4.21 by RMS Equation 4.24 inclusive consideration of factor 2 from Equation 4.26, what leads to the resulting formalism:

$$THD = \frac{\sqrt{\sum_{h=2}^{H} \left(\frac{1}{\sqrt{2}} \cdot \hat{f}_h \cdot 2 \right)^2}}{\frac{1}{\sqrt{2}} \cdot \hat{f}_1 \cdot 2} \tag{4.28}$$

where $\hat{f}_n$ are the magnitudes of frequency components. This Equation is valid for consideration of components within the frequency range of 0 to $F_S/2$. The derived Equation 4.28 was used in the context of performance analysis of angular error correction algorithms in Section 5.7.5. All harmonics components were included into the calculations of THDs.

5 Investigations

This chapter represents the second main part of this thesis.
At the beginning the system conditions with respect to the behavior of encoder magnet fields are examined. Afterwards, the considered methods for the possible preprocessing are investigated and their suitability according to the present problem is proved. In the second step, parameter optimizations are carried out. Thereafter, a preliminary investigation of algorithms is conducted. In the next step, the created test plan for investigation of algorithms within the full frame of the Safe Operating Area is explained in detail. In full accordance to the investigations plan, the limits for required angular accuracy within the SOA are determined and the potential of algorithmic extensions for the accuracy increase is exhibited. The algorithm results, their potential and performance are discussed in detail and the conclusion for the implementation proposal is made. Finally, a summary of the obtained results and discussion about the gained knowledge is given. Strategic proposals are made, not only regarding the algorithms itself but also in relation to the physical construction of the sensor array as well as the encoder magnet system, which will make use of the advantages found during the investigations.

5.1 Field conditions analysis

Among the most important information in relation to the aspired development of a sensor array system is the knowledge about the strength of the gradients which are measured by a sensor array within the full operational area.
This information is necessary for hardware-specific designs and dimensioning of dynamic ranges of data acquisitions units. Furthermore, a direct comparison of field gradients in the field of the sphere and the commercial disc magnet is required in order to draw a conclusion about the applicability, get an understanding about the gradients behavior and its characteristic parameters. It is necessary to obtain the knowledge about the tendency of reduction of gradient strengths in dependence of the distance from the encoder magnet in order to know in which range of the field components the content about angle information is present and to get a comparison in relation to the strength of the direct components. Last but not least, the physical values of present field components will provide important indicators of field conditions with respect to noise interference in a real application.

With a view to it the $\mathbf{H_x}$ components of the encoder magnets within the full operational range of the z-distance were analyzed. The magnetic field data from total simulated xy-plane was taken into account. Figure 5.1 illustrates the $\mathbf{H_x}$ components of the field generated by the sphere and the disc encoder magnet.

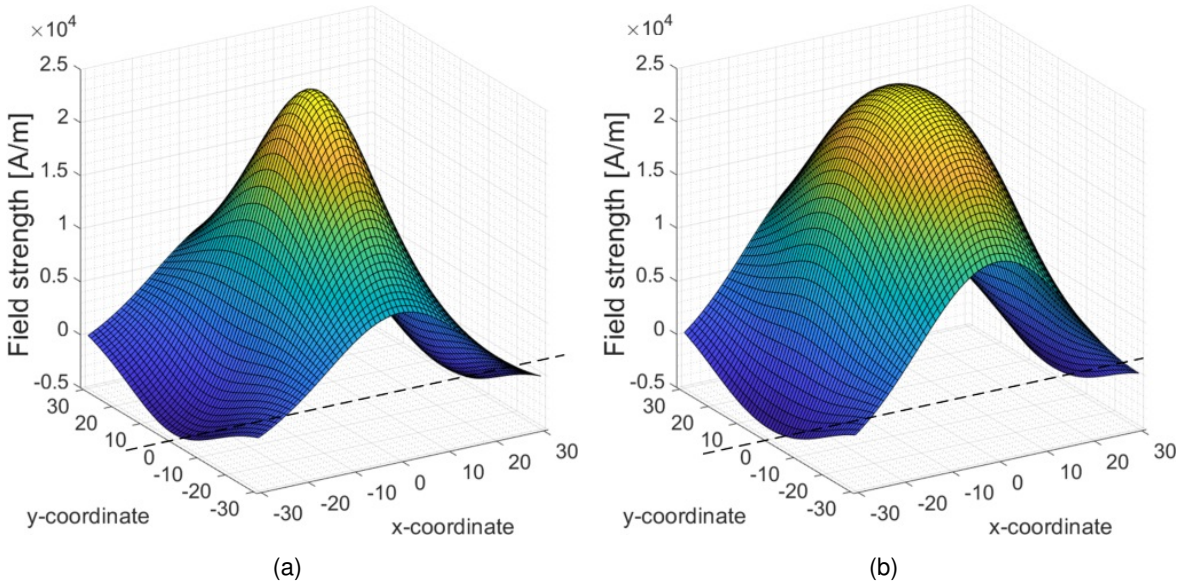
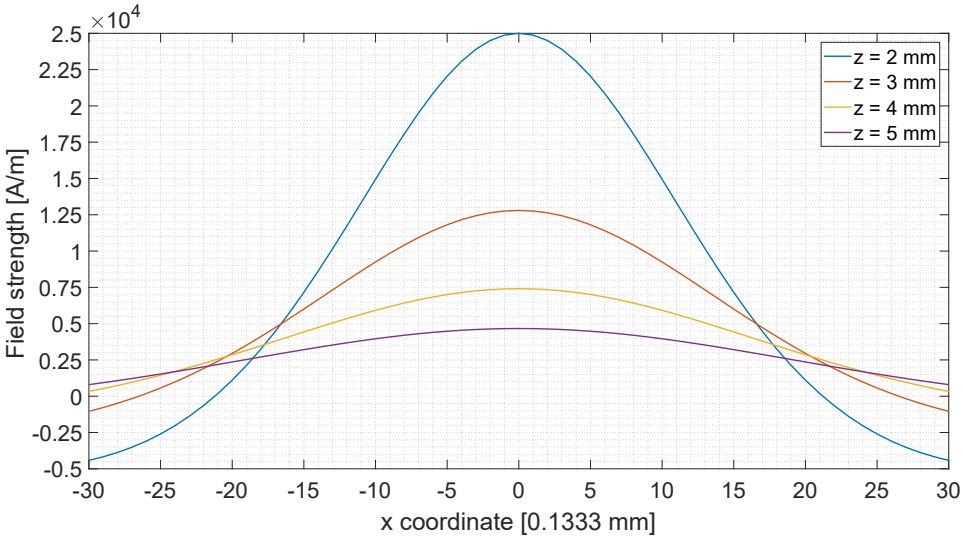
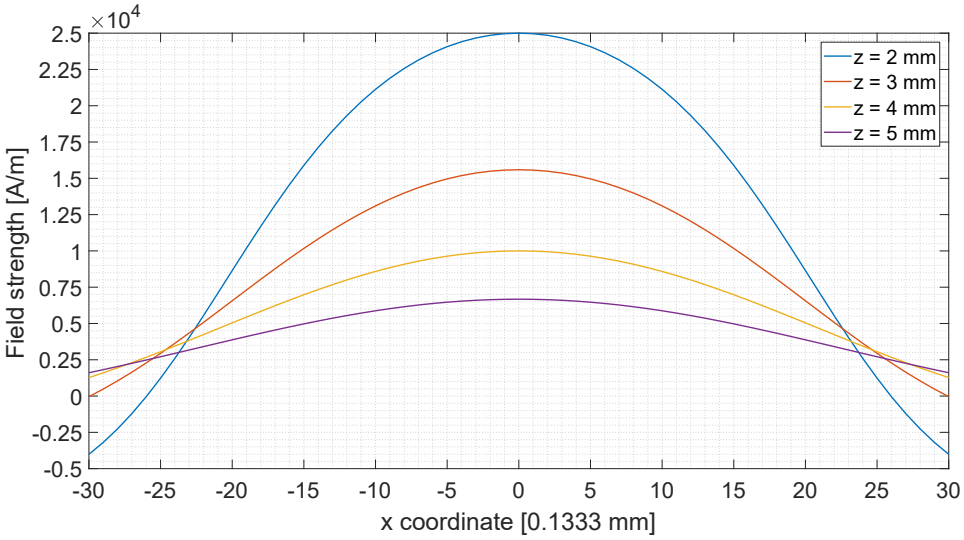


Figure 5.1: Strength of $\mathbf{H_x}$ field components of simulated encoder magnets at z-distance of 2 mm. (a) Sphere magnet $\mathbf{H_x}$ components. (b) Disc magnet $\mathbf{H_x}$ components. The cross sections are illustrated as dashed lines. Both magnet fields exhibit different curvature behavior. The sphere magnet field is much more inhomogeneous.

In order to extract the field gradients the $\mathbf{H_x}$ components were examined at the cross section at the y-coordinate = 0, depicted as dashed line in Figure 5.1(a) and 5.1(b). The $\mathbf{H_y}$ components are equal to zero in this cross section. Thus, the differences of $\mathbf{H_x}$ components represent exactly the field gradients of the magnetic field $\vec{H}$.
The content of the cross sections for both encoder magnets is depicted in Figure 5.2.
From the graphs in 5.2 it can be seen that the initial field curvatures at the z-distance of 2 mm in relation to the center coordinate x = 0 looks differently. The sphere magnet exhibit a higher gradient behavior. However, this higher curvature fails rapidly if the z-distance is enlarged and gets close to these of the disc magnet. In total, the strengths and curvatures of both magnet fields fails promptly by an increase of the z-distance. Remarkable is the fact that at higher distances from the encoder magnet in case of disc magnet more field strength is present.



(a)



(b)

Figure 5.2: Cross sections in the simulated xy-plane in x-direction at y-coordinate = 0 mm. (a) $\mathbf{H_x}$ components of the sphere magnet. (b) $\mathbf{H_x}$ components of the disc magnet. In comparison to the disc magnet field cross section there is a much higher initial curvature at the closest distance to the encoder magnet of 2 mm of sphere magnet field observable. At lower distances the strength of the disc magnet field is higher than of the sphere magnet.

In order to get a knowledge about the constraints resulting from the examined system the field gradient of the components was analyzed. The specification for gradient calculation with respect to the simulation setup is illustrated in Figure 5.3.

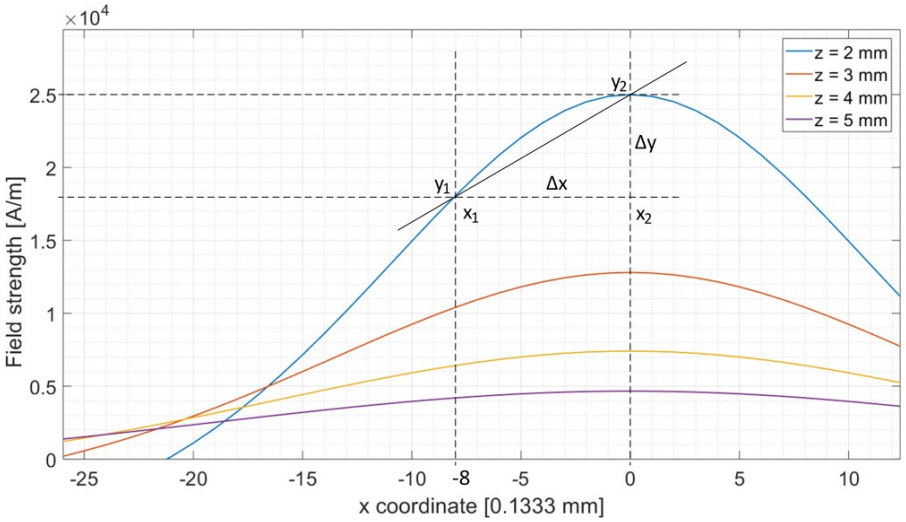


Figure 5.3: Delineation of the gradient calculation from $\mathbf{H_x}$ components at defined $\mathbf{H_y}$ cross section. The field gradient is calculated with respect to the symmetrical center point of the sensor array. The distance Δx was specified to $\approx$ 1 mm.

The field gradient is calculated as the slope m in x direction, within the defined Δx range of 1.06 mm and with respect to the central point of the sensor array, with the following formula:

$$\textit{Field gradient} = m = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1} = \frac{y_2 - y_1}{1.0\bar{6}mm} \tag{5.1}$$

To get a perception about field behavior the field gradient specified in 5.3 was determined in dependence of the z-distance from the encoder magnet. The absolute field strength at x-coordinate = 0 mm, the specified field gradient and their percentage reduction in dependence of the z-distance for both magnet shapes are recorded in the following Table 5.16.

Encoder magnet	**z-distance [mm]:**	2	3	4	5
Sphere	*Field strength [kA/m]*	25	12.8	7.40	4.66
	Field strength reduction [%]	-	48.8	70.38	81.35
	Field gradient [kA/m/mm]	6.50	2.24	0.92	0.43
	Field gradient reduction [%]	-	65.56	85.77	93.31
Disc	*Field strength [kA/m]*	25	15.59	10	6.67
	Field strength reduction [%]	-	37.64	60	73.31
	Field gradient [kA/m/mm]	2.27	1.503	0.86	0.48
	Field gradient reduction [%]	-	34.06	62.2	78.68

Table 5.1: Field strengths, specified field gradients and their percentage reduction in dependence of the z-distance.

From table 5.1 it becomes evident that the decrease of the sphere magnet field strength over the full z-distance is higher than of the disc magnet. Only about 20% of the initial field strength is present at the maximum z-distance. The level of the field strength relative to the initial one of the disc magnet is here about 30%. The field gradients where the information for angle calculation is extracted from behave for both magnets very interestingly. The initial gradient of the disc magnet is almost three times smaller than the gradient of the sphere magnet. However, the reduction of the sphere magnet gradient is higher than that of the disc magnet. More than 90% of the field gradient of the sphere magnet field are lost at the z-distance of 5 mm. A very remarkable fact is that the disc magnet exhibit at z-distance of 5 mm a higher gradient in comparison to the sphere magnet. This comes from the near field to far field transition behavior associated with its shape.
From Figure 5.2 and the Table 5.1 it becomes evident that field range where the angle information is extracted from vary over the full operation distance in z extremely. If the system is operation at large distances from the encoder magnet only a tiny dynamic range of the sampling units is exploited.

Conclusions: The gradients vary over the full z-distance extremely and at the same time high reduction of field strength is observable. The field strength fails from 25 kA/m to approximately 6 to 5 kA/m. However, the differences of the field components, which represents the content of useful information, measured within 1 mm distance vary only from $\approx$ 0.4 kA/m to max. 6.5 kA/m. It therefore follows that a more suitable way for the angle information extraction would be a direct evaluation of the gradients by the ADC units. In this way its dynamic range could be used more appropriately, adjusted to a range of determined values. The gradients could be measured either by differential ADC configuration or by a construction of a sensor array consisting of magnetic gradiometer instead of magnetometer e.g measure the difference values by analog circuits. In such a case, the developed approaches for angle information extraction based on LoG and approximation described in Sections 3.3.4 and 3.3.3 needs to be adapted due to operation on the gradient fields. The angle information evaluation level (order of the derivatives) have to in this case decreased by one. The 2D-DFT approach will most probably also work on gradient fields due to its universality. Such an approach could be definitely investigated in the context of ISAR project continuation.

5.2 Preprocessing

The considered filter methods for preprocessing described in Section 3.4 were applied on simulation data in order to investigate their feasibility according to the present problem and to find out optimal parameters for the future sensor application in the project.

5.2.1 Filtering in spatial frequency domain

Appropriately chosen low pass filtering can on the one hand improve the signal quality and prepare it suitably for the main processing, but on the another hand a filtering can cause a deformation of signals that leads to a distortion of useful information. The answer to the question: in what kind the considered methods for preprocessing of data are associated with distortion of useful information and where are the acceptable parameter limits of applied filters in spatial frequency domain are provided in the following subsections.

Butterworth low pass filtering

As already discussed in Section 3.4.2, the application of the Butterworth low pass filter in spatial frequency domain is considered as the most suitable candidate for the preprocessing. In order to get a knowledge about the deformation of the field components associated with filtering a study was carried out, which examines the field deformation in dependence of the cut-off frequency and the order of the Butterworth filter. For the 2D-discrete Fourier transformation the input signal was treated as complex: $\mathcal{F}\{\mathbf{B_x} + j\mathbf{B_y}\}$.
In Figure 5.4, a comparison of ideal non-distorted and filtered $\mathbf{B_x}$ and $\mathbf{B_y}$ components of a sphere encoder magnet is shown. For this preliminary test, the cut-off frequency as well as the order of the Butterworth filter were set to 1. By non adequate choices of filter parameters, the sampled field components can be drastically deformed and in this way a large part of useful information will be distorted. This is clearly observable by consideration of the results in Figure 5.4.
The central information content of the measured field is found in the lower frequencies regions which has been already shown in Figures 3.7 and C.7. To guarantee a suppression of a higher frequency magnetic interference, optimized filter parameters needed to be found. Appropriate bandwidths and cut-off frequencies need to be determined, to obtain a good filtering effect and do not distort the field shape significantly. In order to avoid the loss of information, appropriate filter parameters depending of the shape of the encoder magnet field must be chosen. For this purpose the following investigation was conducted: The cut-off frequency of the filter was swept from 1 to 15 and its order from 1 to 5, at a z-distance of 2 mm and a sensor array size of 15 x 15 elements. The caused field deformation expected as MEFC over one full encoder magnet rotation was calculated using Formula 4.20. Obtained results are shown in Figure 5.5.

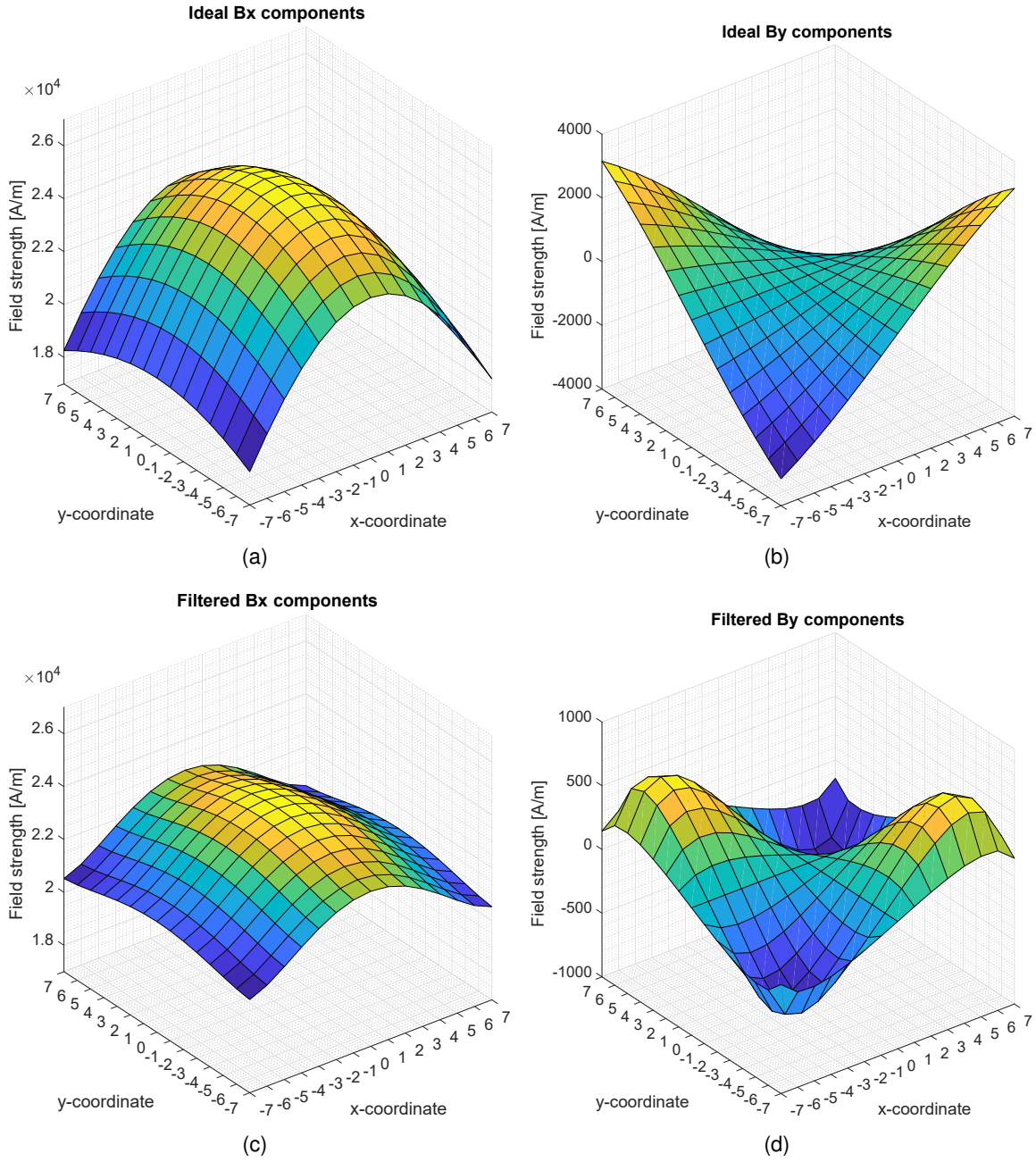


Figure 5.4: Deformation of field components caused by Butterworth low pass filtering. Order = 2 and cut-off frequency = 5. (a) and (b): Ideal field components $\mathbf{B_x}$ and $\mathbf{B_y}$. (c) and (d): Filtered components $\mathbf{B_x}$ and $\mathbf{B_y}$. Due to low cut-off frequency of the Butterworth low pass filter the field components are deformed. It is especially apparent for $\mathbf{B_y}$ components. *Simulation setup: 15 x 15 array, 2 mm z-distance, sphere encoder magnet, angular magnet position = 0°.*

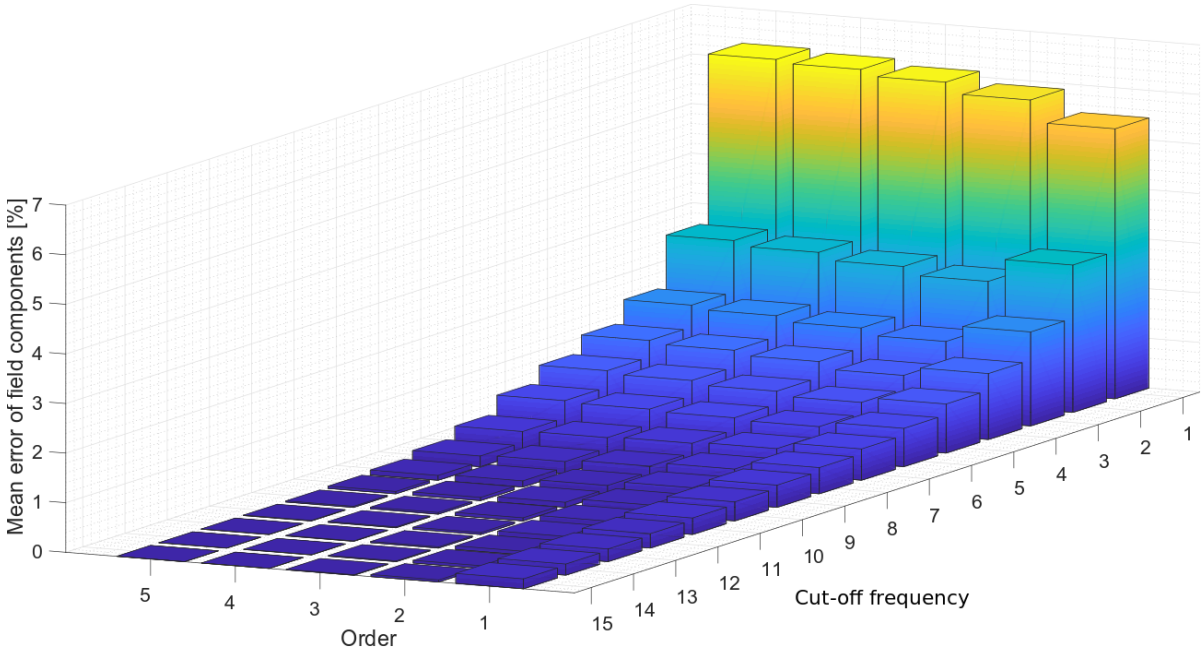
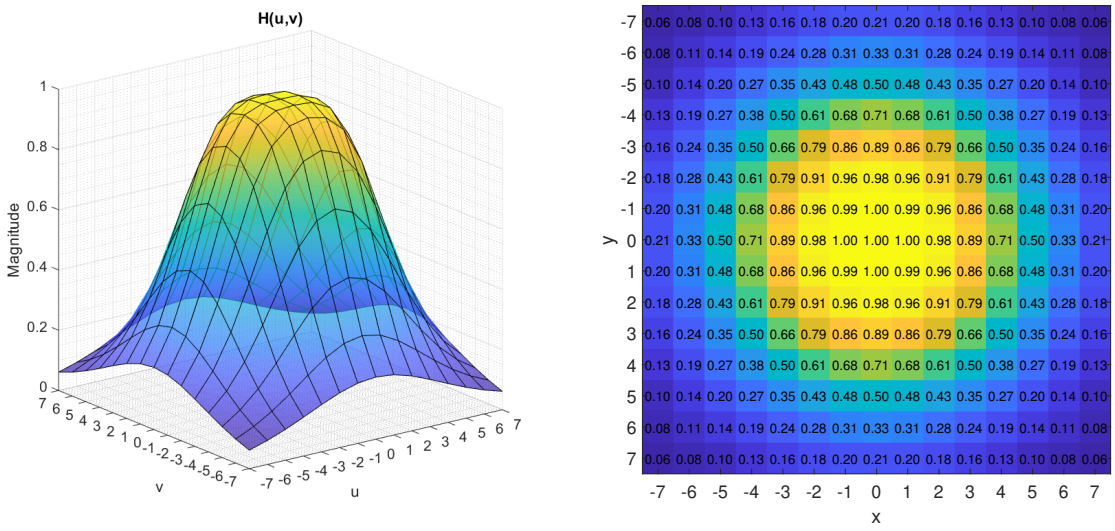


Figure 5.5: Mean error of field components in % between ideal and filtered components caused by application of Butterworth low pass filter depending on the cut-off frequency and order. The largest error reduction is obtained if the cut-off frequency is set to 2. *Simulation setup: 15 x 15 array, 2 mm z-distance, sphere encoder magnet.*

The Butterworth low pass filter with the order and cut-off frequency of 1 exhibits a non satisfying behavior and deforms the field significantly, due to low weighting of first spatial frequencies, what can be seen in 3.18. An increase of cut-off frequency from 1 to 2 leads to a percentage error reduction from approximately 6% to $\leq$ 3%, what represents the highest percentage error decrease. Choosing of subsequent cut-off frequency parameters leads to an error reduction less than 1% after each other. In general, in accordance to results from Figure 5.5 high order filter will guarantee a much better suppression of higher frequent magnetic fields and noise components and in the same time cause less field deformation. But here one needs to be careful. Due to a sharp edged filtering in spatial frequency domain so called ringing artifact could arise if the filtered image after performing of 2D-IDFT [59]. These effects become usually especially visible near sharp transitions of gray values in the picture [78, Chap. 4]. In case of the present problem the measured field components exhibit a low frequency behavior, hence the ringing artifact would most likely not occur. Nevertheless, this effect needs still to be always kept in mind. The results of this investigation are also valid and applicable for suitable filtering of disc magnet fields because there are less higher frequent spatial components contained and due to this the field components percentage error after filtering will be probably even less.
Based on the results obtained from performed preprocessing investigation it is recommended to keep the cut-off frequency and the order of the filter at least ≥ 2. To obtain a smaller field deformation it is recommended to increase the cut-off frequency so that the MEFC is held under 1%.

In accordance to results in Figure 5.5 for the performing of further investigations with Butterworth filter the order of 2 and cut-off frequency of 5 was chosen. In comparison to other filter orders the error is under 1% obtained with these parameters at first.
The Butterworth filter mask with chosen parameters is shown in Figure 5.6. These filter coefficients were applied for the tracking algorithm shown in Figure 3.23 and tested in Section 5.7.3.



(a) (b)

Figure 5.6: Chosen Butterworth low pass filter kernel for preprocessing purposes in spatial frequency domain. Order = 2 and cut-off freq. = 5. Left: 3D view of filter mask. Right: filter mask coefficients matrix.

Butterworth low pass filtering with zero padding

During this work the filtering in spatial frequency domain applied on a zero padded signal was also considered, what is normally almost always the case in signal processing. The advantage of such extension is the projection of spectral frequency components of the input signal to a much wider spectral bandwidth what leads to the fact that the noise contained in the signal will be equally distributed to all higher frequent components. This will automatically cause a much better noise robustness on the one side. But on the other side the main disadvantage of this method, especially with regard to the present problem, is the occurrence of leakage effects in the spatial frequency domain because the joints from where the signal is extended by zero padding will represent a discontinuity that will automatically becomes visible in the spectral domain of 2D-DFT and in real does not represent the useful information. Due to this the useful information contained in the low spectral components will be overlapped with a constant spectrum needed for representation of the leakage.
Before performing of spectral filtering appropriate windows techniques including of subtraction

of constant spatial component were considered and applied in order to decrease the spectral leakage effects. The windowing represents at the same time the disadvantage of this method because the manipulated values at the boundaries cannot be really used for the subsequent information extraction. However, this will practically play a subordinated role if the physical array size and the amount of sensor pixels are increased.
In this experiment only three last elements around the boundaries were modified by windowing. Therefor Tukey (tapered cosine) window was used. In the simulation a tiny translation of sphere encoder magnet in y-direction of 0.26 mm was set in order to observe if the area of maximum field strength gets re-transformed from the 2D-DFT properly when it comes closer to the boundary. The magnetic field components $\mathbf{B_x}$ and $\mathbf{B_y}$ were interfered by white Gaussian noise of an SNR of 38 dB in order to test the robustness of the methods. The 2D-DFT was applied analogously to 5.2.1, on a zero padded 2D input signal increased from 15 x 15 to 31 x 31 elements. To see the differences between non zero padded and zero padded filtering approaches and compare the noise robustness of both methods the same parameters for the Butterworth filter kernel as applied in 5.2.1 were used. The increased applied Butterworth filter kernel is shown in Figure 5.7. The resulting zero padded data are included in Appendix E.1.1. The obtained results are depicted as graphical comparison in 5.8.

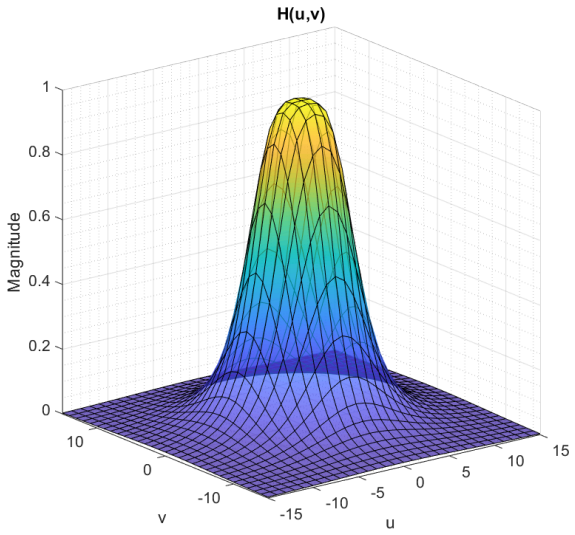


Figure 5.7: Chosen Butterworth low pass kernel for preprocessing purposes in spatial frequency domain applied on the zero padded input signal. Order = 2 and cut-off frequency = 5.

Indeed, in comparison to not increased 2D-DFT, much better noise suppression results could be achieved, what is clearly visible in comparison of Figures 5.8(c) and 5.8(d). The filtered field components in spatial frequency domain of a size 15 x 15 exhibit still a noise interference whereas the components filtered in spatial frequency domain of a size 31 x 31 no remained noise interference is apparent at all. It therefore infers that by zero padded filtering approach the sensitivity against noise is significantly decreased. With this experiment the feasibility and robustness increase of this extension with regard to the present problem has been proved.

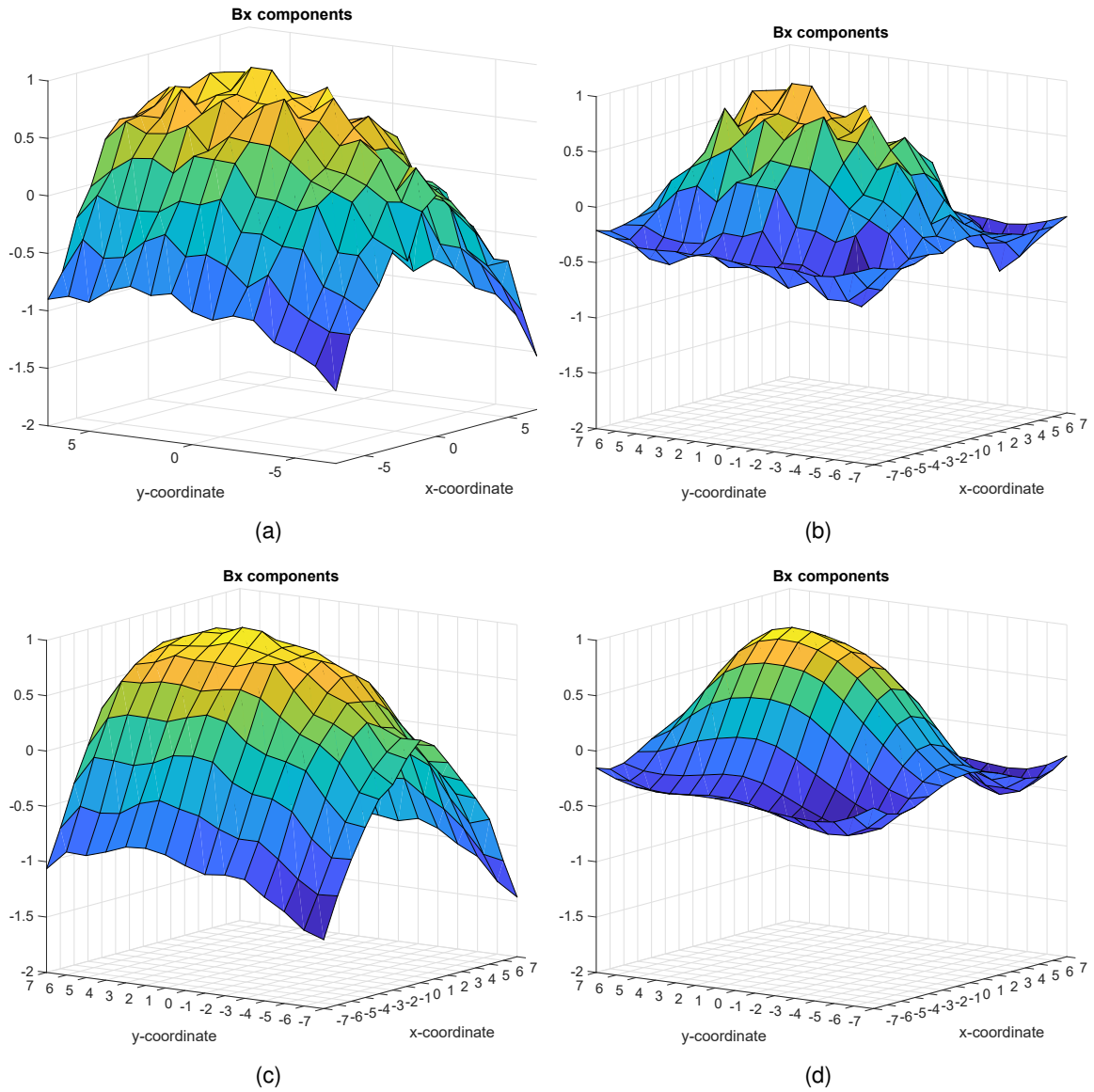


Figure 5.8: Comparison of filtering effects gained from filtering of a not padded and zero padded signals in spatial frequency domain. (a) Distorted $\mathbf{B_x}$ field components. (b) Distorted $\mathbf{B_x}$ field components after application of a window. (c) Filtered $\mathbf{B_x}$ field components in 15 x 15 2D-DFT. Its clearly observable that the noise is still contained in the filtered signal. In addition to that the components at the boundaries are slightly skewed. (d) Filtered windowed and zero padded $\mathbf{B_x}$ field components in 31 x 31 2D-DFT. There are no remained noise components at all. Due to windowing the components at the boundaries are skewed more distinctively. *Simulation setup: 15 x 15 array (data zero padded to 31 x 31 elements), 2 mm z-distance, sphere encoder magnet, angular magnet position = 0°, 0.26 mm y-translation, 38 dB SNR.*

The filtering in spatial domain with zero padding of the input signal and the effects associated with spectral leakages and windowing should be definitely investigated within the scope of further work. For the present problem application of the following most suitable window functions is recommended:

- Tukey window
- Planc-taper window
- Kaiser-Bessel-derived window (KBD)

5.2.2 Median filter application

As the second candidate for preprocessing, the nonlinear median filter was chosen, which advantages are described in Section 3.4.1. This filter is particularly suited to remove high frequency spikes or outliers from the sampled signals. This is extremely crucial for extraction of field curvature from relatively small images and or course, for estimation procedures. On the one hand, the field components could obtain an immense field deformation if a median filter is applied. Deformation is associated with the filtering effect itself as well as due to the handling of boundaries. But on the other hand, the deformation effect could be compensated in case of higher noise levels. It is expected that at certain signal-to noise ratios, break even points could be reached, from where it will become more useful to apply this nonlinear filter not only for removal of spikes but also for standard filtering purposes.
Since the values at the boundaries need to be filtered as well, a suitable boundary handling condition needs to be applied. As the padding option needed for filtering of components at the boundaries, a symmetrical extension was chosen. In order to investigate the field deformation effect associated with median filter of different sizes, the following study was performed: The measured field components $\mathbf{B_x}$ and $\mathbf{B_y}$ of a sphere magnet were interfered by white Gaussian noise in a range from 100 dB to 20 dB and compared with the ideal components. The difference to ideal components was calculated using Equation 4.20. Subsequently, noisy field components were filtered with median filter of mask sizes 3 x 3, 5 x 5 and 7 x 7 and compared to ideal components as well. In Figure 5.9, a comparison between ideal, noisy and filtered field components caused by filtering with the median filter is shown.

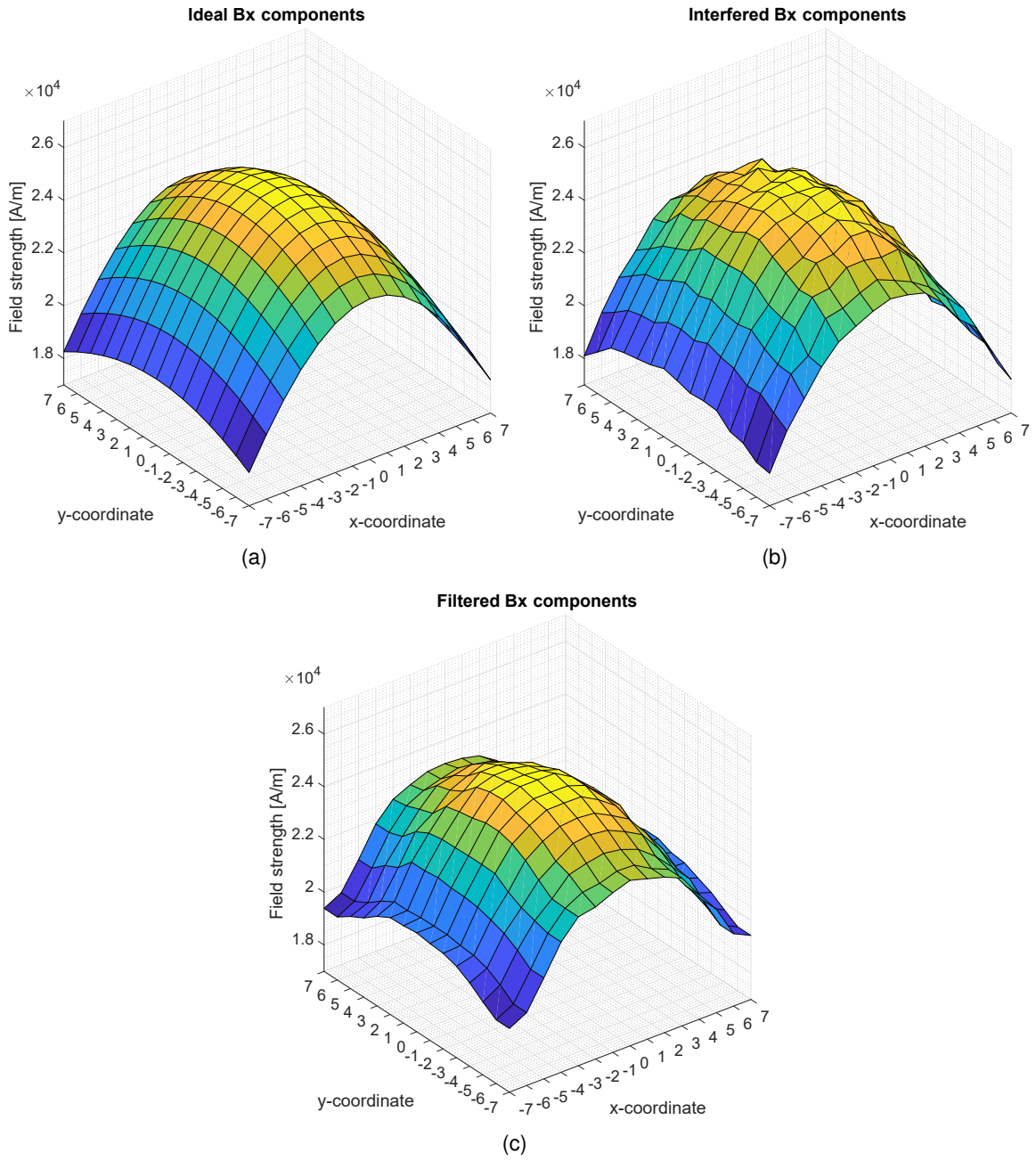


Figure 5.9: Deformation of field components caused by nonlinear filtering with median filter. (a) ideal $\mathbf{B_x}$ components. (b) $\mathbf{B_x}$ components interfered by white Gaussian noise. (c) $\mathbf{B_x}$ components filtered by median filter with of a kernel size = 5 x 5. The noise components are filtering out. But associated with application of the median filter the field components become staircase-shaped deformation. *Simulation setup: 15 x 15 array, 2 mm z-distance, sphere encoder magnet, angular magnet position = 0°, 45 dB SNR.*

In Figure 5.10 the percentage error between ideal and noise interfered as well as filtered components in dependence of SNR is shown.

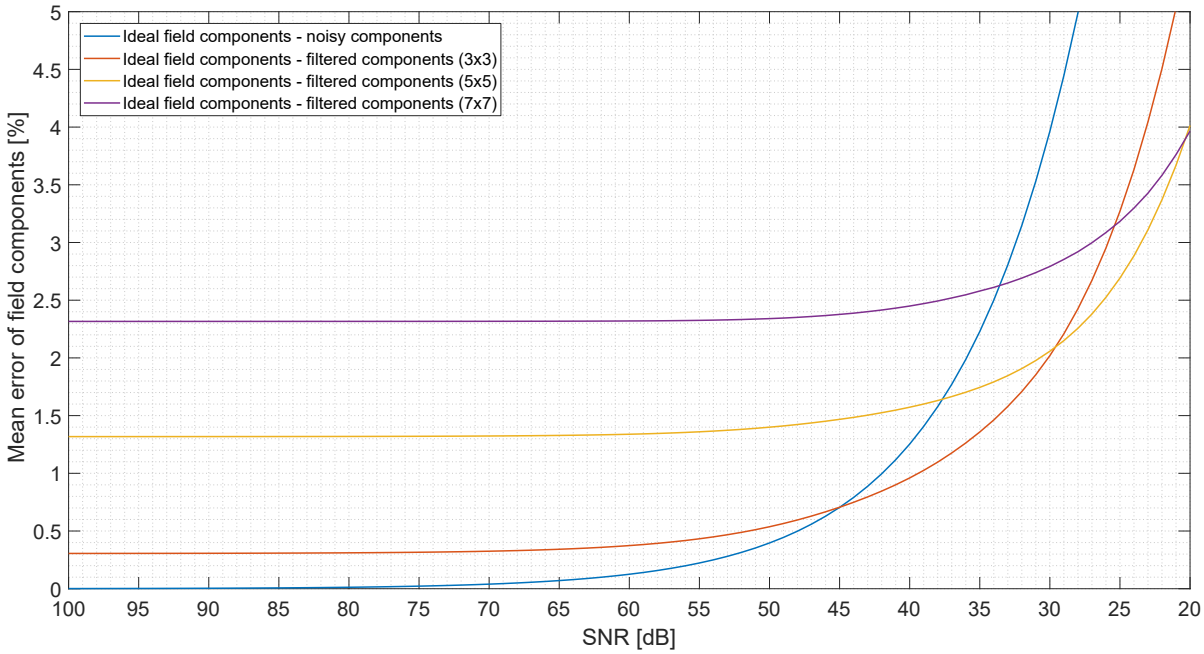


Figure 5.10: Mean error of field components in % interfered by white Gaussian noise and components filtered by median filter in dependence of SNR. Break-even points are observable at around 45 dB, 38 dB and 33 dB SNR for median filter kernels of 3 x 3, 5 x 5 and 7 x 7 elements.

All three masks cause an initial deformation of field components what is below 2.5%. First break-even point is reached at around 45 dB SNR for the 3 x 3 median filter kernel, 38 dB for the filter size of 5 x 5 and 33 dB for 9 x 9 kernel. A notable break-even points between considered filter mask sizes itself is also very interesting. Below the SNR level of 20 dB the 7 x 7 filter mask size will exhibit the best filtering result, below 30 db a 5 x 5 filter mask. However, an SNR area below 30 dB is not really practical relevant. Analogically, the same investigation was performed for a disc magnet and can be found in Appendix E.2. The field deformation cause by median filter is in that case below 1% for all filter masks. The tendency in filtering effects is identical.

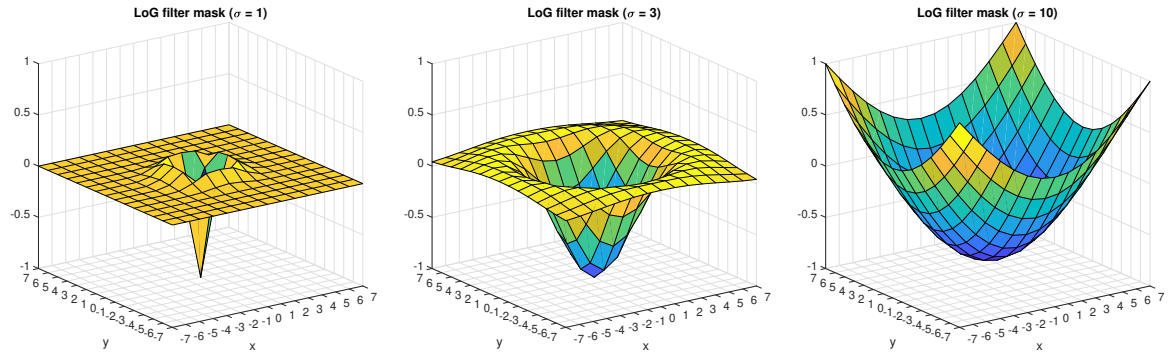
Conclusions: To draw a conclusion one can say that in practice the application of nonlinear median filter in preprocessing step for filtering purpose becomes definitely useful if the quality of sampled signal, according to the applied case, is below $\approx$ 45 dB. In case of application for a guaranty of high frequent spikes removal, the filter kernels will cause initial field deformation in range of determined values what will in any way cause a negative impact on calculation of the angle. This shall be taken into account. The usefulness of application of this filter for possible spikes illumination shall be proved in a real measurement case.

5.3 Optimizations

During preliminary investigations firstly the determination of the optimal σ parameter for the developed approaches based on LoG filter approach was performed. Secondly, the exponent of the field weakening denominator matrix for the disc magnet was determined. This is required by the encoder magnet geometry dependend approximation approach, described in Section 3.3.3.

5.3.1 LoG filter bandwidth optimization

By proper adjustment of theLoG σ parameter, the area from where the second derivative is taken can be increased. This leads automatically to a higher smoothing of the resulting image because in this way the bandwidth of the filter is decreased. But at the same case a much better robustness against noise is achieved. An important aspect that always has to be considered in image processing is the occurrence of rotation variance of filter kernels. For image processing purposes such as convolution with filter kernels in the spatial domain one needs usually to make sure that the approximations of the filter masks fits into the total filter kernel and all elements of the boundaries of the filter kernel equal zero. Otherwise, anisotropy behavior arises due to the abrupt ending of the filter masks what becomes clearly visible by transformation of the filter kernel into the spatial frequency domain. Frequencies sidelobes will come up and in this way additional noise could enter into the filtered images. This will automatically decrease the quality of the filtered image. A simple example to explain this is filtering of an image with a mean filter. In Figures 5.11 and 5.12, different LoG filter masks and their 2D-frequency responses are shown.



(a) $\sigma = 1$ (b) $\sigma = 3$ (c) $\sigma = 10$

Figure 5.11: LoG filter kernels of 15 x 15 size with different σ parameters. Due to the increased σ parameter the curvature contained in the picture is extracted more globally, what automatically leads to an increase of noise robustness.

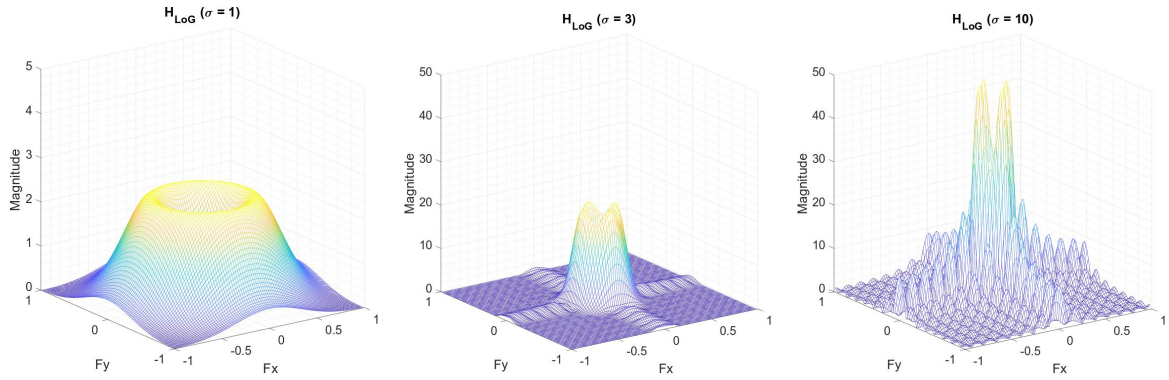


Figure 5.12: LoG filter kernels of 15 x 15 size with different σ parameters transformed into spatial frequency domain. For the calculation of frequency responses the *freq2z()* function was used, where interpolation and FIR-filtering are used. The amount of elements contained in 2D-frequency response was set to 100 x 100. It is apparent that the bandwidths of filter kernels become smaller and are shifted to the areas of lower frequencies. But simultaneously the rotation variance occurs. This is visible in the repetitions of higher frequent components similar to magnitude peaks of sinc-function what arises due to the resulting discontinuities at filter kernel boundaries while performing of 2D-DFT.

The fact is that a decrease of filter bandwidth leads to a better noise robustness, but can cause an occurrence of rotation variance if the filter kernel elements at the boundaries are $\neq 0$. But especially for the present problem and in the case for angle information extraction algorithms the mentioned aspects do not play a role because of the following facts:

- No convolution is performed for direct curvature extraction. The field curvature is extracted globally.
- The curvature values extracted from $\mathbf{B_x}$ and $\mathbf{B_y}$ components for the LoG convolution approach are accumulated and not compared with each other in the result image. That means that no spatial analysis of the picture is performed afterwards.

In order to investigate robustness of the LoG approach against noise and enhancement of its stability by increasing of σ parameter the following test was carried out: The total sensor array size was set to 15 x 15 and the SNR to 60 dB. Spherical magnet was chosen as encoder magnet. The z-distance from the magnet was set to 2 mm. The angular error coming from the quantization noise gained from single LoG approach was compared with approximation approach over full encoder magnet rotation. The σ parameter was increased in 5 steps. The obtained results are shown in Figure 5.13.

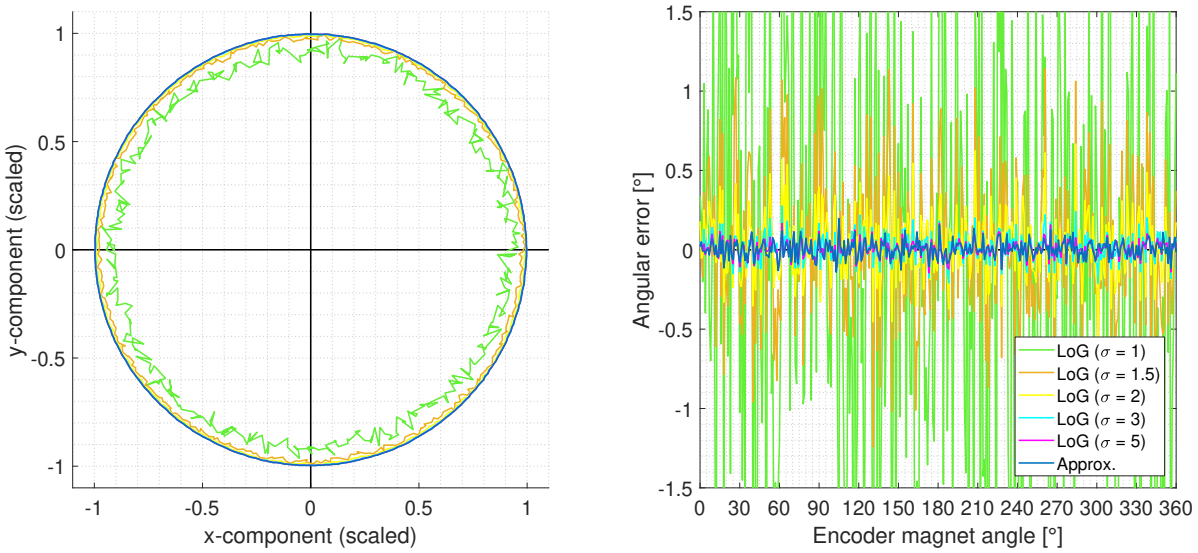


Figure 5.13: Optimization of LoG filter bandwidth. Left: xy-paths of LoG algorithm with different σ parameters and the xy-paths of approximation approach. Right: corresponding angular errors. By increase of σ parameters the quantization effect is drastically decreases and becomes very closely to the angular error of approximation algorithm. *Simulation setup: 15 x 15 array, 60 dB SNR, 2 mm z-distance, sphere encoder magnet.*

Conclusions: Given the obtained results, it's obvious that by an increase of σ parameter of LoG filter kernel the angular error due to the quantization noise is extremely reduced. The angular error of LoG comes extremely closely to the angular results obtained by approximation approach and is almost identical. This implies that there is a clear correspondence between the optimal solution of least squares for the chosen basic functions for the approximation approach and the minimal bandwidth of the LoG filter. With an excessive σ parameter of the LoG filter the same robustness as of approximation approach is obtained. Therefore, the LoG filter masks with the smallest bandwidths are used in all following investigations. Indeed, this means than the shape of the LoG filter is not approximated in this case. Thus, the obtained coefficients of the LoG function represents the coefficients for direct numerical extraction of the second derivate in all directions.
The optimized bandwidth of the LoG filter kernel is obtained, if the central element of a 3 x 3 mask becomes equal 1, or for larger mask sizes, if the elements at the corners of the LoG mask becomes equal 1. The optimized LoG filter kernel coefficients for the mask sizes 3 x 3 to 15 x 15 are included in Appendix F.2.

Important note: For computation of LoG filter mask assigned with excessive σ parameter an additional operation for subtraction of the mean value of the filter is required to make sure that the homogeneous parts is suppressed completely.

5.3.2 Exponent optimization for modified approximation approach

As described in Section 3.3.3 the prerequisite for the modified approximation approach for angle information extraction is the knowledge about the field weakening coefficients which depend on dimensional parameters of applied encoder magnet. For this reason a second important optimization was executed in order to determine the exponent for the field weakening denominator matrix for the optimal mathematical description of magnetic field of chosen disc magnet. The optimization of the exponent n was carried out for the full range of z-distances [2, 5] mm, sensor array size [3 x 3, 15 x 15] and the extraction mask size [3 x 3, 15 x 15]. The sweep range for the n exponent was set to [-5, 5], with a step size of 0.01. The optimization was performed for one angular position only due to ideal field constraints, at $\varphi = 0°$ and for ideal mechanical alignment constraints, naturally. Term for model of homogeneous stray field part were not included into the estimation of the basis functions during the optimization process because the interest is only in the model parameters.
The change of field components error in % within the full sweep range of the exponent n for an array size of 15 x 15 elements is shown in Figure 5.14.

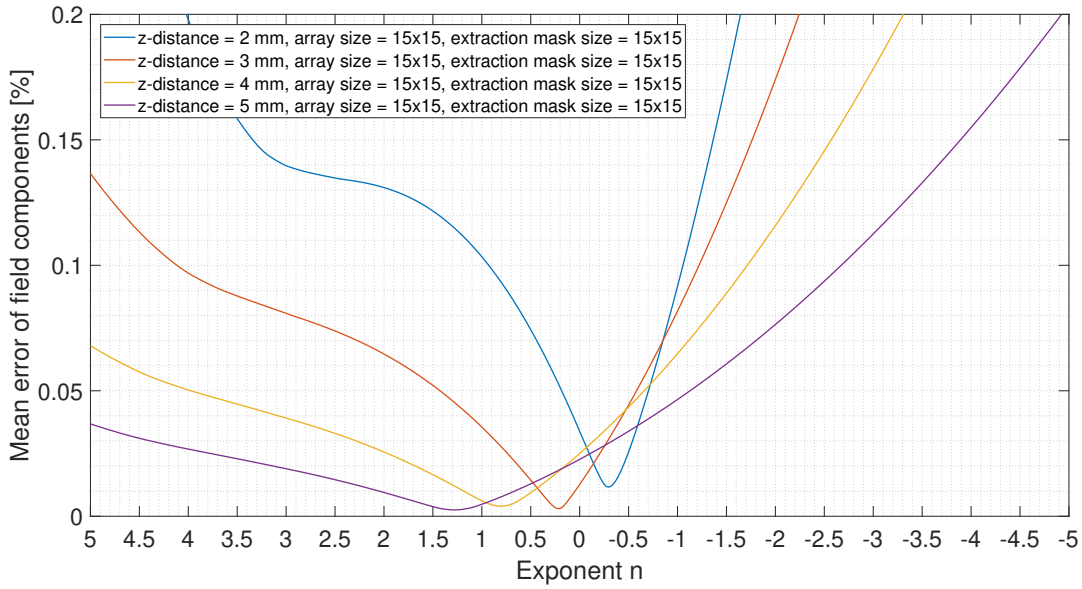


Figure 5.14: Optimization of exponent n for the disc magnet for z-distance 2 mm to 5 mm and array size of 15 x 15 elements. The remained error after the optimization of exponent n is below 0.05%.

An interesting observation is that the optimized exponent at the closest distance to the encoder magnet, at 2 mm, is negative. This is related to the far field and near field behavior of the disc magnet. The determined, optimized values of exponent n for the full range of z-distances [2, 5]

mm and sensor array sizes [3 x 3, 15 x 15] and remained field components errors are recorded in the Table 5.2.

z-distance [mm]	**Array size:**	3x3	5x5	7x7	9x9	11x11	13x13	15x15
2	*Exponent n*	3.31	-0.28	-0.28	-0.28	-0.29	-0.29	-0.29
	Error [%]	1.e-4	0.015	0.0140	0.0132	0.0126	0.0117	0.0116
3	*Exponent n*	4.08	0.22	0.22	0.22	0.22	0.21	0.22
	Error [%]	4e-5	0.004	0.0040	0.0035	0.0033	0.0030	0.0030
4	*Exponent n*	0.70	0.78	0.81	0.81	0.80	0.79	0.80
	Error [%]	2e-6	0.0058	0.0051	0.0045	0.0043	0.0040	0.0040
5	*Exponent n*	1.12	1.24	1.29	1.28	1.28	1.26	1.28
	Error [%]	5e-5	0.0037	0.0032	0.0028	0.0027	0.0025	0.0025

Table 5.2: Optimized exponents n for the disc magnet and remained MEFC in % depending on array size and z-distance used for modified approximation approach.

For comparison the error that remains after field approximation without modification by field weakening denominator matrix was also calculated with the same setup as described above. The results for the disc magnet are recorded in Table 5.3, for sphere magnet in Table 5.4.

z-distance [mm]	**Array size:**	3x3	5x5	7x7	9x9	11x11	13x13	15x15
2	*Error [%]*	2.3636	2.3064	2.1601	2.0711	2.0053	1.9342	1.9235
3	*Error [%]*	2.5125	2.4408	2.2933	2.2044	2.1386	2.0666	2.0560
4	*Error [%]*	2.2420	2.1806	2.0503	1.9709	1.9120	1.8480	1.8384
5	*Error [%]*	1.8783	1.8314	1.7235	1.6573	1.6082	1.5548	1.5468

Table 5.3: MEFC in % of field components of the disc magnet without modification of approximation depending on array size and z-distance.

z-distance [mm]	**Array size:**	3x3	5x5	7x7	9x9	11x11	13x13	15x15
2	*Error [%]*	8.0776	7.5579	7.0283	6.7065	6.4760	6.2461	6.2063
3	*Error [%]*	4.9966	4.7624	4.4525	4.2628	4.1243	3.9805	3.9582
4	*Error [%]*	3.3995	3.2755	3.0722	2.9473	2.8554	2.7580	2.7426
5	*Error [%]*	2.4651	2.3916	2.2477	2.1591	2.0936	2.0233	2.0125

Table 5.4: MEFC in % of field components of the sphere magnet without modification of approximation depending on array size and z-distance.

The transition from near to far field for the disc magnet becomes visible by calculated field components approximation error in Table 5.3. At z-distance of 3 mm the remained approximation error is maximum. In general, the extension of approximation term 3.22 by field weakening denominator matrix and optimization of its exponent shows extremely high approximation error reduction. According to Tables 5.3 and 5.2 the approximation for the disc magnet reduces

from max 2.5% to almost 0%. Notable is also the fact that in comparison to approximation of disc magnet field the sphere magnet field error is higher and is located within the range of approximately 2% to 8%.

For presentation of the optimizations further figures are shown in Appendix F.1.2, depending on z-distance, array size and extraction mask size.
According to the theory explained in 3.3.3 the exponent of field weakening denominator for a sphere magnet is 2.5. With this knowledge the optimization procedure was proved by optimizing the exponent of the field weakening denominator matrix for the sphere magnet. The correctness of the optimization script was confirmed, see Appendix F.1.1.

5.4 Preliminary investigations

The preliminary investigation documented in this section were carried out in order to determine possible algorithm weaknesses and include the algorithmic extensions before the start of main investigations. Furthermore, another goal of this investigations was to identify a possible extreme outliers against quantization in the algorithms group. Therefore, two short test were carried out: a sensitivity check against mechanical misalignment of the encoder magnet and quantization noise. For all tests an array size of 15 x 15 elements was chosen. Sphere magnet was used and the z-distance was set to 2 mm.

5.4.1 Misalignment test

Within the framework of this preliminary misalignment test a single tilt related to the x-axis as well as a mixed misalignment case consisting of a small tilt and x-/y-translation of the encoder magnet were applied, in order to check the misalignment robustness of algorithms and obtain an understading about its sensitivity. In this section the affect of the signal tilt on the 2D-DFT algorithm is introduced. See Appendix G.2 and G.3 for more discussion. A single x-tilt misalignment case of 0.5° was applied. The angular accuracy results of the algorithms are shown in Figure 5.15.
Indeed, all algorithm are not sensitive to the simple tilt variation of the encoder magnet, except of the 2D-DFT algorithm. The angular error caused by the applied tilt misalignment of the encoder magnet of only 0.5° amounts to a maximum angular error of approximately 1.5°. As a conclusion it follows that unfortunately, this standalone 2D-DFT algorithm is extremely sensitive against misalignment effects. It was also observed that the 2D-DFT algorithm do not exhibit a comparatively robustness to misalignment in the form of a x-/y-translation as all other candidates. Therefore, for the subsequent main investigations of the algorithms the 2D-DFT approach was coupled with the algorithmic extension called realignment of field components, also denoted as dynamic misalignment compensation, which is documented in Section 3.5.2. After this extension the 2D-DFT algorithm became much more robust against single tilt variation and exhibited also a much better robustness against mixed misalignment cases. For more information see additional investigation results in Appendix G.2.

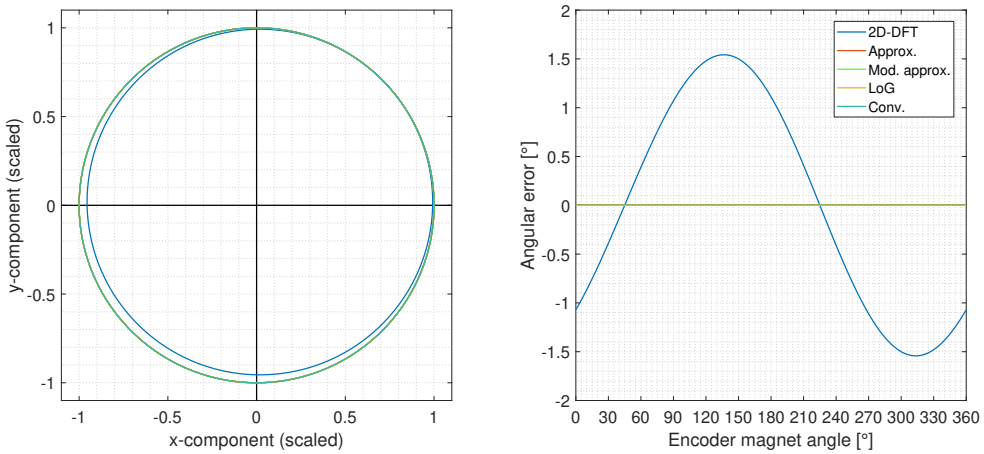


Figure 5.15: Preliminary investigation: misalignment test 1. Left: xy-paths of algorithm candidates. Right: corresponding angular errors. A tiny tilt addition of 0.5° causes an angular error of 2D-DFT algorithms of more than 1.5°. *Simulation setup: 15 x 15 array, 2 mm z-distance, 0.5° x-tilt, ADC not activated, sphere encoder magnet.*

5.4.2 Quantization noise sensitivity test

To investigate the sensitivity of algorithms against quantization noise and perform an early identification of possible outliers, a relative high ADC resolution of 14 bit was applied in this test. Ideal alignment conditions were set in the simulation. The impact of quantization noise on the angular accuracy of algorithms is shown in Figure 5.16.
The modified approximation approach represents an extreme outlier with resepct to the angular error caused by sampling of field components. In this applied case its maximum angular error due to quantization amounts to more than 2.5° while all other candidates shown a neglectable quantization error. The quantization robustness of the modified approximation algorithm becomes better, if the coefficients of the basis functions for the modelling of the homogeneous stray field component p_{x6} and p_{y6} are excluded from the estimation. The quantization error is this case is shown in Figure 5.17. But still, the quantization of this algorithm is relative high in comparison to other candidates. Furthermore, in this case the guaranty the stray field suppression won't be given at all.
Unfortunately, based on the result from conducted preliminary investigation of quantization noise effect it was decided to exclude the modified approximation algorithm from all further investigations completely[1]. Its practical feasibility leaves something to be desired due to its significant sensitivity against quantization. For further discussions see Appendix G.4.

[1]The modified approximation algorithm exhibited an extreme sensitivity against all variation parameters in the simulation, but especially, the impact of noise interference. It is presumed that this effect results from the division by the terms in the field weakening denominator matrix which is coupled to all coefficients of the magnet basis functions. Differences in the behavior were observable if the coefficients for the homogeneous part were excluded from the basis terms, as shown e.g. in Figure 5.17. This issue shall be definitely investigated in line with project continuation. For further information see documentation and discussion in Appendix G.4.

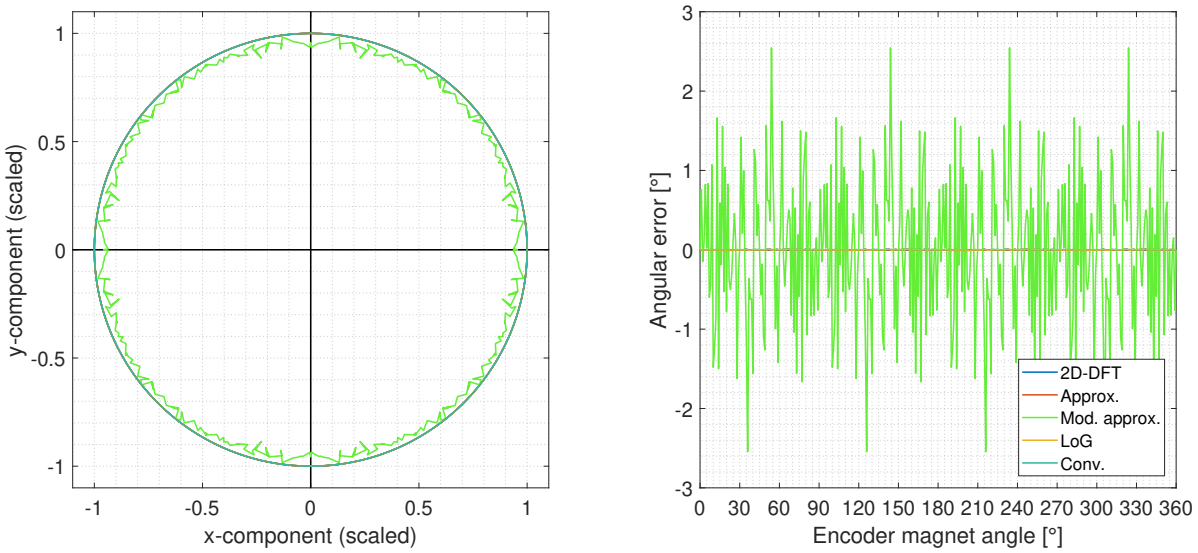


Figure 5.16: Preliminary investigation: quantization test 1. Left: xy-paths of algorithm candidates. Right: corresponding angular errors due to quantization. The modified approximation approach represents an extreme outlier in the whole group. All other candidates exhibits a marginal small quantization error which is not visible in the graph. In comparison to it a large maximum angular error due to quantization that is more than 2.5° is observable for the modified approximation approach. *Simulation setup: 15 x 15 array, 2 mm z-distance, ideal alignment, 14 bit ADC resolution, sphere encoder magnet.*

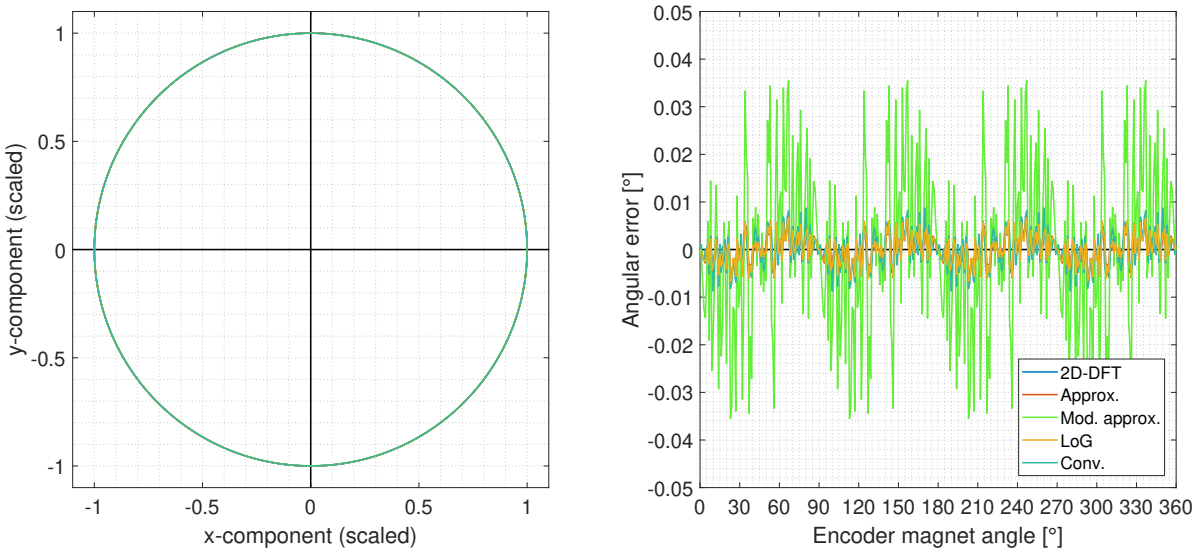


Figure 5.17: Preliminary investigation: quantization test 2. Left: xy-paths of algorithm candidates. Right: corresponding angular errors due to quantization. The angular error of modified approximation algorithm is around 0.035°. The maximum angular error of all other candidates is less than 0.01°. *Simulation setup: 15 x 15 array, 2 mm z-distance, ideal alignment, 14 bit ADC resolution, sphere encoder magnet.*

5.5 Applied algorithm conditions

To make the created test plan more understandable and provide better guidance through performed investigations all the applied conditions for the algorithm and their main parameter sweeps are summarized in this section.

For the angle information extraction all 5 developed algorithms described in Section 3.3 were considered. Their details and applied conditions are summarized as following:

1. **2D-DFT based algorithm including re-alignment of field components**
   In line with preliminary investigations it was proved that the developed 2D-DFT algorithm do not exhibit neither immunity against elementary encoder magnet tilt without any x-/y-translation nor a comparatively x-/y-translation robustness as all other candidates (see results and discussion in Appendix G.2). Due to this reason the 2D-DFT based algorithm was extended by dynamic misalignment compensation with reduced basis functions presented in Section 3.5.2. In the following, this algorithm is called *Extended 2D-DFT approach*.
2. **Approximation approach**
   The polynomial degree of chosen basis functions was set to 2, what represents the absolute minimum needed for the curvature extraction. Higher polynomial degrees were not considered in the context of this work.
3. **Direct curvature extraction with a single LoG mask**
   For the direct curvature extraction from the total array size the LoG masks with exceeded σ parameters, i.e. smallest bandwidths as possible were used. The coefficients are recorded in Appendix F.2.
4. **Convolution with LoG mask**
   The convolution mask size is kept always 2 values less than the total array size in order to achieve higher noise robustness. That means that with the convolution approach always 9 curvature values from $\mathbf{B_x}$ and $\mathbf{B_y}$ field components were extracted in total. Smallest bandwidths were used here as well.

For tracking algorithms and associated misalignment compensation directly in spatial domain by moving of extraction mask 5 algorithms in total were considered as well. The applied conditions for algorithm candidates are as follows:

1. **Application of Butterworth filter in spatial frequency domain**
   The filtering was applied without zero padding of the input signal. The filtering method including zero padding of the input signal exhibited a extremely well robustness against noise but its initial error was unfortunately too high and hence not considered as a suitable candidate for tracking algorithms.

2. **Approximation method**
   Also here the polynomial degree of chosen basis functions was set to 2. A higher approximation degree could probably increase the localization accuracy but was not included into the performed investigation.
3. **Simplified approximation method in combination with analysis**
   Applied conditions here are in full accordance to the description of this method in Section 3.6.4.
4. **Derivative of Gaussians method**
   The mask sizes of Derivative of Gaussian filter kernels in x- and y-directions was set to 7 x 7 and the σ parameter to 1 in order to make sure that neither a rotation variance nor ringing artifacts occur in the resulting image.
5. **Convolution with LoG mask**
   Also here the LoG-mask size was set to 7 x 7 with σ = 1, to avoid the rotation variance.
6. **Convolution with Prewitt gradient filter mask**
   The size of Prewitt filter mask was set to 5 x 5 in order to get a better noise stability in comparison to the minimum required 3 x 3 filter kernel.

For angular error correction algorithms associated with an angular teach-in the following conditions and parameter limits were set:

1. **Homography approach**
   The considered degrees of freedom are 6 and 8, with estimation of 6, 8 or 9 **H** matrix coefficients. The minimum amount of matching points starts with minimum required depending on DOF and is 3, 4 or 5. The maximum amount of matching points is 360, according to a full encoder magnet rotation.
2. **Nonlinear error correction based on geometric distortion cancelation**
   There were 3 polynomial degrees considered: 2, 4 and 6. The minimum amount of matching points starts with minimum required depending on degree: 6, 8 or 10. The maximum amount of matching points is set to 181, according to a half rotation of encoder magnet for each polynom. In total, the maximum amount of matching points needed for the estimation of all four rectification polynomials counts to 181x4.

For all algorithms where an estimation procedure is required the function *mldivide*[2] was used, which applies most appropriate solver depending to the problem formulation.

All the test were performed by usage of created evaluation software and additional test scripts, coupled on the main program.

[2] \ operator: Solver for systems of linear equations. Takes advantages of the symmetries in the matrices. Function aims to minimize the computation time. https://de.mathworks.com/help/matlab/ref/mldivide.html

5.6 Investigation plan

For the tests of developed algorithms applied on in the simulation built 1:1 replicated encoder magnet - sensor array system, an extensive test plan based on corner cases within the limits of safe operating area was created. The created investigation plan consists of 5 blocks in total. In each block, the results and observation are discussed and based on it a knowledge related on particular practice relevant questions is gained.
Firstly, a deep proof of concept test in block 1 is carried out. The knowledge gain obtained from test results in this block gives most significant information about the feasibility and applicability of developed algorithms on a real sensor array system. The knowledge about within which technological limits for electronic and signal quality parameters it becomes possible to rely on the concept for field curvature extraction is obtained by performed tests in this block. In the second block the robustness of algorithms against mechanical misalignment effects of encoder magnet is investigated and the their acceptable sensitiveness limits regarding required angular accuracy determined. A corner simulation within parameter space of the SOA is performed in this investigation. In the third block the localization accuracy of developed tracking algorithms is investigated. Their performance is compared in detail and the best candidate for this algorithmic extension for teach-in application in spatial domain is proposed. Subsequently, the combination of angle information extraction algorithms passed the blocks 1 and 2 are combined with the tracking algorithm in block 4. Their combinational performance, associated angular accuracy enhancement and the absolute stability limits are found out in this block. As the last investigation, the developed algorithms for angular error correction are examined. Their performance and behavior depending on degrees of freedom / polynomial degree and amount of matching points is inspected. In the final step, based on gained knowledge from the performed test, a recapitulation and discussion of results takes place and the best suitable algorithm candidates are proposed for the application and implementation on the ASIC.

Remarks:
To decrease the documentation effort the test in blocks 2 and 4 were carried out for the sphere magnet only. It is expected that the algorithm in these blocks applied on disc magnet data behave analogically.
Wherever the SNR parameter is swept, a sampling of values with the ADC does not take place because the possible quantization noise is already represented in this parameter sweep and, in case of investigation in block 3 and 4, not implicitly needed. The amount of SNR repetitions for each SNR sweep step in all test cases where the behavior of algorithms depending on this parameter is investigated, was initially set to 100, in accordance to conclusion gained in Section 4.2.4. All tests in exception of test 1.3 in block 1 *proof of concept test* the maximal arithmetic accuracy (double precision) was used.

The total investigation plan is illustrated in Figure 5.18.

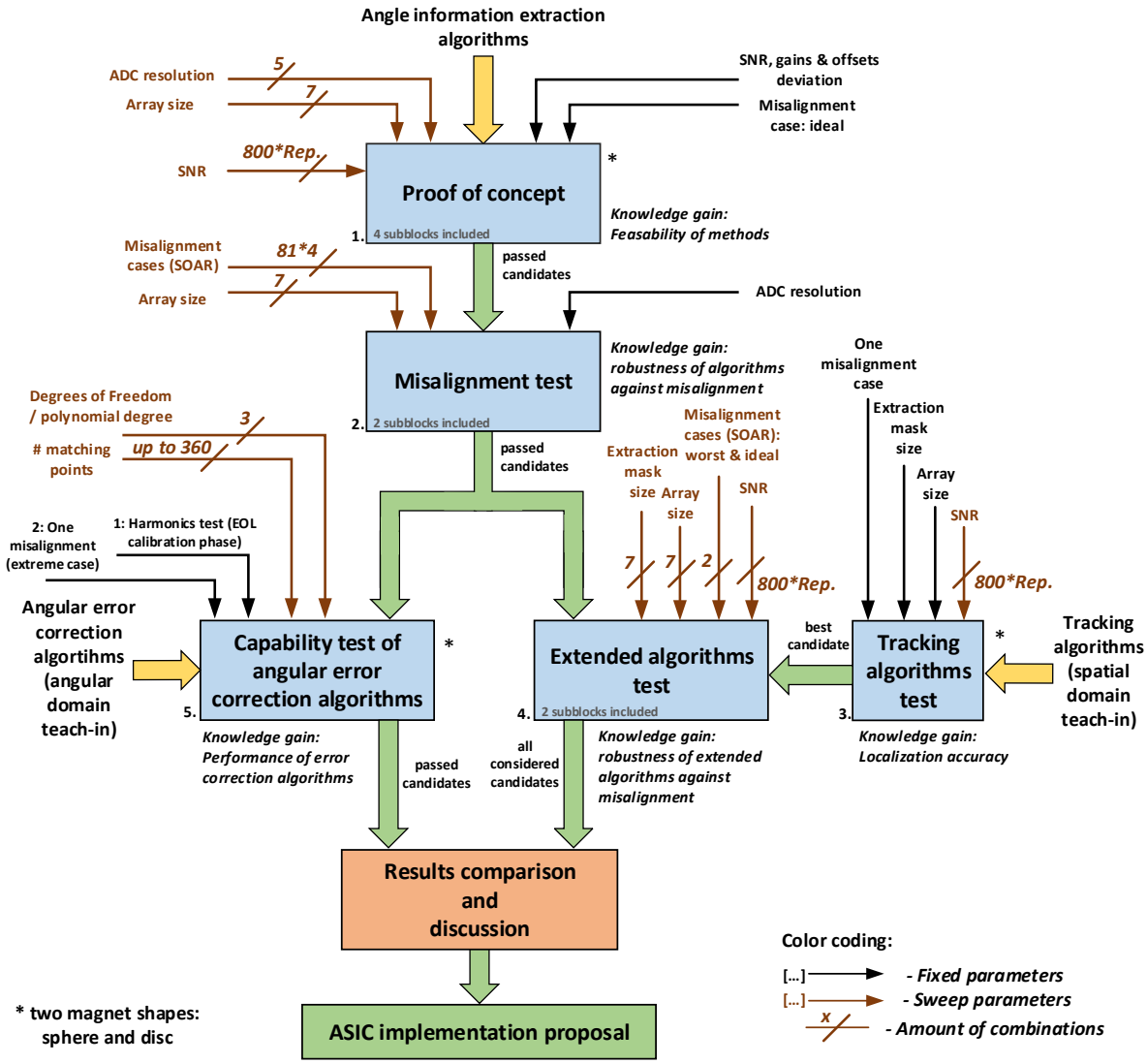


Figure 5.18: Investigation plan: created total test plan for execution of investigations of developed algorithms. The test plan consists of 5 main blocks where particular practice relevant questions are answered. Blocks 1, 2 and 4 include multiple subordinated subblocks. In all blocks the test parameters are color coded. The main algorithms for angle information extraction are passing through all test blocks from top to the bottom, extended by angular error correction algorithms from left and right. At the end the obtained results as well as algorithm performance are discussed and the proposal for implementation of algorithms is made.

5.6.1 Content of block 1

The first block represents the proof of concept and consists of 4 subblocks in total. The content of block 1 is illustrated in Figure 5.19.

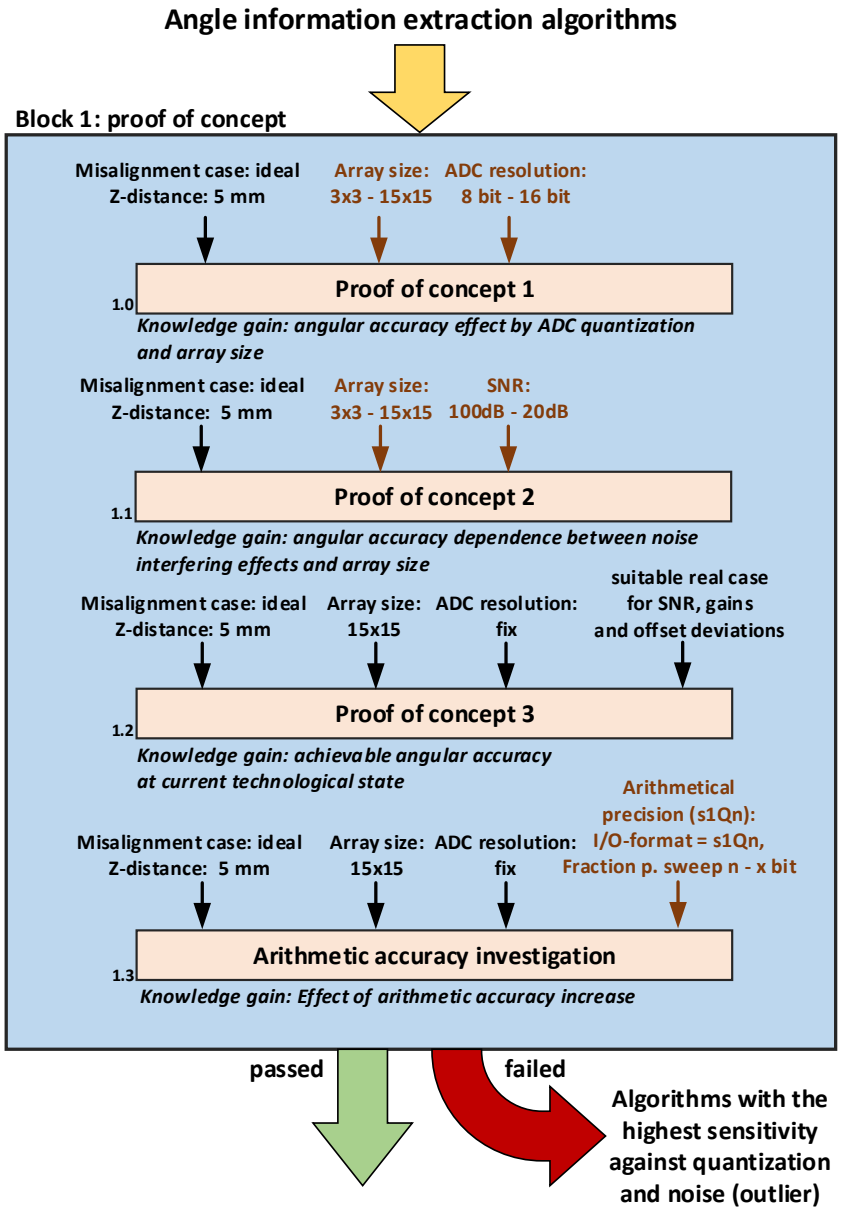


Figure 5.19: Investigation plan: Content of block 1.

The following questions are answered in the context of proof of concept test:

1. What is the necessary amount of sensor points within the sensor die area of 2 mm x 2 mm in combination with an ADC resolution what fulfill the angular accuracy requirement of $\varphi_{err} \leq 1^{\circ}$? Is it even possible to achieve the required accuracy in the assumed range of technological parameters and for an encoder magnet system which is being used in the series? How does the tendency of angular accuracy improvement depending on amount of sensor pixels and ADC resolution look like? This questions are answered in subblock 1.0 *Proof of concept 1*.
2. Of course the quality of signals which are propagated to the electronic acquisition units itself plays the most important role. Where are the signal quality limits measured as SNR for required angular accuracy? Which angular accuracy can be expected if the physical signal quality is known? Subblock 1.1 *Proof of concept 2* gives the answers to these questions.
3. Is it possible to achieve the required angular accuracy at current technological state with a realistic assumption for SNR, gain and offset deviations of OPs? Are the pre-calibrations of each sensor array element for compensation of electronic tolerances indeed needed or do the gain and offsets deviation play a subordinate role, if a sensor array is used? What is the maximal possible angular accuracy within investigated limits? These questions are answered in the subblock 1.2 *Proof of concept 3*.
4. Within the framework of further implementation of algorithms the arithmetic accuracy for computation operations associated with limitation of register bit widths in the arithmetic units[3] will surely play an important role for the angular accuracy. The question in what tendency an increase of registers for arithmetic computations affect the angular accuracy and what is the minimum bit width for arithmetic operations needed for adherence to the angular accuracy requirement $\varphi_{err} \leq 1^{\circ}$ is answered in subblock 1.3 *Arithmetic accuracy investigation*.

Algorithm candidates which represent outliers in the angular accuracy and exhibit a higher sensitivity during the parameter sweeps in comparison to other considered candidates within the total group do not pass the tests. They are sorted out and are not propagated to further test blocks.

[3] Mainly the limitation of register bit widths for multiplication operations is crucial. Refer to Chapter 4, Section 4.2.4 Relevant functions for investigations: Fixed-point arithmetic.

5.6.2 Content of block 2

In Figure 5.20 the subblocks included in block 2 are illustrated. This block covers the test where the algorithms robustness against mechanical misalignment of encoder magnet is examined. The questions about maximum acceptable misalignment limits and sensitivity of sensor array system against misalignment effects are answered in this block.

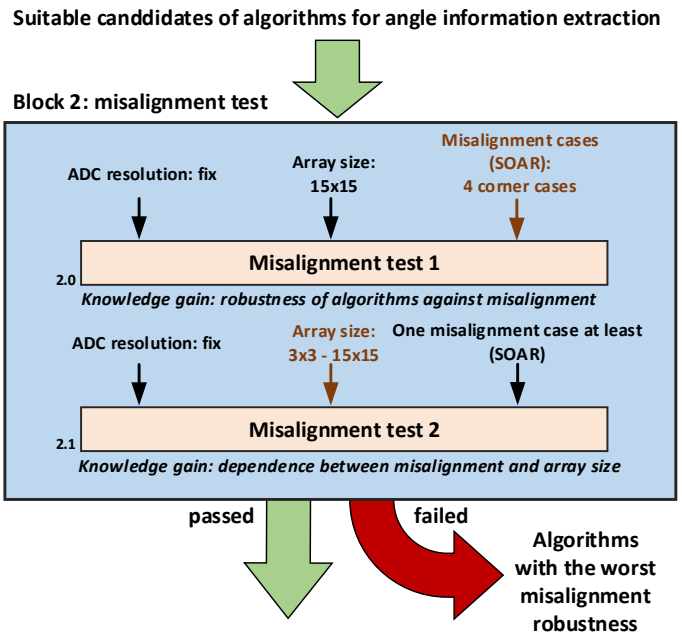


Figure 5.20: Investigation plan: Content of block 2.

In the first subblock 2.0 *Misalignment test 1* the robustness of algorithm candidates that passed the proof of concept test is investigated. Acceptable angular accuracy limits are found out in this block. The misalignment parameter x-translation, y-translation and x-tilt are swept as 4 corner x-/y-translation planes where for each plane the maximal misalignment parameters for z-distance and x-tilt are set. The ADC resolution is set to an appropriate value in order to keep the quantization error small enough. The array size here is set to maximum because its expected, that the angular error caused by misalignment will not depend on variation on array size significantly. In order to investigate the dependence between misalignment and variation of sensor array size a test for couple of random chosen misalignment cases within the SOA is carried out in subblock 2.1 *Misalignment test 2*.
Algorithm candidates which represent the highest sensitivity against misalignment effects in comparison to other considered candidates within the total group are sorted out as well and do not pass to the subsequent test in the next block.

5.6.3 Content of block 4

In the tests contained in block 4 all the suitable candidates for angle information extraction that passed the misalignment test in block 2 are combined by tracking algorithms for localization of AMM points. The investigations carried out in this block give a knowledge about the maximum possible increase of angular accuracy in combination with teach-in phase in spatial domain. In addition to it, the stability of this algorithmic combination against noise is examined, in order to get a knowledge about the dependence between SNR and extraction mask size and find out the signal quality limits needed for this algorithmic extension.

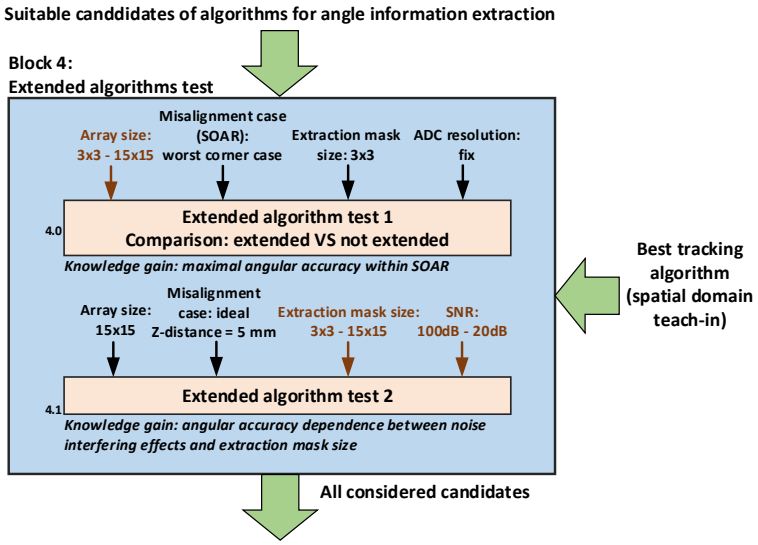


Figure 5.21: Investigation plan: Content of block 4.

In the first subblock 4.0 *Extended algorithms test 1* The extraction mask is set to absolute minimum and the array size is swept to maximum. The worst corner case for the misalignment is applied here, which is found out in the *Misalignment test 1* in the block 2. A direct comparison to the non extended algorithms is done in this block and the maximal possible angular error decrease by movement of extraction mask within sensor array area is determined. The angular error decrease tendency which results from the increase of total array size will give an important knowledge about the potential of this algorithmic extension. As well as in block 2.0 an appropriate ADC resolution is set to hold the quantization error at minimum. In the second subblock 4.1 *Extended algorithms test 2* the sensor array size is set to maximum and the extraction mask size as well as SNR are varied within their full ranges. Ideal misalignment case is applied here in order to keep the initial angular error at 0°.
No candidate can fail through this test because their robustness is already proved until this investigation.

5.7 Investigation results

All the performed investigation in full accordance to the test plan illustrated in Figure 5.18 are documented and discussed in this section.

5.7.1 Proof of concept

In context of proof of concept and applicability investigations of the algorithms 4 tests were carried out in total in correspondence with 5.19.

Sensitivity against quantization noise (subblock 1.0)

In this test the sensitivity of all 5 algorithm candidates against quantization noise caused by ADC sampling is examined. The goal of this test is to find the combinations of absolute minimum required amount of sensors with an ADC resolution for both encoder magnets which are required to fulfill the angular accuracy requirement of $\varphi_{err} \leq 1^{\circ}$.
The total amount of sensor elements on a die area of 2 mm x 2 mm was varied within the limits of 3 x 3 to 15 x 15 elements and the ADC resolution from 8 to 16 bit. Ideal misalignment condition are applied for this test. Since the absolute limit within the SOA limitations needs to be found in this test, only the z-distance of 5 mm needs to be set as the distance condition. Nevertheless, identical test for the z-distance of 2 mm was carried out as well. The summary of most significant combinations for both encoder magnets are recorded in Table 5.5.The full Tables with maximum and mean angular errors for both magnet shapes and z-distances 2 and 5 mm are included in Appendix G.5.1.

ADC	**Magnet shape:**	**Sphere**						**Disc**					
[bit]	**Array size:**	3x3		5x5		7x7		3x3		5x5		7x7	
	Error [°]:	max	mean	max	mean	max	mean	max	mean	max	mean	max	mean
12	*Ext. 2D-DFT*	1.674	0.554	1.281	0.398	0.849	0.261	1.524	0.467	1.065	0.400	0.763	0.272
	Approx.	1.674	0.554	1.203	0.384	0.712	0.273	1.524	0.467	1.068	0.397	0.739	0.270
	LoG	1.673	0.554	1.201	0.383	0.717	0.273	1.522	0.467	1.071	0.398	0.743	0.269
	Conv.	1.673	0.554	1.600	0.474	1.083	0.361	1.522	0.467	1.560	0.567	0.941	0.301
14	*Ext. 2D-DFT*	0.386	0.125	0.288	0.098	0.234	0.063	0.292	0.106	0.282	0.075	0.196	0.055
	Approx.	0.386	0.125	0.289	0.098	0.249	0.064	0.292	0.106	0.275	0.076	0.191	0.051
	LoG	0.386	0.125	0.288	0.098	0.248	0.063	0.291	0.106	0.275	0.076	0.190	0.050
	Conv.	0.386	0.125	0.412	0.157	0.249	0.087	0.291	0.106	0.266	0.109	0.223	0.073

Table 5.5: Proof of concept results. Sensitivity against quantization noise. Maximum and mean angular errors caused by quantization at z-distance = 5 mm.

From the results shown in Table 5.5 two statements are obtained:

- For the approaches extended 2D-DFT, Approximation, single LoG and Convolution with the LoG mask with an applied ADC resolution of 10 bit at least 7 x 7 sensor elements are necessary to obtain the maximum angular error of $\leq 1^{\circ}$. With an ADC resolution of 14

bit the smallest array size of 3 x 3 elements within the sensor die area of 2 mm x 2 mm would be sufficient.

- Both sphere magnet and disc magnet encoder system show the same limits for the fulfillment of angular accuracy requirement $\varphi_{err} \leq 1^{\circ}$

In Figure 5.22 the impact of quantization noise on the xy-paths the angular accuracy of all algorithms is depicted. For the applied simulation setup shown in Figure 5.22 the quantization effects are clearly visible. While the extended 2D-DFT approach, LoG, convolution (slightly different deviation only) and approximation algorithms exhibit a quite similar quantization noise what leads to a maximum angular error around $\pm$ 0.6°.

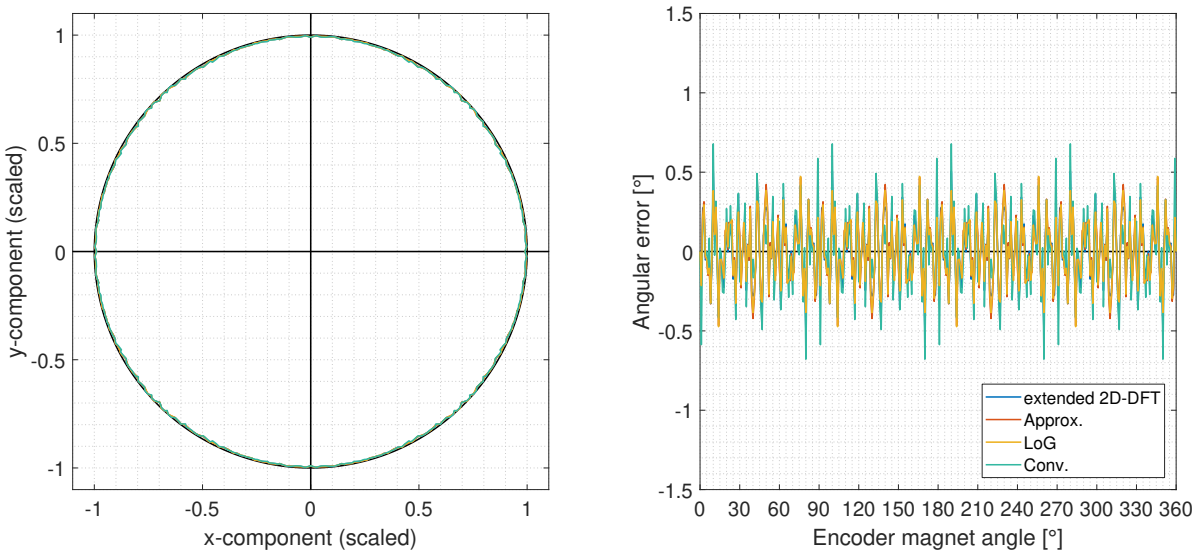


Figure 5.22: Proof of concept investigation: Effects of quantization noise. Left: xy-paths of algorithm candidates. Right: corresponding angular errors. *Simulation setup: 15 x 15 array, 5 mm z-distance, 12 bit ADC resolution, sphere encoder magnet.*

Gained knowledge:
The most important knowledge obtained from this investigation is the fact, that it is indeed practically possible to rely on the approach of the field curvature extraction for the angular error calculation and suppression of homogeneous part for guaranty of stray field suppression requirement. Within set technological limits the developed concept is practically usable. It is possible to use both encoder magnet systems in this concept. The commercially used disc encoder magnet system exhibit a field inhomogeneity which is sufficient to make the field curvature based angle calculation practically feasible.

Conclusions:

- According to the results in G.5.1 it is not possible to obtain an angular accuracy $\leq 1^{\circ}$ with an ADC resolution of < 10 bit within the investigated array size limits at all.

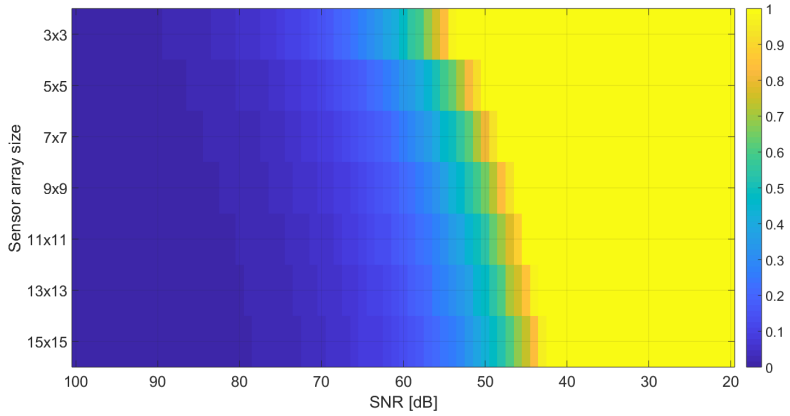
- With an ADC resolution of 12 bit at least 7 x 7 sensor elements are necessary to obtain an angular error due to quantization smaller than 1°. Based on the tendency taken from Tables G.5.1 a 6x6 sensor array would most probably also fulfill the required criteria for maximum angular error $\varphi_{err} \leq 1°$.
- With an ADC resolution of $\geq$ 14 bit the developed concepts will also work with a sensor array of 3 x 3 elements.

The determined conclusions above are valid for both magnet shapes.

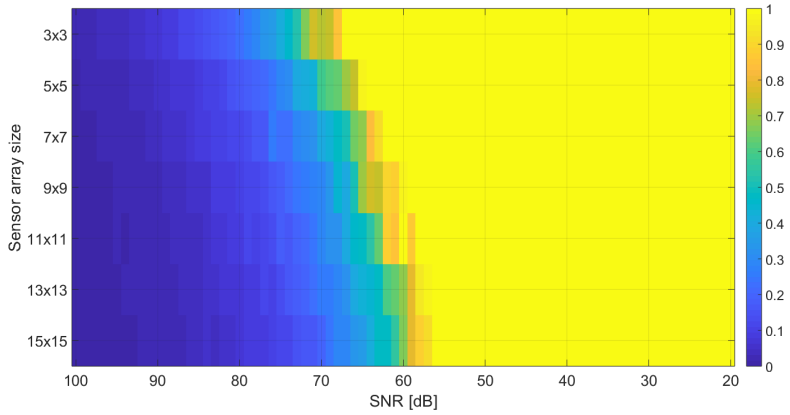
Robustness against signal-to-noise ratio variation (subblock 1.1)

In this test the impact on angular accuracy of algorithms by additive white Gaussian noise is investigated. The first goal is to get the information of minimum necessary signal quality measured in SNR before the propagation of signals to the electronic data acquisition units in order to achieve the angular accuracy requirement of $\varphi_{err} \leq 1°$. The second goal is to see the tendency of noise robustness of the algorithms in dependence of the array size.

Also here all algorithms exhibit a similar performance. In Figure 5.23 the maximum and mean angular error borders in dependece of SNR for the approximation algorithm are show. The figures from other candidates are included in Appendix G.5.2. From Figure 5.23 it can be taken that the SNR value for the required angular accuracy is located between 60 and 70 dB. Notable fact is also that the mean angular error 1 ° border is between 45 and 55 dB. The tendency of required SNR to adhere to the angular error criteria depending on the array size is in both figures almost linear.



(a)



(b)

Figure 5.23: Investigated impact of noise on the angular accuracy of algorithms. Results of approximation algorithm. (a) Mean error depending on SNR and array size. (b) Maximum error depending on SNR and array size size. *Note: the maximum limit of color was set to 1 in order to see the* $\varphi_{err} \leq 1^\circ$ *boundary clearly. Simulation setup: 15 x 15 array, ideal alignment, z-distance 5 mm, sphere encoder magnet.*

Additional bar plots were created so that the angular error behind the $\varphi_{err} \leq 1^\circ$ limit can also be evaluated and included in Appendix G.5.2 as well. In Table 5.6 the determined limits of SNR for the angular accuracy requirement of $\varphi_{err} \leq 1^\circ$ are recorded.

In Table 5.18 the determined limits of SNR for the angular accuracy requirement of $\varphi_{err} \leq 1^{\circ}$ are represented.

	Array size:	3x3	5x5	7x7	9x9	11x11	13x13	15x15
$\approx$ SNR for 1° angular error limit [dB]	*Mean error*	56	52	50	48	47	46	45
	Max. error	70	67	65	63	61	60	59

Table 5.6: Limits for an angular accuracy $\varphi_{err} \leq 1^{\circ}$ for the algorithms in dependence of the array size. Results of approximation algorithm.

Conclusions: To be able to rely on this concept from the physical point of view, the designers and process engineers have to make sure that the quality of the signals propagated into the electronics exhibits at least a signal-to-noise ratio of 60 - 70 dB. Also here it needs to be mentioned that this values are practically possible.

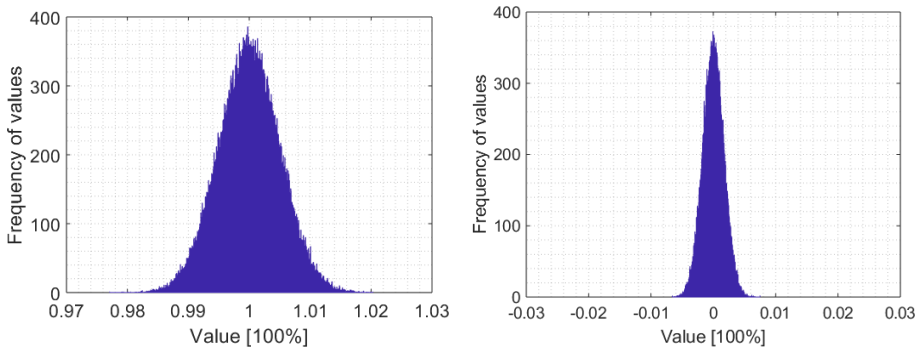
Gain and offset deviations (subblock 1.2)

The aim of the following investigation is to get to know whether the pre-calibrations of each sensor array element are indeed substantial needed or do the gain and do the real-case based deviation of gains and offsets of OPs have an unimportant impact on the system accuracy.
The test setup and the sweep parameters are set in analogy to the test in Section 5.7.1. Only 10 repetitions of gain and offset variations including averaging of results were carried out due to time-consuming simulation duration. According to obtained reesults this amount of repetitions was sufficient.It is expected that in near future the performance of newest TMR sensors will be very close the the performance of current AMR sensors. Based on this expectation the values for RMS level of noise was taken from [20, Table 7]. Calculated by Equation D.2 the SNR value amounts to approximately 65 dB. Suitable values for gain mismatches and offset variations were applied, for this assumption it is referred to [68]. The applied gain and offset deviations are recorded in Table 5.7.

Characteristics	**Deviations**	**corresponding σ value**
Gain deviations	$\pm$ 3 %	0.5 %
Offset deviations	$\pm$ 1 % $\widehat{=}$ $\pm$ 600 kA/m	$\pm$ 0.1667 % $\widehat{=}$ $\pm$ 100 A/m

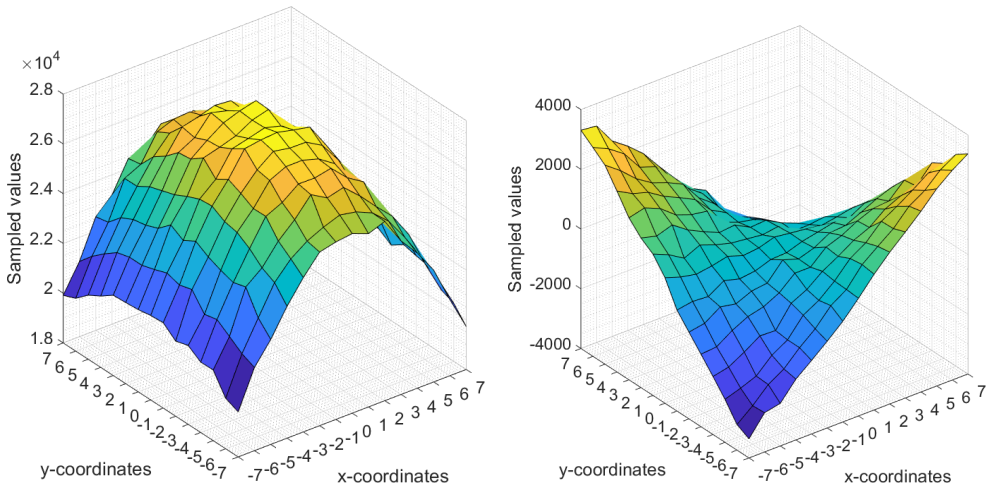
Table 5.7: Applied limits for gain and offset deviations for the proof of concept test.

The histograms of applied deviations are illustrated 5.24. The impact of gain and offset deviations of fictive sensor electronics elements on the measured field components are shown in Figure 5.25.



(a) Distribution of gains (b) Distribution of offsets

Figure 5.24: Histogram of distribution of applied gains and offsets within set limits (100000 values in total and 100 bins).



(a) sampled $\mathbf{B_x}$ components (b) sampled $\mathbf{B_y}$ components

Figure 5.25: Impact of gain and offset deviations of fictive sensor electronics elements on the measured field components at encoder magnet angle = 0°. *Simulation setup: 15 x 15 array, 2 mm z-distance, 16 bit ADC resolution, sphere encoder magnet.*

The impact of gain and offset deviations on resulting the xy-paths and calculated angle is shown in Figure 5.26.

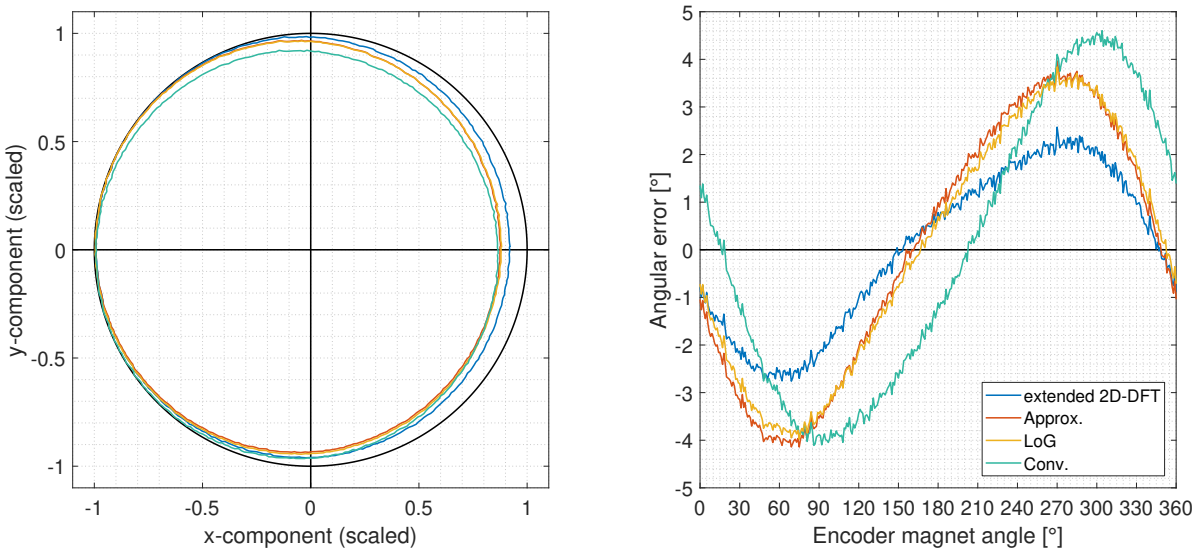


Figure 5.26: Effects of interfering of sensor signals by white Gaussian noise. Left: xy-paths of algorithm candidates. Right: corresponding angular errors. *Simulation setup: 15 x 15 array, 5 mm z-distance, 16 bit ADC resolution, sphere encoder magnet.*

The effects of gain mismatches and offset deviations are clearly visible. Although there is no mechanical misalignment of encoder magnet applied in the simulation a progression of angular error is observable. Results of approximation and LoG are close to each other. Convolution and extended 2D-DFT behave differently.
The obtained results for caused maximum and mean angular error by gain and offset deviations for both magnet encoders at SOA maximum z-distance of 5 mm are recorded in Table G.5.3. From the results contained in the Table G.5.3 it follows that the minimum angular error which can be obtained at an array size of 15 x 15 elements and an ADC resolution of 16 bit resolution is around 8° for all algorithm candidates and is significantly above the angular accuracy requirement.

Conclusions: With the assumed realistic expectations for the gain and offset deviations it has been proved that within set array size limits it is absolutely not possible to obtain the desired angular accuracy of $\varphi_{err} \leq 1°$. The gain mismatches and offset deviations are extremely critical for angular accuracy of developed algorithms and need in any case to be canceled during the calibration phase at the end of line. The angular error correction algorithms developed in Section 3.7 and tested in Section 5.7.5 shall be used for this purpose.

Arithmetic accuracy investigation (subblock 1.3)

Within the project the s1Q10 fixed point data format was used as an appropriate format for the digital representation of values so far. This format was chosen more or less pragmatically and in accordance to used ADC resolution of the applied μC-boards. Due to this, this test was performed by application of a 12 bit ADC resolution.
The first main question which is answered in this test is: Is it necessary to use wider registers than the minimum required bit width in accordance to the initial format resulting from the specification of the ADC resolution[4] for the internal calculations of the angle information extracted from the sensor array in order to fulfill the angular accuracy requirement? If yes, how is the tendency of the angular accuracy improvement is associated with it? The first question leads to another question which is related to the implementation of the arithmetic operation: Is it possible to optimize the computation operation process so that smaller register width can be used for the achievement of the angular accuracy specification?
In the tests the bit width for the fractional part of the smQn format is varied, which is responsible for main information and angular accuracy. In accordance to the set questions, two investigations are performed in this section. In the first investigation a possible enhancement of accuracy by enlargement of register for arithmetic operations by use of two upscaling operation is performed. The second investigation implies only one upscaling operation. In both tests the width of register for the fractional part n is varied from 10 to 20. The scaling operations are needed in order to do decrease the information loss after the complete arithmetic operations. The required scaling operations can be performed in the hardware using bit shift operations. In the tests the scaling were performed by multiplication of values with determined scaling factors.
The z-distance was set in this test to 5 mm in order to keep the distance at the maximum level in accordance to SOA. As a magnetic encoder the sphere magnet was chosen. The array size was set to 15 x 15 elements. The fractional part was varied for both tests from 10 to 20 bits in 1 bit step. Only the single LoG approach was applied in this test. All the arithmetic operations associated with the fractional part variation are performed by use of embedded.fi-class methods.

In the context of the first investigation the field components from the simulation are processed as follows:

1. Sampling of $\mathbf{B_x}$ and $\mathbf{B_y}$ values by an ADC of 12 bit resolution.
2. Formatting of all sampled $\mathbf{B_x}$ and $\mathbf{B_y}$ values to s1Q10-format.
3. Formatting of LoG filter mask coefficients to s1Qn-format.
4. **Multiplication operation:**
    a) Element-wise multiplication of $\mathbf{B_x}$ and $\mathbf{B_y}$ values with the LoG mask coefficients.

[4] For the ADC resolution of 12 bit $\Rightarrow$ s1Q10 format ($m = 1$, $n = 10$) the initial register width for the storage of the multiplication result starts from 23 bits ($(m + n) \cdot 2 + 1$), and for addition result from 13 bits ($m + n + 2$). See more in Chapter 4, Subsection 4.2.4 Relevant functions for investigations: Fixed-point arithmetic.

b) Upscaling to maximum limits [-2,2) using common scaling factor[5].

c) Casting of each obtained element in $\mathbf{B_x}$ and $\mathbf{B_y}$ to the s1Qn-format in order to come back to the set format after the multiplication steps.

5. **Accumulation operation:**

a) Calculation of x- and y-signal: summation of corresp. values in an accumulator[6].

b) Scaling back of the x- and y-signal to maximum limits [-2,2) using the common scaling factor[7].

c) Casting of x- and y-signal to the s1Q10-format in order to come back to the initial format after the multiplication and accumulation steps.

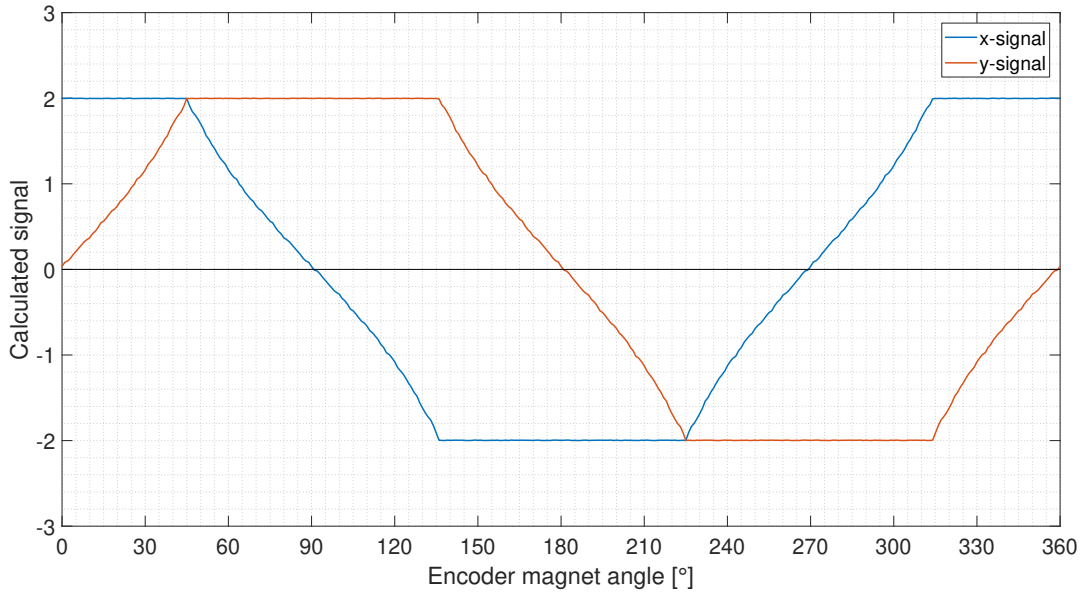
6. Angle calculation.

The first optimization operation is done at the step 4 c). Here the values are upscaled to the maximum possible values of ± 2 in order to preserve the accuracy after the following casting operation to s1Qn in the final multiplier operation step. In the step 5 a) the values need to be accumulated for the curvature extraction and calculation of the x- and y-signal for the final angle calculation. Before the angle calculation the x- and y-signal values need to be scaling back to to maximum range ± 2. The back scaled values are propagated to the arctangent function module which would be dimensioned to the same bit width as the initial format, so 12 bit for this setup.
Figure 5.27 shows the progression of the x- and y- signal and the angular error which results from the performing of whole arithmetic operations chain by use of the fractional bit width of 10 bits, according to the initial s1Q10 format. Impact of insufficient bit width for the arithmetic operations is visible. The maximum angular error amounts in this case to approximately 1.8°. The performed arithmetic operations by use of initial s1Q10 format exhibit a unsatisfied accuracy. The useful information after the multiplication by the LoG coefficients is not covered fully.
In Figure 5.28 the reduction of maximum and minimum angular error in dependence of the variation of the fractional part is depicted. The values of maximum and mean errors are shown in the Table 5.8. The angular error decreases rapidly at the beginning of the variation of the bit width for the fractional part n. It becomes evident, that for this setup, by sampling of the values by a 12 bit ADC at 5 mm distance from the ender magnet, the minimum required bit width for the fractional part is 12 bit. The maximum angular error by use of this format for arithmetic operations amounts to 0.82° and the mean angular error to 0.29°.

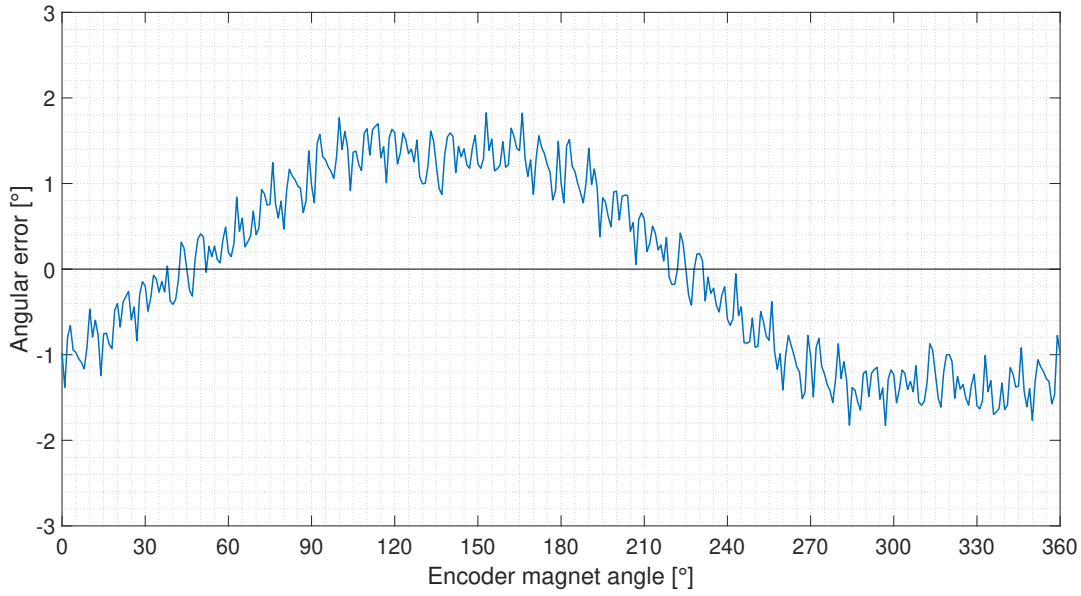
[5] This operation implies a division of all sampled components by the common maximum value from both data arrays $max(|\mathbf{B_x}|, |\mathbf{B_y}|)$ and a multiplication by 2.

[6] Accumulator register has the format s9Q10 in order to be capable to store the maximum sum: $2 \cdot 15^2$. Fractional part remains the same as from the initial format, but can be dimensioned smaller.

[7] This operation implies a division of the extracted x- and y-signal by the common maximum value $max(|x|, |y|)$ and multiplication by 2.



(a) Calculated x- and y-signal



(b) Angular error

Figure 5.27: Impact of insufficient bit width for arithmetic operations using minimum required bit widths resulting from initial s1Q10 format. (a) The upscaled extracted x- and/y-signal. Arithmetic inaccuracies are visible. (b) Calculated angular error. A quantization as well as a floating effect of not sufficient bit width is visible. The error is floating between $\pm$ 2°. *Simulation setup: 15 x 15 array, 5 mm z-distance, 12 bit ADC resolution, sphere encoder magnet.*

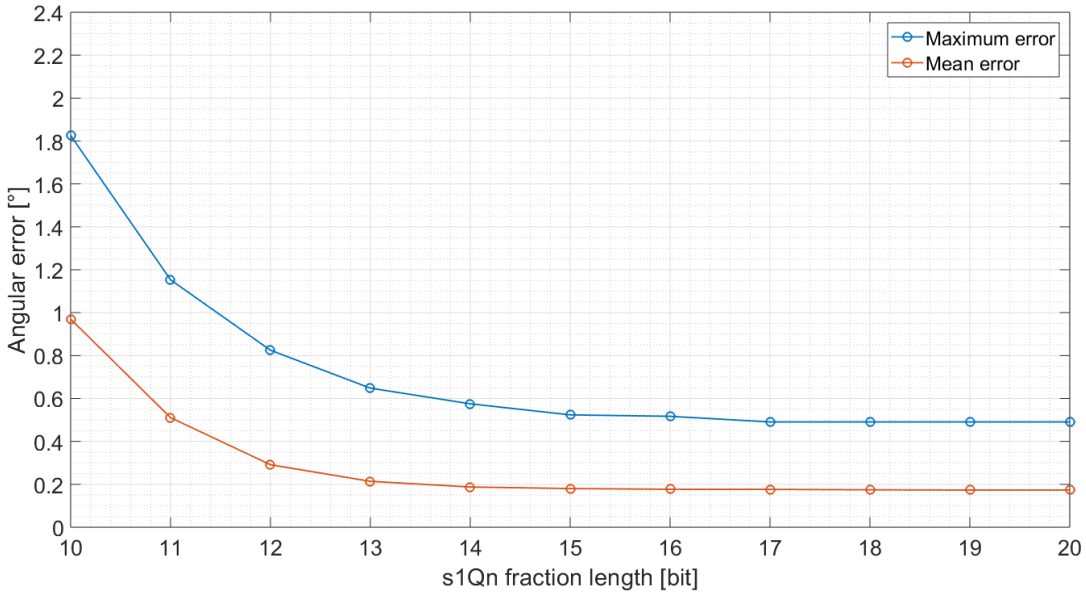


Figure 5.28: Increase of angular accuracy by enlarging of the bit width for the fractional part of arithmetic operations registers: Maximum and mean angular errors (test 1). A rapid increase of angular accuracy is observable by increase of fractional part from 10 to 12 bit. Angular accuracy requirement of $\varphi_{err} \leq 1^{\circ}$ is met at a fractional part bit width of 12 bit. Saturation is reached by a fractional part width of 14 bit. *Simulation setup: 15 x 15 array, 5 mm z-distance, 12 bit ADC resolution, sphere encoder magnet.*

Fraction length [bit]:	10	11	12	13	14	15	16	17	18	19	20
Maximum angular error [°]	1.82	1.15	0.82	0.64	0.57	0.52	0.51	0.49	0.49	0.49	0.49
Mean angular error [°]	0.97	0.51	0.29	0.21	0.18	0.18	0.17	0.17	0.17	0.17	0.17

Table 5.8: Maximum and mean angular errors depending on the register width for the fractional part of applied s1Qn fixed point format.

In the context of the second investigation the processing of data looks as follows:

1. Sample the $\mathbf{B_x}$ and $\mathbf{B_y}$ values by an ADC of 12 bit resolution.
2. Formatting of all sampled $\mathbf{B_x}$ and $\mathbf{B_y}$ values to s1Q10 format.
3. Formatting of LoG filter mask coefficients to s1Qn-format.
4. **Multiplication operation:**
    a) Element-wise multiplication of $\mathbf{B_x}$ and $\mathbf{B_y}$ values with the LoG mask coefficients.

b) Casting of each obtained element to the s1Qn-format in order to come back to the set format after the multiplication steps.

5. **Accumulation operation:**

   a) Calculation of x- and y-signal: summation of corresp. values in an accumulator.

   b) Scaling back of the x- and y-signal to maximum limits [-2,2) using the common scaling factor[8].

   c) Casting of x- and y-signal to the s1Q10 format in order to come back to the initial format after the multiplication and accumulation steps.

6. Angle calculation.

In this setup the upscaling of the values after the step 4 a) is not carried out. This leads to a loss of information during the back casting in step 4 b) but on another hand to a significant increase of processing speed because no conditioning of the individual value in the data arrays $\mathbf{B_x}$ and $\mathbf{B_y}$ is done here.
The obtained test results for the angular accuracy depending on the variation of the fraction length n is depicted in Figure 5.29, the angular errors recorded in Table 5.9.

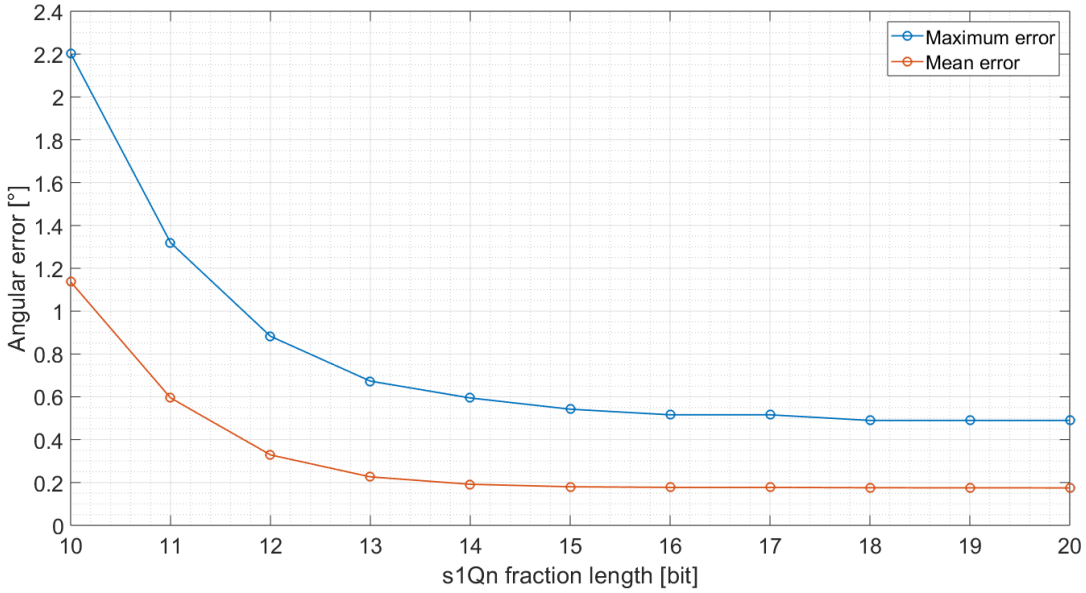


Figure 5.29: Increase of angular accuracy by enlarging of the bit width for the fractional part of arithmetic operations registers: Maximum and mean angular errors (test 2). The initial errors are slightly higher that in the first test. Also here a rapid increase of angular accuracy is observable by increase of fractional part from 10 to 12 bit. Angular accuracy requirement of $\varphi_{err} \leq 1°$ is met at a fractional part bit width of 12 bit as well. Saturation is reached by a fractional part width of 14 bit as well. *Simulation setup: 15 x 15 array, 5 mm z-distance, 12 bit ADC resolution, sphere encoder magnet.*

[8]Same as previous, $max(|x|, |y|)$ and multiplication by 2.

Fraction length [bit]:	10	11	12	13	14	15	16	17	18	19	20
Maximum angular error [°]	2.20	1.32	0.88	0.67	0.59	0.54	0.51	0.51	0.48	0.48	0.48
Mean angular error [°]	1.13	0.59	0.32	0.22	0.19	0.18	0.17	0.17	0.17	0.17	0.17

Table 5.9: Maximum angular errors depending on the register width for the fractional part of appplied s1Qn fixed point format.

Conclusions: Unfortunately, by using the s1Q10 format and performing the arithmetic operations without an additional increase of bit widths for arithmetic operations it is not possible to meet the angular accuracy requirement of $\varphi_{err} \leq 1^{\circ}$. Based on the assumed simulation constraints it was shown that an enlargement of data registers which are used exclusively for arithmetic operations in the multipliers is indeed necessary.The enlargement of amount of bits for the fractional part from 10 to 12 it leads to the fulfillment of set angular accuracy requirement.
Furthermore it was shown that an upscaling of the values after the multiplication operations to the maximum range of [-2,2) before their casting is not necessarily required. The increase of the angular accuracy associated with it is marginal. Based on obtained results, it is proposed to use the bit widths for the multiplies and adder units according to the format s1Q12 at least.
In analogy to the angular error reduction associated with an increase of register width observed for the LoG approach it is expected that the other candidates will exhibit a similar behavior. But, in contrast to the LoG approach, more multiplications and much more complex procedures are required, especially for the estimation of coefficients of the 2D polynomial basic functions for field approximation, where the arithmetic accuracy will be compulsorily diminished. The arithmetic operations for the 2D-DFT and the approximation are also associated with high demand of optimizations. Thus, it can be possible that the 2D-DFT and approximation approach will require larger register widths for the fulfillment of the angular accuracy requirement.

Final conclusions for further investigations of algorithms

In the average, the extended 2D-DFT, LoG, approximation and convolution algorithms exhibited the same performance and pass this test as complete group.

5.7.2 Misalignment test

In this test the robustness of all algorithms passed through the proof of concept is examined. The maximum possible misalignment bounds where the angular accuracy requirement $\varphi_{err} \leq 1^\circ$ is still met are determined.

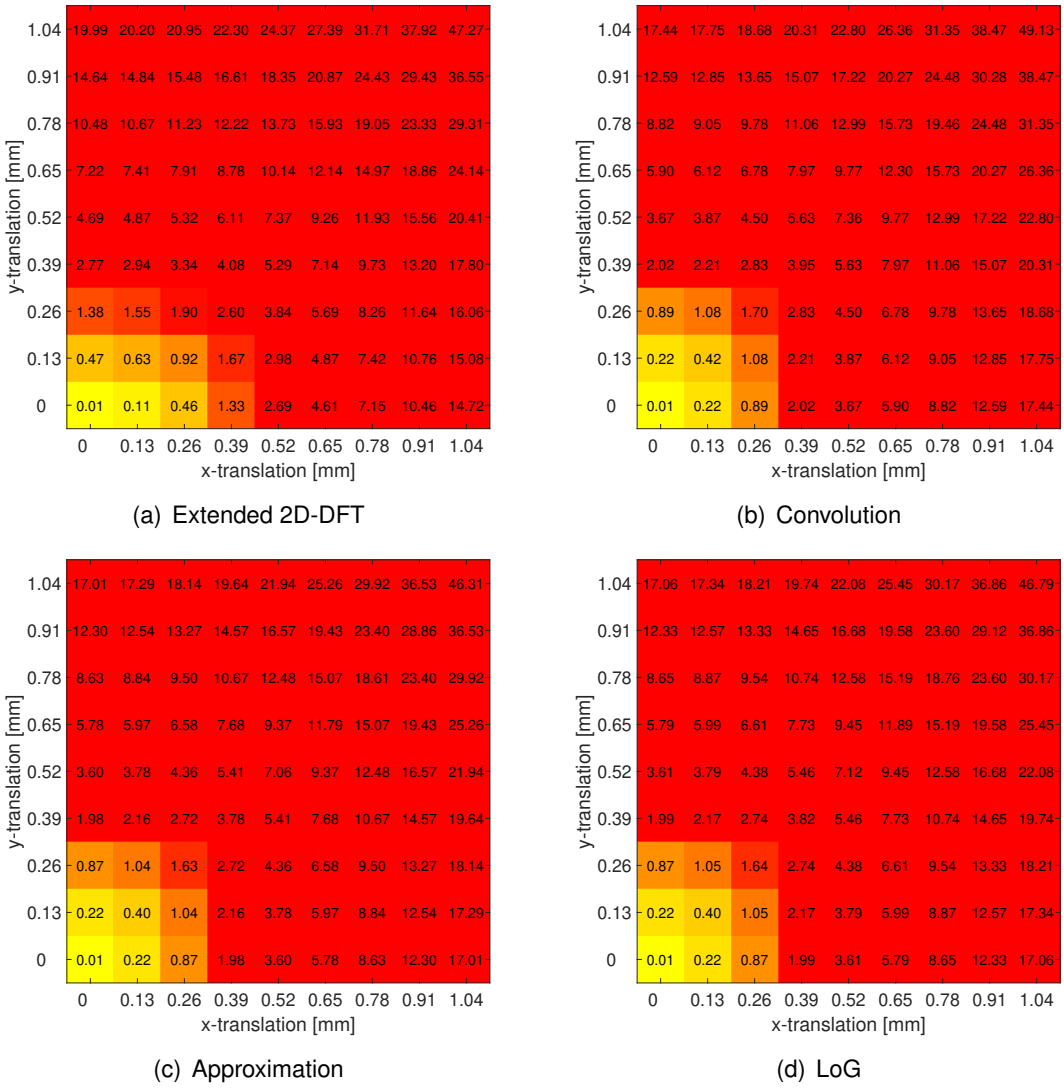
Misalignment robustness (subblock 2.0)

In the frame of the safe operating area the corner cases shown in Table 5.10 were applied in the simulation. In total, 4 combinations of x-tilt and different z-distances between sensor plane and encoder magnet were set. For each combination, the full areal of x- and y-translation limits was varied. The array size is set to 15 x 15 elements and the ADC resolution to 14 bit, in accordance to the obtained results in Section 5.7.1 in order to get a minor quantization error.

Misalignment corner case	x-translation [mm]	y-translation [mm]	z-distance [mm]	x-tilt [°]
1	-1 to 1	-1 to 1	2	0
2	-1 to 1	-1 to 1	2	5
3	-1 to 1	-1 to 1	5	0
4	-1 to 1	-1 to 1	5	5

Table 5.10: Corners of misalignment parameters space within the SOA

For the visualization of results, images with x-/y-translation planes and angular error entry at each x-/y-translation combination were created and are included in Appendix G.5.5.
All examined algorithm candidates exhibit approximately identical sensitivity against applied misalignment effects.
In Figure 5.30 a direct comparison between results of extended 2D-DFT, approximation and LoG algorithms in xy-range of [0,1] mm is shown. The maximum errors caused by misalignment case 1 are compared in this visualization. It is evident from this figure that all algorithm candidates are extremely sensitive against misalignment of the encoder magnet system. The angular accuracy obtain by approximation and LoG with optimized bandwidth are very similar and for small misalignments completely identical. A tiny larger angular error is observable for the convolution algorithm. The angular error of extended 2D-DFT is not symmetrical but in mean it corresponds to angular errors of other candidates. In this investigated case, the angular accuracy requirement $\varphi_{err} \leq 1^\circ$ gets violated from all algorithms by a mechanical x-/y-translation of the encoder magnet between 0.26 mm and 0.39 mm.



(a) Extended 2D-DFT

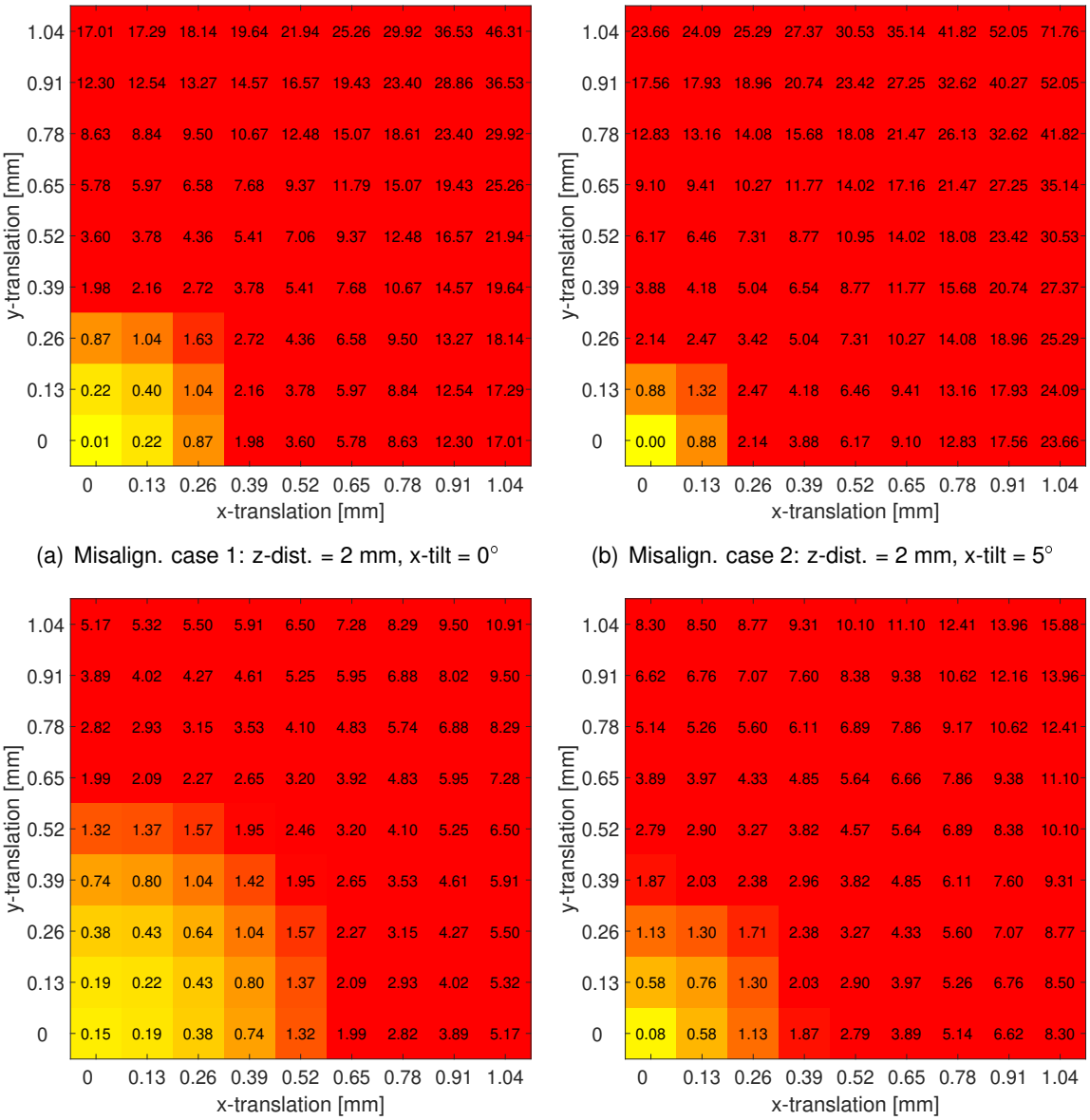
(b) Convolution

(c) Approximation

(d) LoG

Figure 5.30: Misalignment test results. Results of all considered algorithms. Comparison of maximum angular errors caused by misalignment case 1: z-distance = 2 mm, tilt = 0°. Yellow x-/y-translation area is the acceptable area in accordance to the angular accuracy requirement of $\varphi_{err} \leq 1°$. Notable is the non-symmetry in calculated angular errors in case of 2D-DFT, most probably due to limited accuracy of / or multiplications with twiddle-matrices.

In Figure 5.31 the impact of misalignment effect on angular error is compared. For this investigation the approximation algorithm was chosen.



(a) Misalign. case 1: z-dist. = 2 mm, x-tilt = 0°

(b) Misalign. case 2: z-dist. = 2 mm, x-tilt = 5°

(c) Misalign. case 3: z-dist. = 5 mm, x-tilt = 0°

(d) Misalign. case 4: z-dist. = 5 mm, x-tilt = 5°

Figure 5.31: Misalignment test results. Results of approximation approach. Comparison of maximum angular errors caused by 4 applied misalignment cases. Yellow x-/y-translation area in (c) is significantly larger than the others. The smallest acceptable x-/y-translation area is observable in (b) and represents the absolute worst case within the SOA.

The observations in Figure 5.31 deduce the first conclusion that by an increase of the z-distance, less impact of misalignment effect on the system is present due to the field weakening and curvature decrease. The system becomes more robust against misalignment, what can be taken from the comparison of Figures 5.31(a) and 5.31(d). The second conclusion is that the

encoder magnet tilt effects play the most significant role, if the distance between sensor array plane and the encoder magnet is small. In this way a more curved field is measured and the tilt effect becomes higher, what is clearly visible in Figures 5.31(b) and 5.31(d). The algorithms response on tilt variation at closer distances to the encoder magnet is in comparison to other corner cases the worst and the range of acceptable x-/y-translations becomes much smaller.

In summary, the following limits for the maximal acceptable misalignment of the encoder magnets for the fulfillment of angular accuracy requirement of $\varphi_{err} \leq 1^{\circ}$ found in frame of these investigations are shown in Table 5.11

Misalignment corner case		**$\approx$ x-/y translation limits [mm]**
z-distance [mm]	**x-tilt [$^{\circ}$]**	
2	0	$\pm$ 0.26
2	5	$\pm$ 0.13
5	0	$\pm$ 0.39
5	5	$\pm$ 0.26

Table 5.11: Acceptable x- and y-translation misalignment areas depending on z-distance and x-tilt associated with the angular accuracy requirement $\varphi_{err} \leq 1^{\circ}$. *Simulation setup: 15 x 15 array, 14 bit ADC resolution.*

Array size and misalignment effect dependency (subblock 2.1)

The angular error caused by misalignment effects was investigated in dependence of array size variation. For this test 3 random chosen misalignment cases documented in Table 5.12 were applied and its impact on the angular accuracy in dependence of array size examined.
The determined mean angular errors in dependence of array size for all algorithm candidates and applied misalignment cases are reported in Table 5.13. 3D bar graphs for all three cases with maximum and mean angular errors can be found in Appendix G.5.6.

Misalignment case	**x-translation [mm]**	**y-translation [mm]**	**z-distance [mm]**	**x-tilt [$^{\circ}$]**
1	0.39	0.53	2	0
2	0.79	0.66	4	3
3	1.06	-0.93	5	5

Table 5.12: Randomly chosen misalignment cases for the test of array size and misalignment effect dependency.

Misalign. case	Array size:	3x3	5x5	7x7	9x9	11x11	13x13	15x15
	Error [°]:	mean	mean	mean	mean	mean	mean	mean
1	*Ext. 2D-DFT*	3.142	3.418	3.684	3.849	3.960	4.045	4.100
	Conv.	3.143	3.447	3.557	3.579	3.586	3.594	3.589
	LoG	3.143	3.299	3.371	3.409	3.437	3.465	3.476
	Approx.	3.142	3.296	3.365	3.399	3.422	3.445	3.449
2	*Ext. 2D-DFT*	4.176	4.356	4.549	4.670	4.750	4.809	4.853
	Conv.	4.176	4.341	4.410	4.424	4.425	4.430	4.429
	LoG	4.176	4.262	4.302	4.325	4.342	4.357	4.366
	Approx.	4.176	4.260	4.299	4.320	4.334	4.345	4.351
3	*Ext. 2D-DFT*	6.372	6.496	6.581	6.641	6.677	6.709	6.722
	Conv.	6.373	6.568	6.624	6.650	6.656	6.663	6.651
	LoG	6.373	6.476	6.518	6.548	6.565	6.589	6.585
	Approx.	6.372	6.474	6.514	6.543	6.557	6.578	6.570

Table 5.13: Array size and misalignment effect dependency test results. Change of mean angular error for 3 applied misalignment cases.

Due to increase of sensor pixels amount on the chip die and obtain of more spatial resolution of field components the angular error caused by misalignment becomes slightly worse. This is evident for all 3 misalignment cases. Also here a similar effect to the observable in Figure 5.31 is apparent. The smaller the distance between the sensor plane and encoder magnet the higher is the impact of misalignment effect. The angular error of 2D-DFT approach is higher than all others. The lowest error exhibits the approximation approach, closely followed by LoG.

Conclusions for further investigations of algorithms

All algorithm candidates passed the tests well. Despite the fact that the LoG convolution approach exhibits almost the same performance as all other algorithms it is nevertheless not propagated to the next block due to its methodical similarity with the single LoG approach. It needs also to be mentioned that its results are inconsequential in all performed test and negligible smaller than the other results in the group. This effect comes indeed from a smaller filter kernel mask. In consequence whereof, the extended 2D-DFT, single LoG and approximation approaches are propagated to the next blocks.

5.7.3 Tracking algorithms test

The quality of tested tracking algorithms is rated on two criteria:

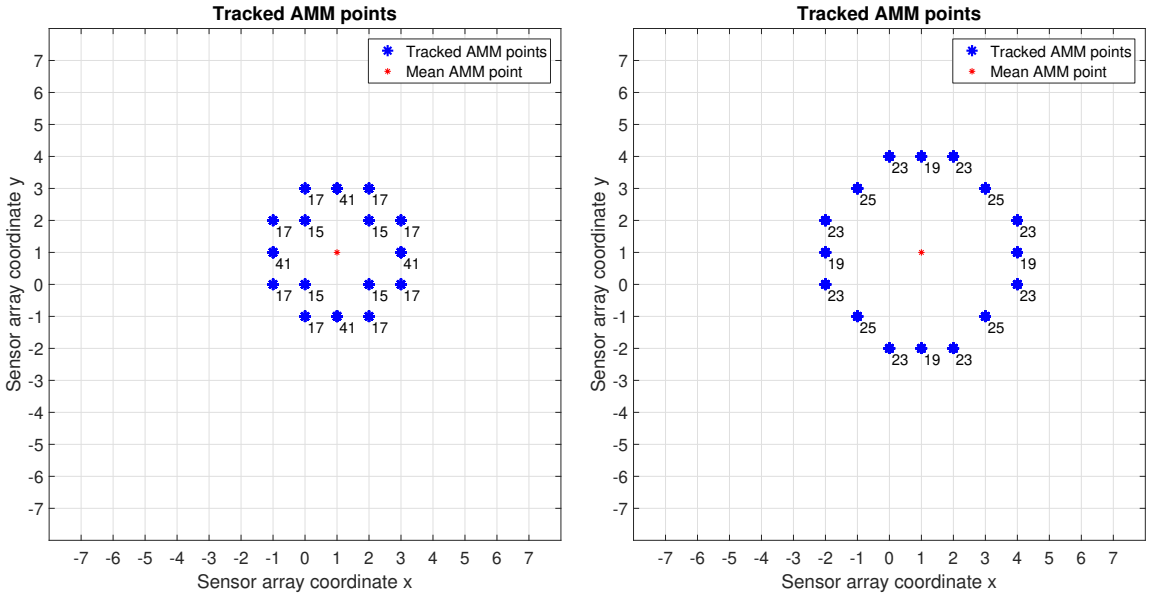
- **Localization accuracy of the xy-paths of all AMM points**
  The localization accuracy is determined over calculation of deviations of all found AMM points from expected AMM points measured as euclidean distances one by one and subsequent averaging of resulting deviation called as MDD, what is described by the Formula 4.12.
- **Deviation from the mean AMM point**
  After averaging of coordinates found for all AMM points within a full encoder magnet rotation the euclidean distance deviation of calculated mean AMM points from the expected one is determined, described by the Formula 4.14.

The SNR was varied from 100 dB to 20 dB in 0.01 dB steps. The amount of SNR repetitions was increased to 200 in order to get a higher averaging. The array size was set 15 x 15 in order to be able to obtain as large as possible potential distance deviation from the expected AMM points. In this case, the sensor-to-sensor distance unit in which the deviation is measured is 0.1333 mm. For the mechanical misalignment of encoder magnet the conditions shown in Table 5.14 were set. Z-distance was set to maximum, In order to examine the tracking algorithms under the worst case for z-distance within the SOA.

x-translation [mm]	y-translation [mm]	z-distance [mm]	x-tilt [°]
0.1333	0.1333	5	6

Table 5.14: Applied mechanical misalignment parameters of encoder magnet for the test of tracking algorithms.

By application of the misalignment parameter combination in Table 5.14, the closed area around which the AMM points are located, is sufficient enough. The boundary condition for the gradient filter based algorithms were also considered so that the expected AMM point do not exceed the maximum localizable area of applied gradient filters. For the approximation based method, in case of localization of AMM point coordinates outside the area of the sensor array, the coordinates of localized AMM point are held on the maximal possible coordinate. By application of this conditions, all the algorithms results are treated equally and are compared appropriately.
The expected AMM points, calculated by 4.13 for the sphere and disc encoder magnet are depicted in Figure 5.32. The xy-paths resulting from all localized AMM points are not the same, due to the different curvatures of the encoder magnets fields.

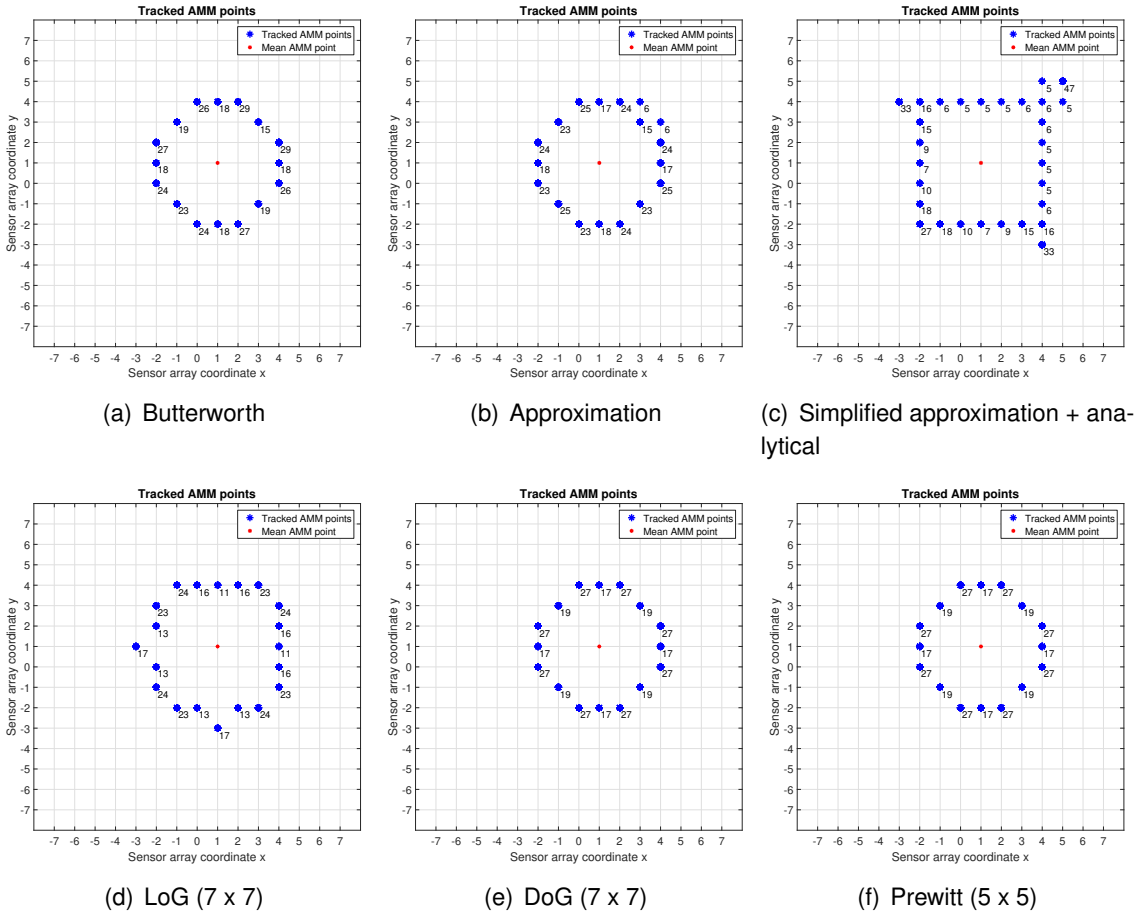


(a) Sphere magnet

(b) Disc magnet

Figure 5.32: Expected localized AMM points within the sensor array area for the test of tracking algorithms for sphere and disc magnet (the numbers represent localization frequency of the AMM point). (a) Expected AMM points resulting from rotation of sphere magnet. (b) Expected AMM points resulting from rotation of disc magnet. The closed area by all AMM points is for the sphere magnet smaller than for the disc magnet. *Simulation setup: 15 x 15 array, 0.1333 mm x- and 0.1333 mm y-translation, 6° x-tilt, 5 mm z-distance, disc encoder magnet.*

Figure 5.33 shows the initially localized AMM points of all considered tracking algorithm candidates, applied to disc magnet data (no White Gaussian noise added). It is clearly visible that the LoG and approximation based methods exhibit the highest initial localizations errors.



(a) Butterworth (b) Approximation (c) Simplified approximation + analytical

(d) LoG (7 x 7) (e) DoG (7 x 7) (f) Prewitt (5 x 5)

Figure 5.33: Initially localized AMM points by considered tracking algorithms. Both approximation methods as well as the LoG method shows initial deviations from the expected AMM point. All mean AMM points are localized by all algorithms exactly.

The test results with the determined deviations in localization of AMM points are illustrated in Figures 5.34 and 5.34 for the sphere magnet and in Figures G.25 and G.26 for the disc magnet. From Figure 5.34 it can be taken that the LoG filter based algorithm exhibits the worst behavior. Its MDD increases rapidly. Its initial error is around 1 sensor to sensor distance unit. Simplified approximation method exhibits also a high initial error. DoG and Prewitt methods behave similarly. The smallest initial error is achieved with Butterworth and approximation methods. The highest stability in the range from 100 dB to $\approx$ 60 dB is achieved by both approximations methods. DoG, Prewitt and Butterworth show a medium localization performance. In the area $\leq$ 60 dB a break even point is reached where the simplified approximation methods exhibits a better MDD than the normal one. In the area < 45 dB the MDD almost does not change at all e.g their MDD goes into the *saturation*, with the exception of normal approximation method. Its *saturation* is reached at $\approx$ 30 dB.

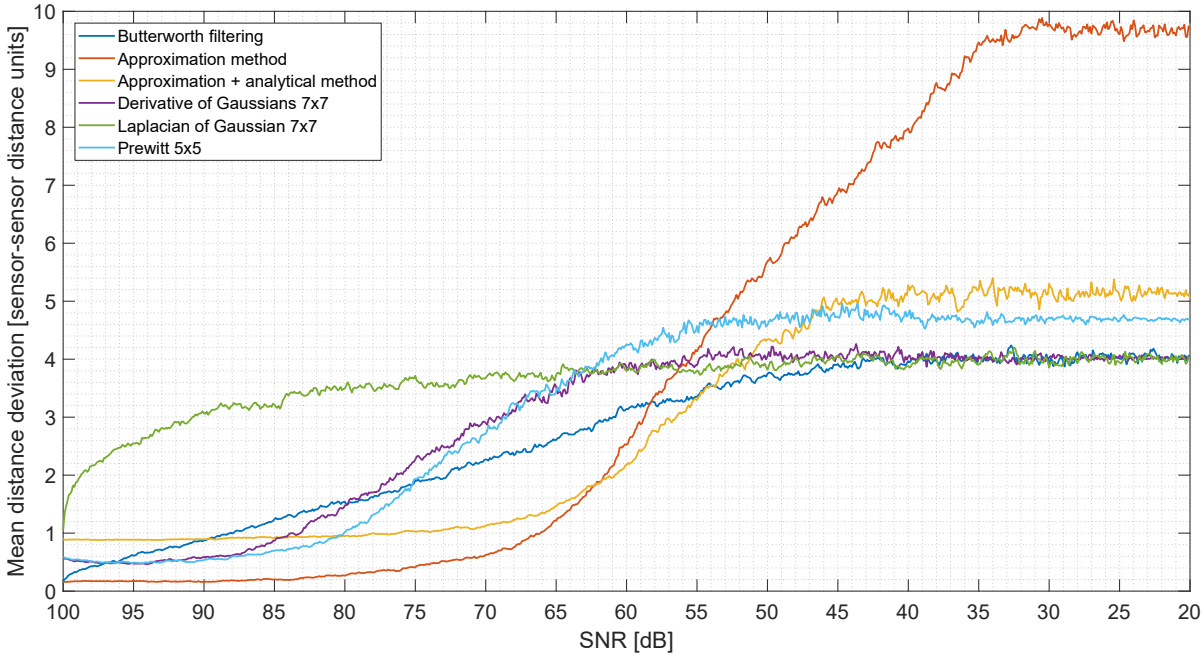


Figure 5.34: Mean distance deviation for all localized AMM points for the sphere magnet data. Initial MDDs of LoG and simplified approximation are the highest. LoG accuracy decreases rapidly. Other gradient filter based algorithms behave equally. Approximations methods are stable until $\approx$ 70 dB.

However, the complete knowledge about the performance of the algorithms is only gained by inspection of the MDD of the mean AMM point, what is shown in Figure 5.35. Here the LoG losses its accuracy and at an extremely high SNR value of $\approx$ 95 dB. The mean AMM point found by LoG jumps to the sensor-to-sensor distance unit of 1 in the area between 95 and 90 dB. Afterwards, at an SNR value of approximately 70 dB the DoG, Prewitt and Butterworth approaches lose their accuracy, one by one. Their accuracy error becomes also 1 sensor-to-sensor distance unit. In comparison to other candidates the approximation method keeps its accuracy much longer. The first decrease in the accuracy is observed in a SNR area between 45 and 40 dB.

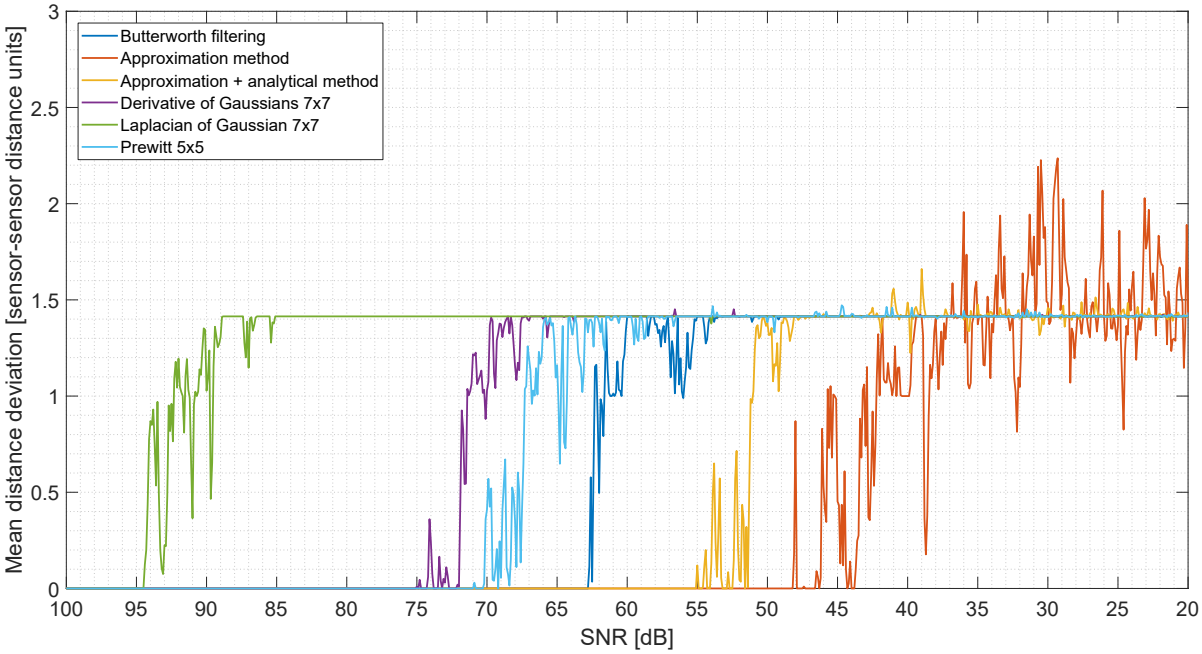
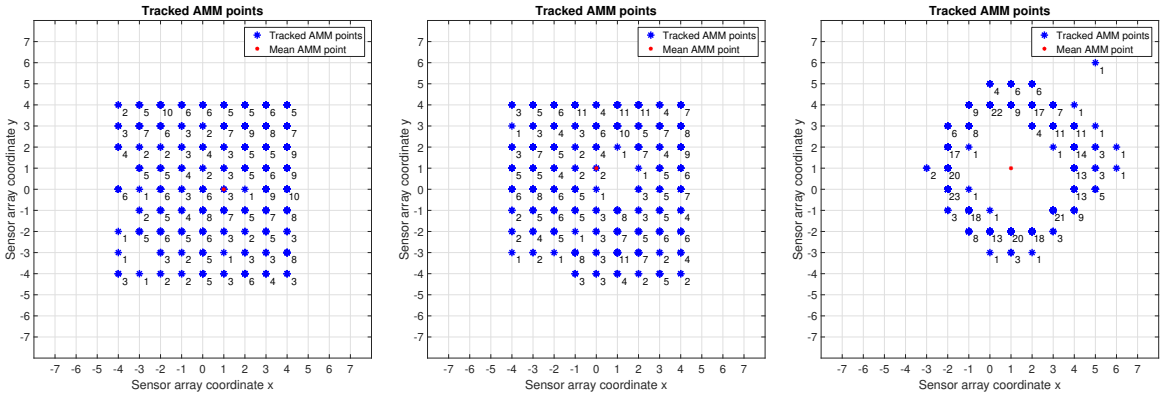


Figure 5.35: Mean distance deviation for localized mean AMM point for the sphere magnet data. LoG exhibits the worst accuracy, normal approximation the best one. Accuracy limits of Butterworth, LoG, DoG are close to each other. Simplified approximation is the second best.

Previously called *saturation* effect observed in Figure 5.34 becomes more clearly and understandable by contemplation of the picture 5.36(a) and 5.36(b). The first observation here is that due to the boundary conditions associated with used mask sizes of the gradient methods the localized AMM points fits only into the localizable area and are equally distributed here, see Figure 5.36(b). Notable is the fact that for Butterworth approach no AMM points at the boundaries occur, presumably due to distortion of components within this areas associated with filtering, see Figure 5.36(a). For approximation method results shown in Figure 5.36(c) the localized points are closer to the borders, what is the reason for a higher distance deviation of all AMM points in total, observed in Figure 5.34. Also with the simplified approximation method it is possible to localize the points exactly at the boundaries, but this did not happen in this case presumably. The second observation is that the accuracy of the mean AMM point for the approximation method shown in Figure 5.36(c) is still 100 % correct, while in 5.36(a) and 5.36(b) it was mislocalized.



(a) Butterworth (b) DoG (c) Approximation

Figure 5.36: Localized AMM points by considered tracking algorithms at 35 dB SNR. In (a) and (b) the AMM points are almost equally distributed over all sensor array area, but not in the areas close to the boundaries. The mean AMM points in (a) and (b) was mislocalized. In (c) the initial shape of xy-path of AMM point is still present and the mean AMM point was localized correctly.

Summary and conclusion:

LoG showed poor and disappointing results. Also from DoG method much better results were expected. The expectation of significant stability of approximation approaches in comparison to the gradient based methods was confirmed. A very interesting observation is that a simple Butterworth filtering in spatial frequency domain and localization of the maximum point in the magnitude image exhibited a better result than all the gradient based algorithms, what points to a good suitability for the application of 2D-DFT in the practice. Simple Prewitt gradient filter with smaller mask exhibited better results than the gradients filter combined with the Gaussian function and its rotation variance most likely did not play a substantial role for the localization accuracy.The observed algorithms behavior was confirmed for disc magnet as well. The results are identical, in exception for the upper SNR limits, which are slightly shifted to higher values due to a lower curvature of the disc magnet field. The results can be reviewed in Section G.6.1.

The approximation method including the reinsertion of sensor array coordinates into the approximated basis functions explained in Section 3.6.4 exhibits the best localization accuracy and is therefore proposed as the candidate for the application. It needs to be mentioned that with this method the initial error and accuracy results can be made even better by consideration of higher polynomial degrees.

However, it needs also to be mentioned that the approach based on simplified approximation and analytical evaluation of information is the second best and requires significantly less computational effort in comparison to the approximation algorithm with reinsertion of sensor array coordinates. From the practical point of view it is a suitable candidate as well. Its stable localization accuracy lays until 55 - 50 dB SNR is still within relevant range.

5.7.4 Extended algorithms test

in the following investigations all the algorithms passed the misalignment test in 5.7.2 are extended by the best candidate for the tracking algorithm determined in 5.7.3.

Misalignment robustness (subblock 4.0)

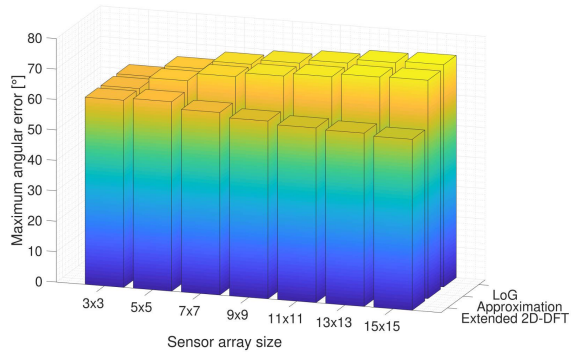
As already explained in Section 3.6, by extension of algorithm for angle information extraction by so called tracking feature, more symmetrical field components are evaluated and in this way a better angular robustness against misalignment effects can be achieved. Main goal of this investigation is to get the knowledge about the tendency of error reduction by application of this algorithmic extension.

For the misalignment case used in this test the worst case within the SOA limitations which was found in Section 5.7.2 was applied and is reported in Table 5.15.

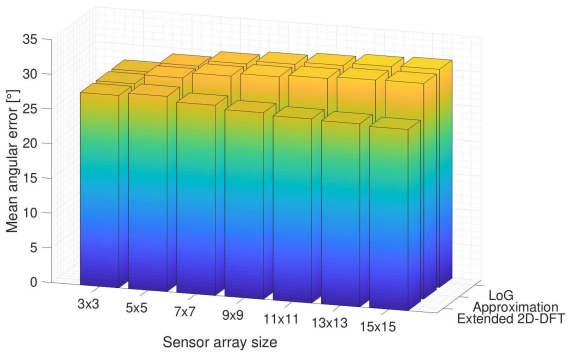
Misalignment case	**x-translation [mm]**	**y-translation [mm]**	**z-distance [mm]**	**x-tilt [°]**
SOA worst case	1.06	1.06	2	5

Table 5.15: Applied mechanical misalignment parameter of sphere encoder magnet for extended algorithms test.

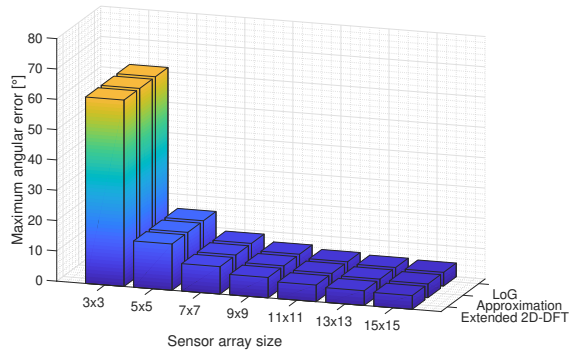
As well as in 5.7.2 also here a 14 bit ADC resolution was chosen, which leads to a small initial quantization error and is practically possible to apply.
The mean and maximum angular errors caused by applied misalignment and their reduction in dependence of an increase of the sensor array size were investigated in this test. The obtained results in direct comparison to non extended algorithms are shown in Figure 5.15 and reported in Table 5.37.
For Figure 5.37 and comparison of angular errors between 5.37(a) and 5.37(c), as well as 5.37(b) and 5.37(d) it can be seen that by an increase of array size and associated movement possibility of the extraction mask within the array area a significant angular error reduction is achieved. The highest error reduction is obtained by increase of the array size from 3 x 3 to 5 x 5 with the extraction mask movement possibility. Between an array size increase from 5 x 5 and 7 x 7 the errors reduction high. By subsequent array size increased it does not change significantly.



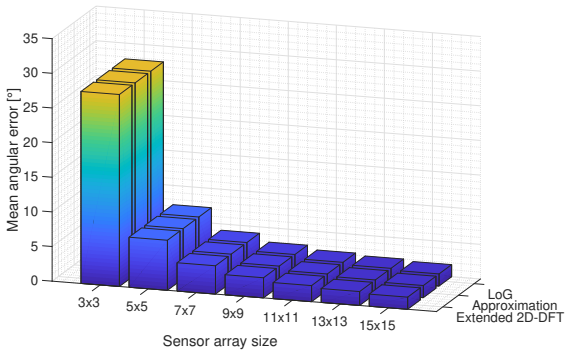
(a) Maximum angular error of non extended algorithms



(b) Mean angular error of non extended algorithms



(c) Maximum angular error of extended algorithms



(d) Mean angular error of extended algorithms

Figure 5.37: Comparison of maximum and mean angular errors of non extended and AMM point tracking extended algorithms. A significant improve of angular accuracy is observed by an increase of array size from 3 x 3 to 5 x 5 elements.

Interesting fact is that the angular errors of the 2D-DFT algorithms (extended with field realignment) observable in Figure 5.37(a) and 5.37(b) reduces when the array size is increased. This is achieved because the 2D-DFT algorithm evaluates the total field information while with LoG and approximation algorithms exclusively only the field curvature is evaluated. Due to this, for large misalignment effects, the 2D-DFT approach including field realignment feature possibly represents a potential advantage. However, the remained angular error of 2D-DFT algorithm is in this setup still not satisfied.

The direct comparison of angular errors values depending on array size are shown in Table 5.16.

Tracking extension	**Array size:**	3x3	5x5	7x7	9x9	11x11	13x13	15x15
	Error [°]:	max	max	max	max	max	max	max
not activated	*Ext. 2D-DFT*	61.52	62.28	59.97	58.39	57.39	56.94	56.18
	LoG	61.54	65.39	68.06	69.71	71.01	72.45	73.03
	Approx.	61.52	65.29	67.84	69.31	70.37	71.54	71.76
activated	*Ext. 2D-DFT*	61.52	15.19	8.862	6.380	5.116	4.461	3.849
	LoG	61.54	15.19	8.863	6.380	5.116	4.461	3.849
	Approx.	61.52	15.19	8.862	6.380	5.116	4.461	3.849

Table 5.16: Direct comparison of mean and maximum angular errors of non extended and AMM point tracking extended algorithms.

To get a clear knowledge about the angular error reduction tendency, the reduction of relative maximum error related to the initial one in percentage was calculated and is recorded in Table 5.17.

	Array size:	5x5	7x7	9x9	11x11	13x13	15x15
Relative error reduction [%]	*Ext. 2D-DFT*	75.31	85.59	89.63	91.68	92.75	93.74
	LoG	75.31	85.59	89.63	91.68	92.75	93.74
	Approx.	75.31	85.59	89.63	91.68	92.75	93.74

Table 5.17: Relative reduction of maximal angular error of AMM point tracking extended algorithms in dependence of sensor array size.

With the application of tracking algorithm the maximal angular error reduction in the first step when the array size is increased from 3 x 3 to 5 x 5 is around 75%. In the step afterwards, the reduction of 85% is achieved. The maximum reduction in frame of this setup amounts to 93%. All the methods show completely the same behavior.

Conclusions: All the investigated algorithms exhibit the same behavior. Their robustness against misalignment effect combined with the tracking algorithm is identical.
With the extension of all algorithm candidates by AMM point tracking algorithm and by application of an extraction mask size of 3 x 3 elements the maximum error caused by worst misalignment case within the SOA limitations could be maximally reduced by **93.74** %. However, the remained error amounts to 3.8° and is still not acceptable, what leads to the conclusion, that no acceptable angular error compensation can be achieved with this algorithmic extension and therefore an angular error correction procedure is definitely needed.

Robustness against signal-to-noise ratio variation (subblock 4.1)

Of course, this extension of all algorithms by in decrease of effective area for angle information extraction associated with it plays directly against the noise effects of the signals. The smaller

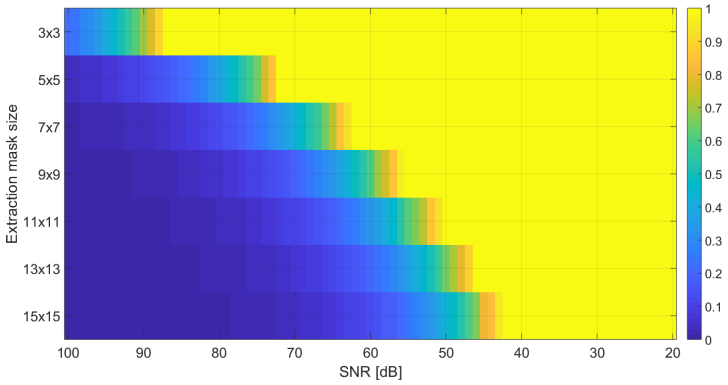
the extraction mask the worse the impact of noise effects become. In order to prove the applicability of the tracking extension the following test was carried out. The total array size within the sensor die area was set to 15 x 15. The SNR was varied in defined limit, from 100 dB to 20 dB. The extraction mask size was varied from 15 x 15 to 3 x 3 elements. No misalignment was applied here in order to get a maximum possible angular error of 0°. Since the limits of this extension need to be found in this test setup the z-distance was set to maximum of 5 mm, in order to obtain the worst possible noise effects.
In analogy to 5.7.1, absolutely identical evaluation images were created to obtain a visual comparison of the noise impacts. In Figure 5.38 the maximum and minimum error associated with the noise impact for the approximation algorithm is shown. The $\varphi_{err} \leq 1^\circ$ boundary is set visually by scaling, as well as G.5.2.

All algorithms exhibit the same performance in this test. Their evaluated figures are included in Appendix G.7.1. From Figure 5.38 it can be taken that SNR value for the required angular accuracy increases rapidly, if the extraction mask size is reduced. The highest tendency for the required increase of the SNR value is observable between 9 x 9 to 7 x 7, 7 x 7 to 5 x 5 and 5 x 5 to 3 x 3 extraction masks. The tendency of required SNR for the adherence to angular error criteria depending on the array size is in both figures is not linear.
In Table 5.18 the determined limits of SNR for the angular accuracy requirement of $\varphi_{err} \leq 1^\circ$ are represented.

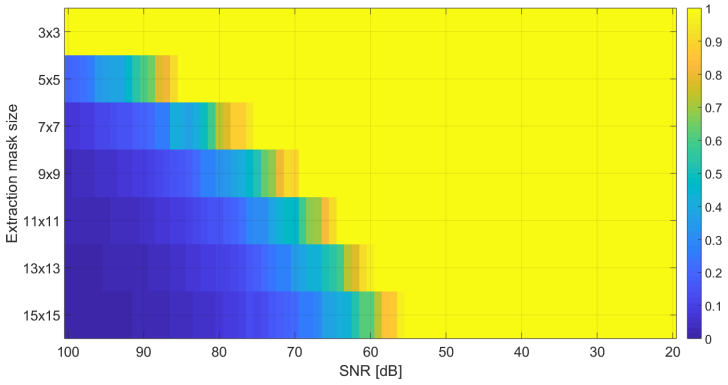
	Extraction mask size:	3x3	5x5	7x7	9x9	11x11	13x13	15x15
≈ SNR for 1° angular error limit [dB]	*Mean error*	87	72	62	56	51	46	43
	Max. error	>100	85	76	70	65	60	56

Table 5.18: Limits for an angular accuracy $\varphi_{err} \leq 1^\circ$ for AMM point tracking extended algorithms in dependence of extraction mask size for an array size of 15 x 15. Results of approximation algorithm.

Additional bar plots were created so that the angular error behind the $\varphi_{err} \leq 1^\circ$ limit can also be evaluated and included in Appendix G.7.1 as well.



(a)



(b)

Figure 5.38: Investigated impact of noise on the angular accuracy of AMM point tracking extended algorithms. Results of approximation algorithm. (a) Mean error depending on SNR and extraction mask size. (b) Maximum error depending on SNR and extraction mask size. *Note: the maximum limit of color was set to 1 in order to see the* $\varphi_{err} \leq 1^{\circ}$ *boundary clearly. Simulation setup: 15 x 15 array, no misalignment, z-distance 5 mm, sphere encoder magnet.*

Conclusions: It will not be practically possible at all to rely on the tracking concept with a extraction mask size of 3 x 3 and array size of 15 x 15. There are more than 100 dB SNR signal quality needed. For the extraction mask size of 5 x 5 elements and an array size of 15 x 15 a SNR value around 85 dB is needed, what is still very ambitious and practically less possible. The combination with 7 x 7 extraction mask for a 15 x 15 array is applicable. With a decrease of amount of sensor pixels on the die the SNR of the signals for the same extraction mask size will definitely will become better because a higher field curvature capture will automatically be obtained. For an expected SNR value of the signal quality an optimal combination between array size and extraction mask size can be found. The search of the optimal combination can be carried out in the context further work.

5.7.5 Angular error correction

The results included in this section represents the last part of the performed investigations of suitable algorithms for angle information extraction passed through the tests. In these investigations, firstly the suitability of angular error correction algorithms associated with the angular teach-in during the calibration phase of the sensor array elements at EOL is examined. This test represents a feasibility study of developed methods for angular error correction. Secondly, the angle information extraction algorithms are combined with algorithms for angular error correction by a the angular teach-in phase. Their performance regarding angular error correction due to mechanical misalignment of encoder magnet in dependence of degrees of freedom / polynomial degree and amount of matching points is examined. Sufficient limits of these parameters are found out in context of performed investigations which lead to a conclusion for their practical implementation.

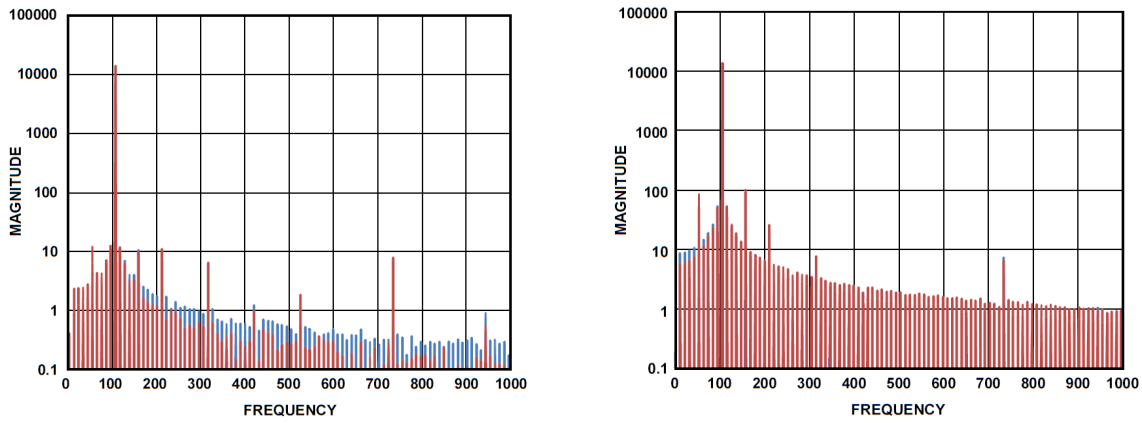
Harmonics cancelation (test 1)

Harmonic frequencies represent the main contributors to an accuracy decrease in a sensor system. Possible origins of signal distortions in the form of superimposed harmonic frequencies are summarized in 3.7. Therefore, it is absolutely essential to cancel their impact on the angular accuracy at the end of line calibration, what represents nowadays the main challenge in sensor quality for all manufacturers.
For the performing of this test a short online study was carried out. The technical application note [47] was taken as a reference for this problem.

The simulation setup looks as follows:

1. Creation of ideal sine and cosine signal with an amplitude of 1 arbitrary unit and fundamental frequency equal to 1 Hz.
2. Creation of additional sine and cosine signal harmonics signals with different values for magnitude and frequencies, based on values derived from [47].
3. Superimposing of created harmonics with x-/y-signals of the fundamental rotation frequency of encoder magnet.

Afterwards, the developed angular error correction algorithms were applied on the distorted x-/y-signals. The teach-in procedures were carried out for starting from the minimum required and ending at maximum possible degrees of freedom / polynomial degree as well as amount of matching points (see 5.5). The angular error reduction was evaluated.
Figure 5.39 shows the FFT carried out for performance examination of the sensor signals analyzed in [47]. The magnitudes of frequency peaks illustrated in 5.39(a) and 5.39(a) were used for the performed test.



(a) (b)

Figure 5.39: FFT analysis of the signals of the AMR sensor analyzed in [47]. (a) FFT analysis of a well aligned sensor. (b) FFT of a sensor misaligned 1 mm away from the magnetic center. The main magnet rotational frequency is not exactly at 100 Hz. Due to this, higher harmonics are slightly shifted to right. Peaks at 0.5x and 1.5x fundamental are indeed the real rotational frequency of magnet what is doubled by AMR effect and in this data not completely suppressed what is usually the case. This effect is technology related and come e.g. due to process related biasing of xMR bridge element. Hence, the peaks at 0.5x and 1.5x are neglected in this investigation.

Based on [47], the following assumptions for harmonics distortion frequencies contained in x- and y-signal of the „fictional“ sensor pixel over the full rotation were applied:

Harmonic frequency:	**2nd**	**3rd**	**5th**	**7th**
Amplitude distortion ratio [%]	0.10	0.07	0.05	0.08
Magnitude [dB]	60.00	63.09	66.02	61.93
Phase [°]	0.00	0.00	0.00	0.00

Table 5.19: Assumed values for the magnitudes and phases of superimposed harmonics for the harmonics cancelation test. Amplitude distortion ratio is related to the initial amplitude of the ideal signal in time domain.

The maximum angular error, caused by applied harmonics amounts to 0.1132°.
Figure 5.40 shows the angular error correction results obtained by homography approach in dependence of degrees of freedom and amount of equidistant distributed matching points.

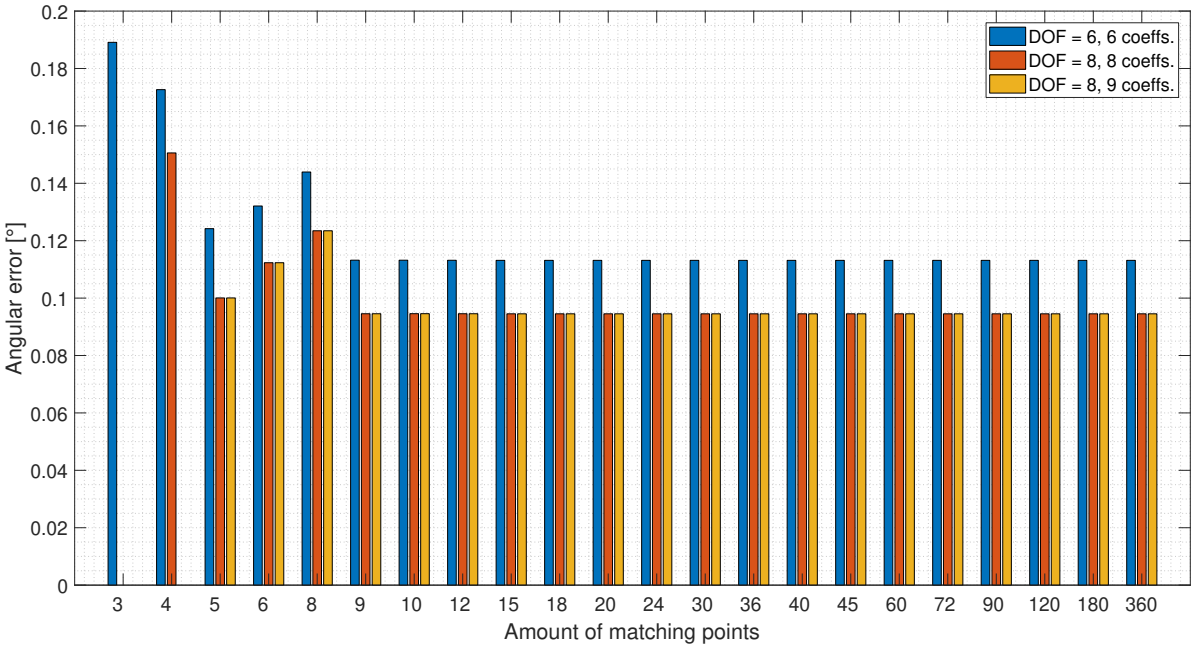


Figure 5.40: Investigation of linear angular error correction algorithm based on homography approach: harmonics cancelation. Remained angular error by DOF 6 is the same as the initial one. Remained angular errors achieved by DOF 8 with 8 and 9 estimated coefficients are completely equally.

From figure 5.40 it is evident that this approach it is unable to decrease the angular error significantly. The remained error after homography based correction of DOF 6 equals to the initial angular one. Errors achieved by DOF 8 with 8 and 9 estimated coefficients are only barely better and completely equal. An interesting fact is that the angular errors at minimum required amounts of matching points, between 3 and 8, is higher than the initial one and only gets smaller if the amount of matching points is ≥ 9. This arises due to unfavorable selection of points within the xy-paths for the error minimization.
Figure 5.41 shows the angular error correction results obtained by angular error correction algorithm based on geometric distortion approach in dependence of degree of the polynomial and amount of equidistant distributed matching points. The angular error reduction here looks different. With a polynomial degree of 2 the angular error is halved. A significant jump in the angular error correction tendency is achieved at polynomial degree of 4. Stability in angular degree reduction is obtained for all polynomial degrees at an amount of matching points around 60.

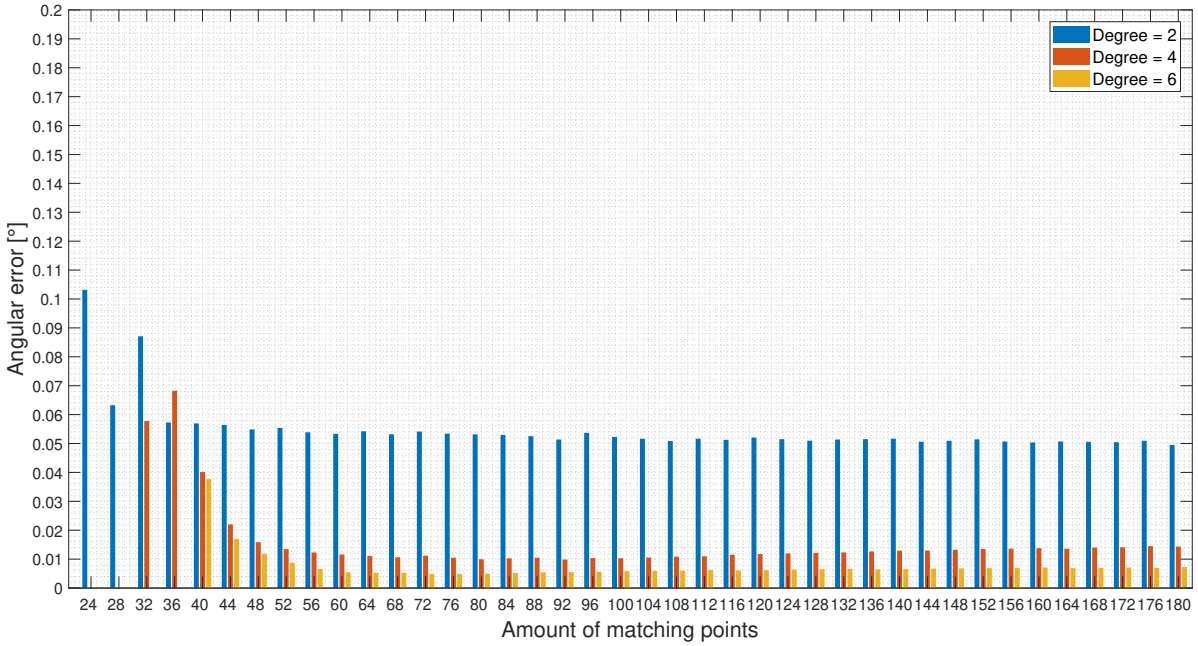


Figure 5.41: Investigation of nonlinear angular error correction algorithm based on geometric distortion compensation: harmonics cancelation. Also here the effect of unfavorable selection of points within the xy-paths in case of small amount of matching points is visible.

The comparisons of uncorrected and corrected angular error progressions for both approaches are in 5.42 and in 5.43.
By application of homography based angular error correction algorithms the angular error in Figure 5.42 gets pressed to the origin but the form of the progression is preserved. The maximum angular error is reduced first by DOF 8. In the central area the error becomes slightly higher, but the maximum error stays in total smaller than the uncorrected one.
From Figure 5.43 it can be seen that the nonlinear error correction exhibits much better result but in the same time associated with discontinuity points, which impact limits the maximal angular error. However, their impact is significantly decreased by higher polynomial degrees. An extremely high reduction of remained maximum angular error is obtained by transition from 2nd to 4th polynomial degree.
To get the full knowledge about the performance of developed algorithms and obtain a direct comparison to 5.39 the corrected x-/y-signals were analyzed in the frequency domain. The DFT analysis results for homography approach with DOF 8 are depicted in Figure 5.44, for nonlinear correction algorithm of degree 4 in Figure 5.45. The DFT analysis for all three degree variations for both algorithms is included in Appendix G.8.1.

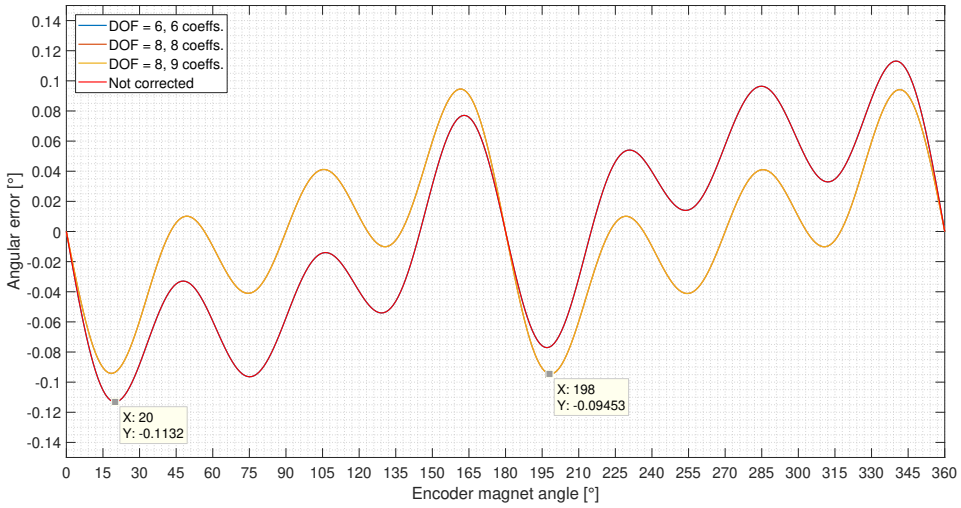


Figure 5.42: Investigation of linear angular error correction algorithm based on homography approach: harmonics cancelation. Comparison of angular errors before and after correction (amount of matching points = 360). DOF 6 curve is overlapped by the uncorrected. Error progressions for DOF 8, 8 and 9 coefficients are completely equal. The maximum angular error is reduced from 0.1132° to 0.0945°.

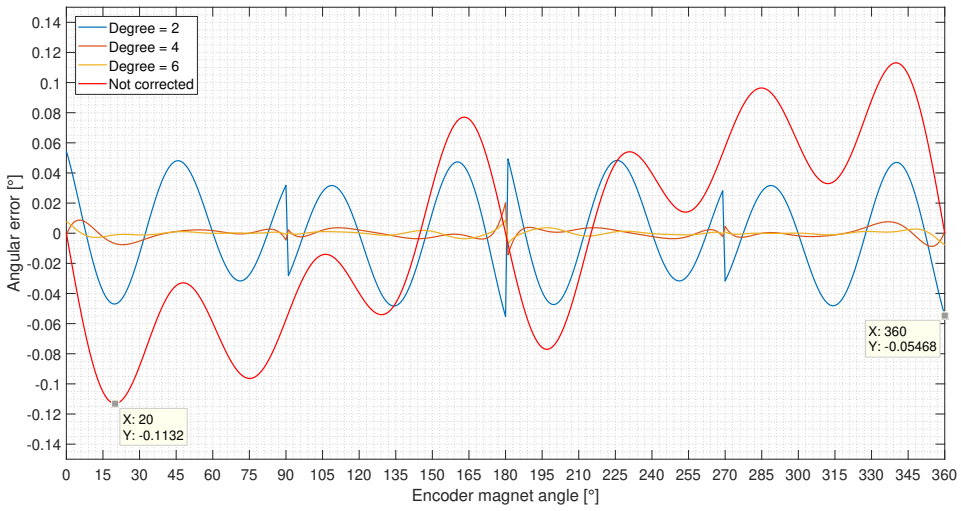


Figure 5.43: Investigation of nonlinear angular error correction algorithm based on geom. dist. compensation: harmonics cancelation. Comparison of angular errors before and after correction (amount of matching points = 181*4). Due to the algorithm functionality the angular error progression is split in 4 parts. High discontinuities occur which are minimized significantly by an increase of polynomial degree. The maximum angular error is reduced from 0.1132° to 0.05468° by polynomial degree of 2, approx. to 0.02° by degree of 4 and approximately to 0.01° by degree of 6.

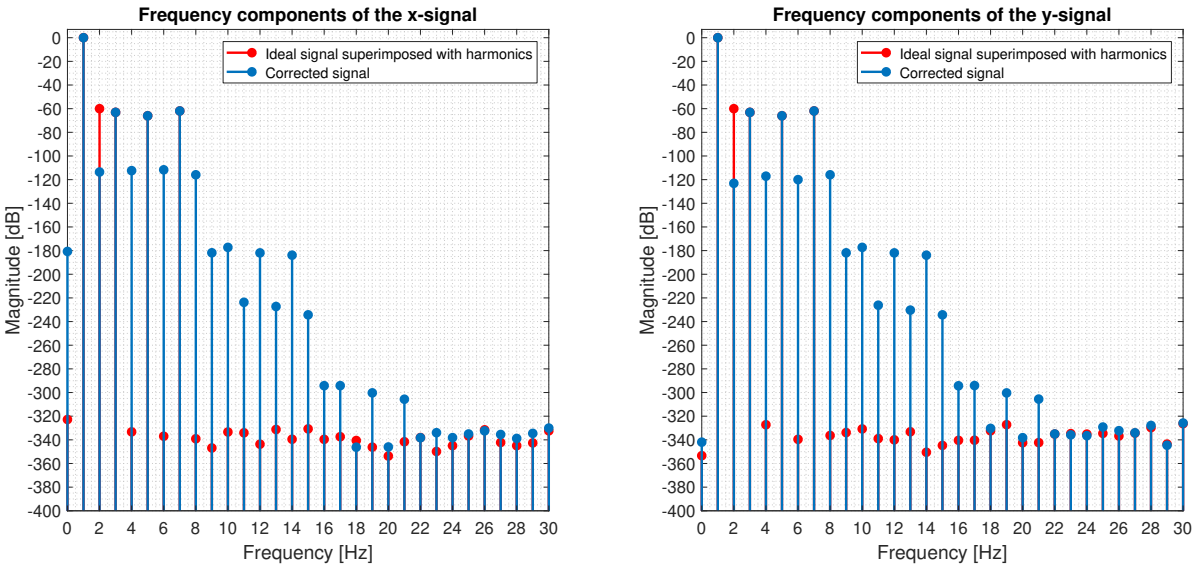


Figure 5.44: Investigation of linear angular error correction algorithm based on homography approach: harmonics cancelation. Comparison of frequency components of the ideal signal superimposed with harmonic frequencies and corrected signal. By application of homography approach only the second harmonics is suppressed in this setup. *Correction setup: DOF = 8, amount of matching points = 360.*

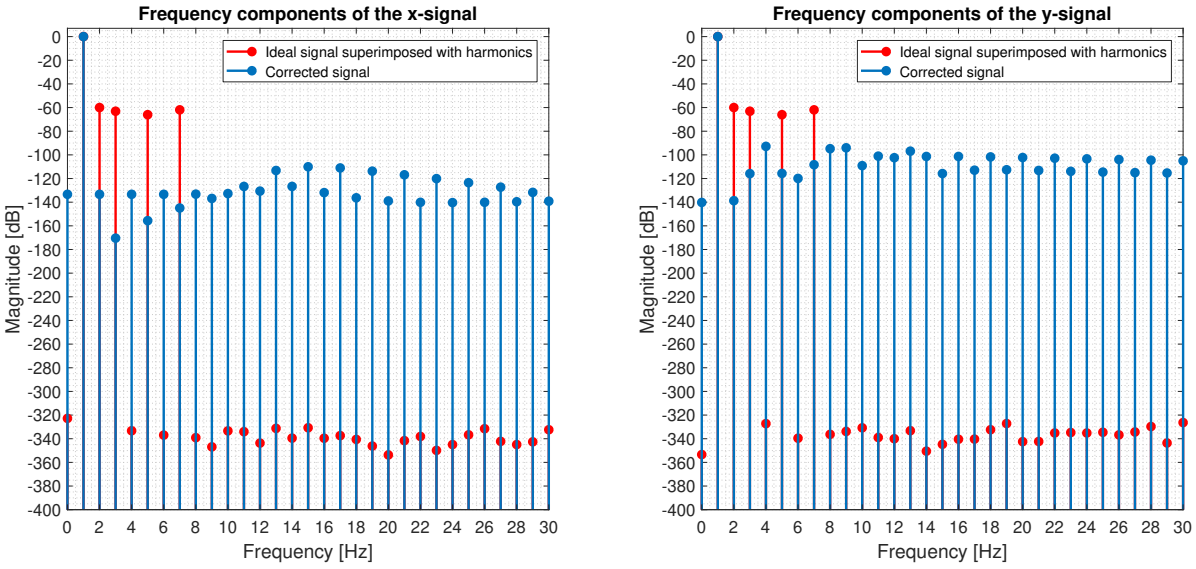


Figure 5.45: Investigation of nonlinear angular error correction algorithm based on geometric distortion compensation: harmonics cancelation. Comparison of frequency components of the ideal signal superimposed with harmonic frequencies and corrected signal. All superimposed frequencies are suppressed by this algorithm to a level < 100 db. *Correction setup: Degree = 4, amount of matching points = 181*4.*

According to the results of performed analysis it becomes clear that by application of homography approach with DOF 8 only the second harmonics is suppressed in this setup. All other harmonics remains. Due to suppression of the second harmonic additional group of harmonics arises in the signals which are below a level of 120 dB.
Due to the discontinuities in the x-/y-signals by application of nonlinear algorithm a constant level of frequencies becomes visible in the spectrum. However, the occurred constant spectral level is below 90dB. This limit is under normal noise level in practice and therefore completely acceptable.

To measure the reduction of harmonics distortion in total and get the knowledge about signal quality increase, the THD described by the Formula 4.28 was calculated for both signal and for all three variations of degree over all components. The THDs are recorded in Table 5.20.

	Not corrected	**Homography**			**Nonlinear correction**		
		DOF 6 (6 coeffs.)	DOF 8 (8 coeffs.)	DOF 8 (9 coeffs.)	2nd ord. (6 coeffs.)	4th ord. (8 coeffs.)	6th ord. (10 coeffs.)
THD x-signal [%]	0.1297	0.1297	0.0987	0.0987	0.0370	0.0025	4.9781e-04
THD y-signal [%]	0.1297	0.1297	0.0987	0.0987	0.0530	0.0090	0.00433

Table 5.20: Angular error correction algorithms test: Harmonics cancelation. Comparision of THDs in % before and after angular error corrections.

The THD values of signals decrease after the homography based correction algorithm only barely. The THD values after the nonlinear correction fell to significant small values.

Conclusions: In general, the performed test shows that both methods are suitable for cancelation of harmonics at the end of line calibrations. Using the angular error correction algorithm based on homography approach only modest results could be obtained. This method seems to be not powerful enough to get a significant cancelation of harmonics. A reduction of maximum error by only approx. 16% is achieved. In comparison to it, the nonlinear error correction algorithm based on geometric distortion cancelation exhibits a much better result and fulfills the expectations. The polynomial degree of 4 seems to be sufficient. The maximum error due to harmonics interference with this polynomial degree is reduced by approx. 82%. Its caused constant noise level visible in the spectrum is below 90dB and thereby outside the practical relevance.

Compensation of misalignment effects (test 2)

This test represents the last test in the total investigations chain. The performance capability angular error correction algorithms for misalignment effects compensation is examined in this test. In accordance to the test plan, two encoder magnet shapes are investigated here in order to determine possible differences in the algorithm performance. Two misalignment conditions of encoder magnets were applied in line with the misalignment compensation investigations: a) an extreme misalignment case and b) the worst SOA misalignment case determined in investigations in Section 5.7.2.

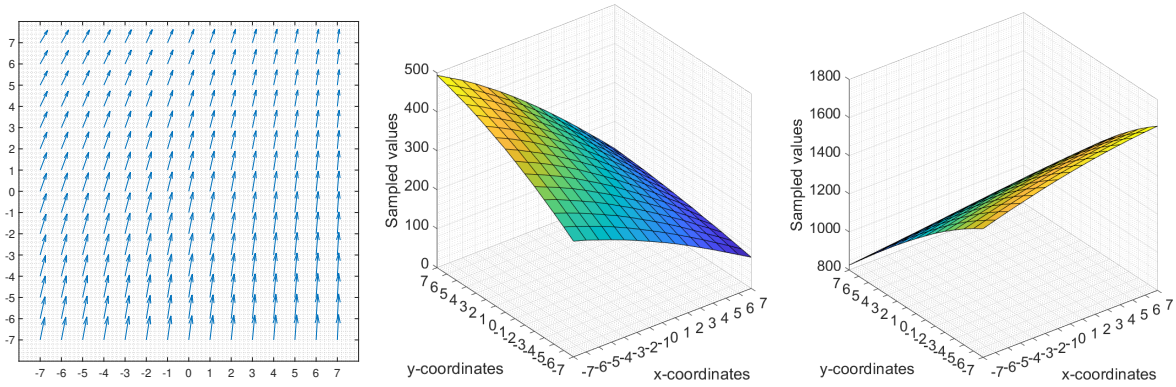
Extreme misalignment case

An extreme misalignment case was applied in this test to examine the aptitude of algorithms to compensate not only marginal mechanical errors but also major misalignments, what is still within the real conditions. The mechanical misalignment parameters of encoder magnet are shown in Table 5.21.

Misalignment case	x-translation [mm]	y-translation [mm]	z-distance [mm]	x-tilt [°]
Extreme case	-1.46	1.59	5	7

Table 5.21: Applied mechanical misalignment conditions of encoder magnets for the angular error correction test: extreme misalignment case.

In Figure 5.46 the vector field and its components in case of applied misalignment of the disc magnet are illustrated.



(a) Field vectors (b) $\mathbf{B_x}$ components (c) $\mathbf{B_y}$ components

Figure 5.46: Investigation of angular error correction algorithm: extreme misalignment case. Field components in applied misalignment case measured by the sensor array at encoder magnet angle = 90°. The AMM point is outside of sensor array plane. Almost **global gradients** are present in (a) and (b). Nevertheless the curvature of the field components is still extractable. *Simulation setup: 15 x 15 array, 1.59 mm x- and 1.46 mm y-translation, 7° x-tilt, 5 mm z-distance, 14 bit ADC, disc encoder magnet.*

Impact of misalignment on the xy-paths and calculated angle are shown in Figure 5.47. The maximum angular error, caused by applied encoder magnet misalignment for all considered algorithms amounts to $\approx 44°$ for the sphere magnet and $\approx 31°$ for the disc magnet.

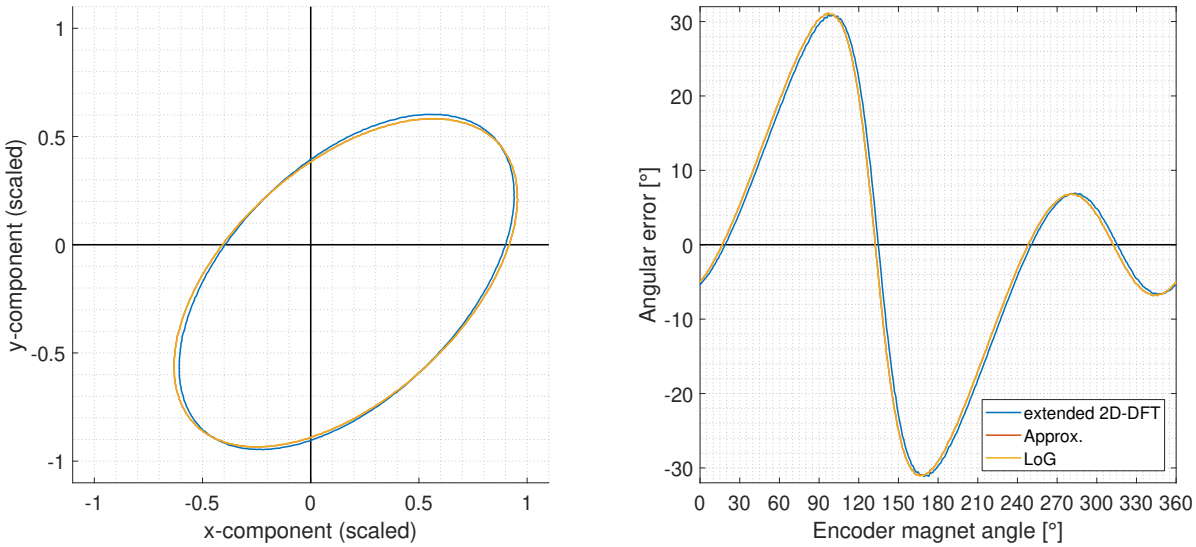


Figure 5.47: Investigation of angular error correction algorithm: misalignment compensation, extreme case. Effects of misalignment. Left: xy-paths of algorithm candidates. Right: corresponding angular errors. *Simulation setup: 15 x 15 array, 1.596 mm x- and 1.463 mm y-translation, 7° x-tilt, 5 mm z-distance, 14 bit ADC, disc encoder magnet.*

In the following, the mean angular error reduction results depending on amount of matching points for both angular error correction algorithms and magnets shapes are discussed. In Figures 5.48 and 5.49 the angular error corrections obtained by homography based algorithm for sphere encoder magnet are illustrated. In Figure 5.50 and 5.51 angular error correction results for both magnets obtained by nonlinear correction is illustrated. The progression of maximum errors and full scale version of the figures can be found in Appendix G.8.2.
Its evident from Figure 5.48 that the angular error correction algorithm based on homography delivers completely sufficient results in case of the sphere magnet. The error is reduced rapidly and there are almost no result differences between the DOFs. Also here the presence of small peaks is visible, due to unfavorable selection of points within the xy-paths as well as quantization noise. Notable is the fact that the extended 2D-DFT approach exhibits a marginal larger error than the LoG and approximation algorithm. The tendency in the error reduction is comparable to a transient settling process. The error can be considered as „settled“ by reaching an amount of matching points of approximately 30. In comparison to the angular error correction process for the sphere magnet the process for the disc magnet looks differently. From Figure 5.49 it is clearly observable that at 6 DOF a large error remains over the total sweep of the amount of matching points. This error gets canceled with an increase of DOF to

8. In general, for both magnet shapes the angular error is getting corrected to $\approx 0.1^\circ$, what is according to results in Appendix G.2 relative close to the noise level at 5 mm z-distance and array size of 15 x 15.

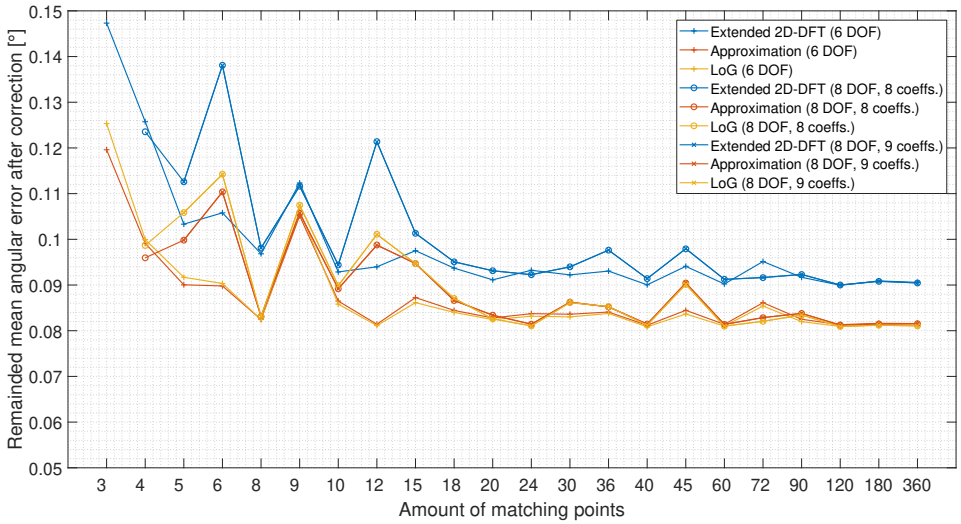


Figure 5.48: Investigation of linear angular error correction algorithm based on homography approach: misalignment compensation, extreme case. Mean angular error. A clear error reduction tendency is apparent. By all DOFs the angular error gets corrected to less than 1°. *Simulation setup: 15 x 15 array, 1.596 mm x- and 1.463 mm y-translation, 7° x-tilt, 5 mm z-distance, 14 bit ADC, sphere encoder magnet.*

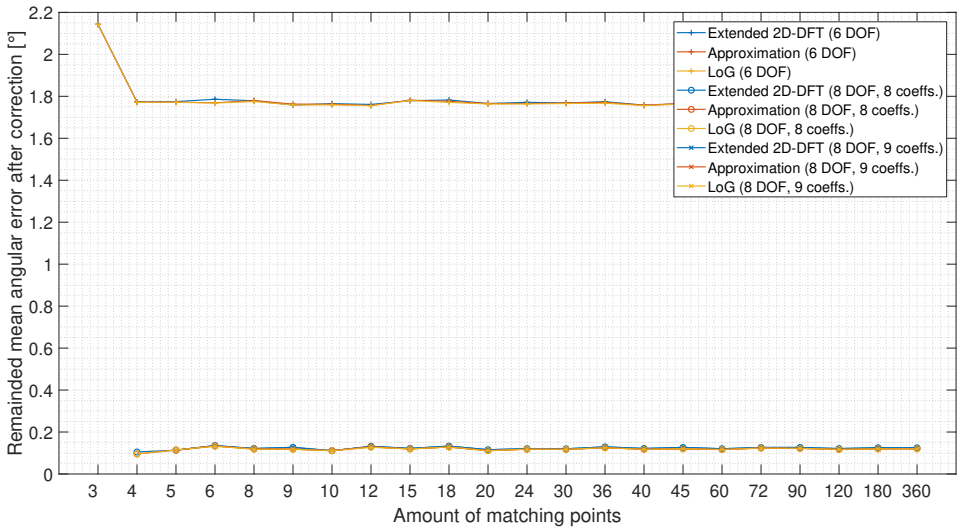


Figure 5.49: Investigation of linear angular error correction algorithm based on homography approach: misalignment compensation, extreme case. DOF 6 is not sufficient to compensate the misalignment effect. The mean angular error stays over full amount of matching points at $\approx 1.8°$. Homography approach of DOF 8 corrects the angular error to $\approx 0.1°$. *Simulation setup: 15 x 15 array, 1.596 mm x- and 1.463 mm y-translation, 7° x-tilt, 5 mm z-distance, 14 bit ADC, disc encoder magnet.*

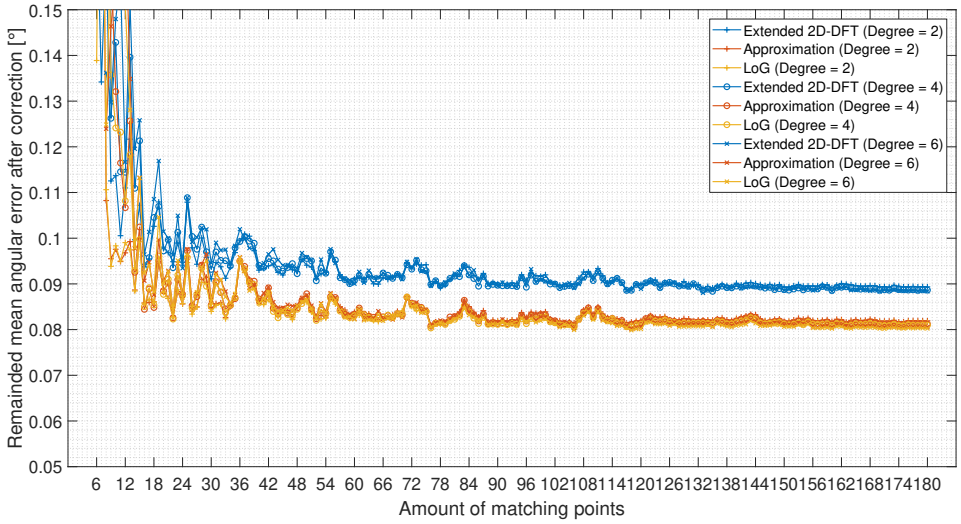


Figure 5.50: Investigation of nonlinear angular error correction algorithm based on geometric distortion cancelation: misalignment compensation, extreme case. Mean angular error. A fast error reduction tendency is apparent. *Simulation setup: 15 x 15 array, 1.596 mm x- and 1.463 mm y-translation, 7° x-tilt, 5 mm z-distance, 14 bit ADC, sphere encoder magnet. Note: the amount of matching points at x-axis is related to one polynomial and needs to be multiplied by 4 for the total amount.*

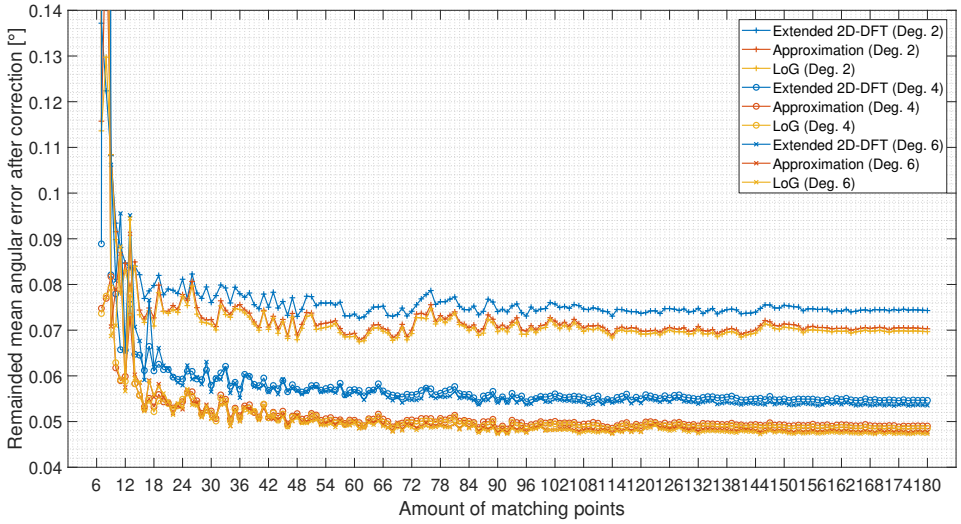


Figure 5.51: Investigation of nonlinear angular error correction algorithm based on geom. dist. cancelation: misalignment compensation, extreme case. A jump of error minimization is observable between degree 2 and 4. In comparison to 5.50 the remained angular error is smaller, due to less quantization effect because of higher field strength of the disc magnet. *Simulation setup: 15 x 15 array, 1.596 mm x- and 1.463 mm y-translation, 7° x-tilt, 5 mm z-distance, 14 bit ADC, disc encoder magnet.*

From the results obtained by application of developed nonlinear correction algorithm in Figure 5.48 it becomes evident that the polynomial degree of 2 is sufficient to cancel the angular error completely until the noise level. Also here an interesting fact is that the error by application of extended 2D-DFT approach is slightly higher that the rest. The angle correction results for the disc magnet look here different as well. The effect observed in Figure 5.49 for the disc magnet becomes here visible as well. In Figure G.44(b) it can be clearly seen that by a polynomial degree of 2 the angular error is not canceled completely. By increasing of polynomial degree to 4 the mean angular error is canceled to a level of around 0.05° and is not getting better by an increase of polynomial degree from 4 to 6. The extended 2D-DFT represent also here a small outlier. Regarding the appropriate amount of matching points according to the error reduction tendency it can be said that for this algorithm the error can be considered as „settled “ by reaching an amount of matching points of approximately 60, so 240 matching points for 4 polynomials in total.

After the error reduction the Fourier analysis as well as THDs calculations of corrected signals of all considered algorithms were carried out in order to measure the reduction of harmonics distortion in total and get the fullest knowledge about the performance of angular error correction algorithms applied for misalignment effect compensation. The direct comparison of THD factors as well as the maximum and mean errors of all algorithms for the sphere encoder magnet are recorded in Table 5.22 and for the disc magnet in Table 5.23. The illustrations of spectrum components are included in Appendix G.8.2.

From Table 5.22 it becomes clear that firstly, the THD factor contained in uncorrected signals is initially very low. The THD becomes slightly larger, due to correction of signal amplitudes. This is clearly visible by contemplation of Figure G.45. Except of the first fundamental frequency there are no other additional harmonics in the signals. The signal amplitudes, offsets and initial phase are corrected by the minimal required degrees of algorithms completely. The mean and maximum angular error are corrected by the minimal required degrees of algorithms to the absolute lowest minimum, from $\approx 44^\circ$ to 0.5° for the maximum error, and from $\approx 19.5^\circ$ to 0.01° for the mean error, what is again, in the near of the normal noise level in this setup.

		Not corr.	**Homography**			**Nonlinear correction**		
			DOF 6	DOF 8 8 coeffs.	DOF 8 9 coeffs.	2nd ord.	4th ord.	6th ord.
THD x-signal [%]	*Ext. 2D-DFT*	0.1252	0.2230	0.2233	0.2233	0.1900	0.1914	0.1942
	LoG	0.1192	0.2120	0.2121	0.2121	0.1744	0.1750	0.1811
	Approx.	0.1193	0.2134	0.2135	0.2135	0.1748	0.1761	0.1813
THD y-signal [%]	*Ext. 2D-DFT*	0.1365	0.2578	0.2571	0.2571	0.2271	0.2268	0.2197
	LoG	0.1196	0.2223	0.2218	0.2218	0.1880	0.1849	0.1878
	Approx.	0.1204	0.2247	0.2243	0.2243	0.1911	0.1883	0.1914
Mean angular error [$^\circ$]	*Ext. 2D-DFT*	19.6889	0.0905	0.0904	0.0904	0.0943	0.0975	0.1001
	LoG	19.4544	0.0811	0.0810	0.0810	0.0849	0.0877	0.0881
	Approx.	19.5065	0.0815	0.0815	0.0815	0.0849	0.0872	0.0878
Max angular error [$^\circ$]	*Ext. 2D-DFT*	44.0549	0.4745	0.4678	0.4681	0.5188	0.5041	0.5452
	LoG	43.4612	0.4002	0.3934	0.3936	0.4190	0.3990	0.4173
	Approx.	43.5705	0.4055	0.3993	0.3996	0.4279	0.4094	0.4286

Table 5.22: Angular error correction algorithms test: Misalignment effect compensation, extreme case. Comparison of angular errors and THDs in % before and after angular error corrections. *Simulation setup: 15 x 15 array, 1.596 mm x- and 1.463 mm y-translation, 7° x-tilt, 5 mm z-distance, 14 bit ADC, sphere encoder magnet. Correction setups: amount of matching points = 181*4 for nonlinear correction, 360 for homography approach.*

In case of disc encoder magnet the starting point looks different. Due to mechanical misalignment of the disc magnet the **second harmonic** arises in the x-/y-signals. This is evident by contemplation of Figure G.48 and is reflected by the calculated initial THDs in Table 5.23. As explained above the homography based approach of DOF 6 is not sufficient to cancel the angular error completely. The THDs become worst in this case. Visible is also that a small THD level remains in the signals corrected by homography based algorithm of DOF 8, what is an indication for occurrence of other harmonics of smaller level what is this this case evident in Figure G.49, and was also already observable in Figure 5.44. In comparison to it, the THDs of signals corrected by nonlinear error correction algorithm are smaller and continue to get smaller by an increase of polynomial degree.

The lowest mean and maximum angular errors achieved by homography of DOF 8 amounts to $\approx 0.12^\circ$ and 0.42°, while by nonlinear angular error correction to $\approx 0.05^\circ$ and 0.25°.

		Not corr.	**Homography**			**Nonlinear correction**		
			DOF 6	DOF 8 8 coeffs.	DOF 8 9 coeffs.	2nd ord.	4th ord.	6th ord.
THD x-signal [%]	*Ext. 2D-DFT*	3.6829	4.0164	0.3002	0.3002	0.1554	0.1201	0.1193
	LoG	3.6539	4.0208	0.2937	0.2936	0.1469	0.1106	0.1098
	Approx.	3.6598	4.0287	0.2945	0.2944	0.1482	0.1109	0.1098
THD y-signal [%]	*Ext. 2D-DFT*	3.5269	4.0174	0.3185	0.3185	0.1559	0.1142	0.1104
	LoG	3.5580	3.9965	0.3012	0.3012	0.1508	0.1063	0.1036
	Approx.	3.5626	4.0030	0.3015	0.3014	0.1518	0.1069	0.1036
Mean angular error [$^\circ$]	*Ext. 2D-DFT*	13.6160	1.7669	0.1246	0.1248	0.078	0.0593	0.0597
	LoG	13.4989	1.7625	0.1200	0.1202	0.0735	0.0538	0.0535
	Approx.	13.5105	1.7654	0.1202	0.1203	0.0749	0.0545	0.0541
Max angular error [$^\circ$]	*Ext. 2D-DFT*	31.5769	2.7888	0.4379	0.4379	0.3279	0.2687	0.2579
	LoG	31.1619	2.7530	0.4161	0.4162	0.3768	0.2793	0.2567
	Approx.	31.2117	2.7461	0.4147	0.4147	0.3603	0.2666	0.2471

Table 5.23: Angular error correction algorithms test: Misalignment effect compensation, extreme case. Comparison of angular errors and THDs in % before and after angular error corrections. *Simulation setup: 15 x 15 array, 1.596 mm x- and 1.463 mm y-translation, 7° x-tilt, 5 mm z-distance, 14 bit ADC, disc encoder magnet. Corrections setups: amount of matching points = 181*4 for nonlinear correction (Degree 4), 360 for homography approach (DOF 8).*

From the performed investigation of angular error correction algorithms applied for misalignment compensation multiple important conclusions are derived:

- Due to the misalignment in case of disc encoder magnet, additional harmonics occur in the output signals.
- Associated with the point above the occurred harmonics are not cancelable by DOF of homography approach equal to 6. Homography approach of DOF 8 cancels the harmonics significantly but not completely.
- No harmonics occur by use of a sphere encoder magnet at all. This is attributable to the „perfect geometric shape“ of the sphere magnet.
- Associated with the point above, minimal required degrees of algorithms are completely sufficient to achieve the absolute lowest minimum of angular error what is in the range of quantization noise.

The above mentioned conclusion lead to another important conclusion that in case of using a system consisting of a sphere encoder magnet less computational effort is required to obtain the angular error correction.
For summarizing, the uncorrected and corrected xy-paths obtained from the LoG algorithm

and the corresponding angular error improvement for homography of DOF 8 and geometric distortion cancelation based algorithms of polynomial degree of 4 are shown in Figure 5.52. It is manifestly visible that for the homography approach a tiny superimposed wave is present in the angular error progression. In comparison to it, a total angular error compensation is obtained by the nonlinear algorithm. The possible discontinuity points disappeared in the quantization noise level completely and are not visible at all.

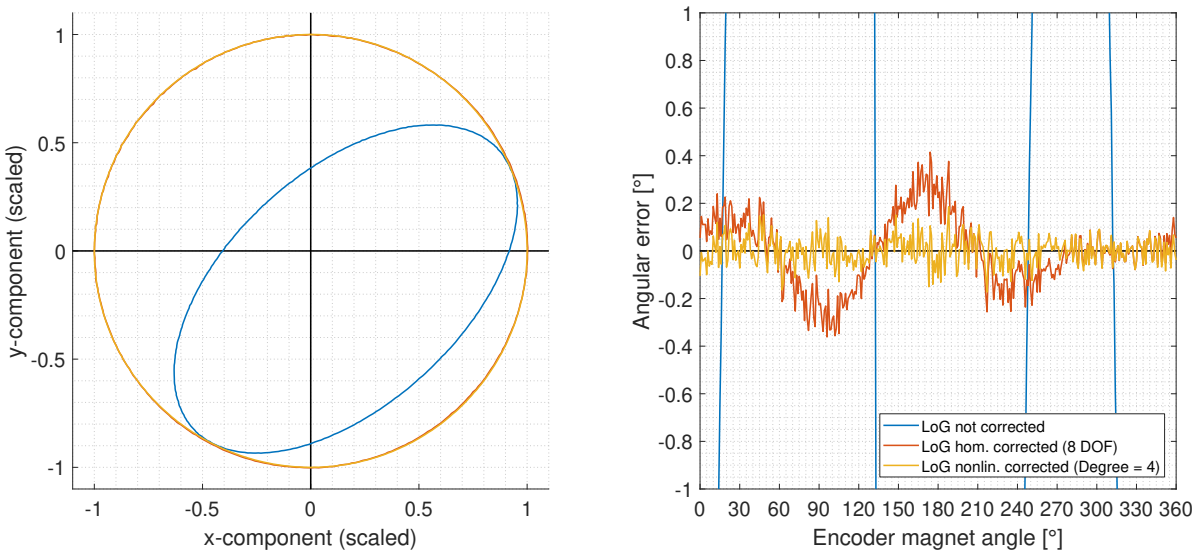


Figure 5.52: Investigation of angular error correction algorithms: misalignment effect compensation, extreme case. Left: xy-paths of uncorrected and corrected LoG algorithm. Right: corresponding angular errors. Both methods exhibits sufficient angular error correction performance to meet of angular accuracy requirement of $\varphi_{err} \leq 1^{\circ}$. However, the nonlinear angular error correction algorithm exhibit better performance and cancels the angular error completely. Mentionable is also the noise progressions correlation of both algorithms what leads to the fact that the discontinuities points of the nonlinear algorithm are negligible small. *Simulation setup: 15 x 15 array, 1.596 mm x- and 1.463 mm y-translation, 7° x-tilt, 5 mm z-distance, 14 bit ADC, disc encoder magnet. Correction setups: amount of matching points = 181*4 for nonlinear correction, 360 for homography approach.*

worst SOA misalignment case

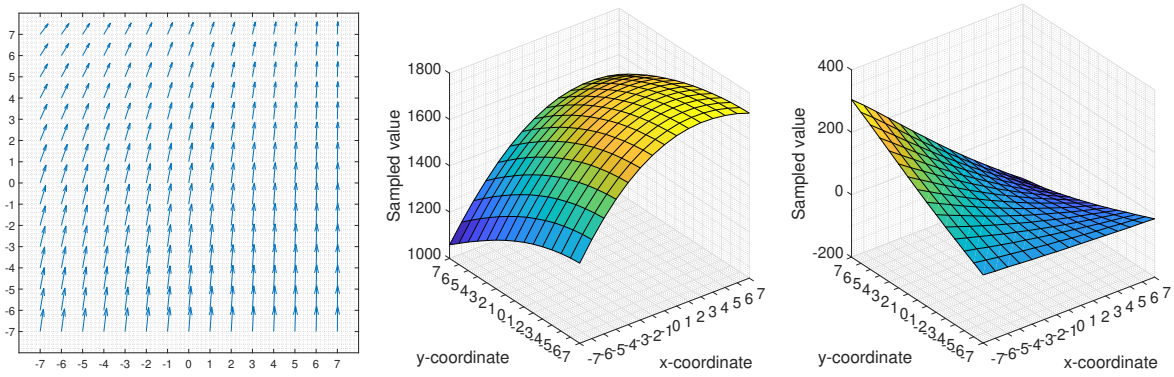
Due to analysis results carried out in the previous investigation this part covers the results of the angular error correction for the disc magnet only. For more discussion and evaluation of results for both magnets see Appendix G.8.3.

As well as in the previous test, the homography based algorithm and nonlinear approach were tested in this investigation. The ADC resolutions was set to 12 bit. The tests were performed in analogy the previous. The misalignment parameters of encoder magnet in case of worst Safe Operating Area case are shown in Table 5.24.

Misalignment case	x-translation [mm]	y-translation [mm]	z-distance [mm]	x-tilt [°]
Worst SOA case	-1	1	2	5

Table 5.24: Applied mechanical misalignment conditions of encoder magnets for the angular error correction test: worst SOA misalignment case.

In Figure 5.53 the vector field and its components in case of applied misalignment of the disc magnet are illustrated.



(a) Field vectors (b) $\mathbf{B_x}$ components (c) $\mathbf{B_y}$ components

Figure 5.53: Investigation of angular error correction algorithm: misalignment compensation, worst SOA case. Field components in applied misalignment case measured by the sensor array at encoder magnet angle = 90°. The AMM point is at the corner of sensor array plane. Presence of large global gradient is visible in (a) and (b) as well. *Simulation setup: 15 x 15 array, -1.06 mm x- and 1.06 mm y-translation, 5° x-tilt, 2 mm z-distance, 12 bit ADC, disc encoder magnet.*

Impact of worst encoder magnet misalignment within the SOA on the xy-paths and the angle calculation are shown in Figure 5.54. The maximum angular error, caused by applied encoder magnet misalignment for all considered algorithms amounts to $\approx 70°$ for the sphere magnet and $\approx 10°$ for the disc magnet, see Figures G.62 and G.57 in Appendix. In Figure G.57 a clear impact of harmonics is visible in the progression of x- and y- signal).
The corrected signals are depicted in Figure 5.55. For homography angular error correction method 8 DOF with 8 coefficient were used, for the nonlinear correction the 2D polynomials of 4th degree. The amount of matching points was set to maximum. By application of angular error correction algorithms the error caused by the worst misalignment case within the SOA reduces to approximately 0.6° for the homography based algorithm of DOF 8 and to 0.1° in case of using the nonlinear angular error correction algorithm of 4th polynomial degree.

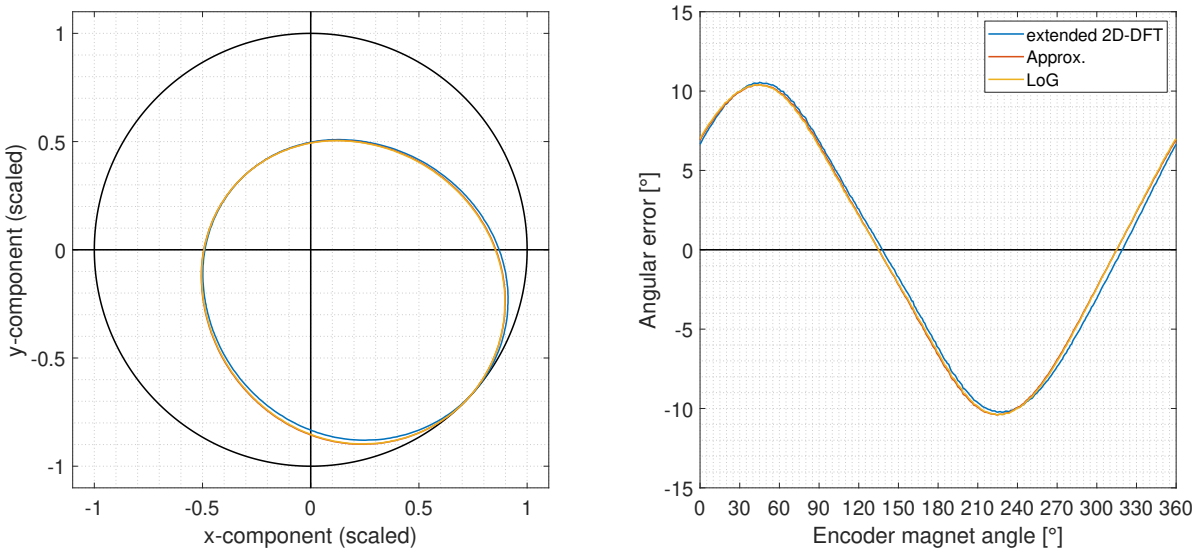


Figure 5.54: Investigation of angular error correction algorithm:misalignment compensation, worst SOA case. Left: xy-paths of algorithm candidates. Right: corresponding angular errors. *Simulation setup: 15 x 15 array, 1 mm x- and 1 mm y-translation, 5° x-tilt, 2 mm z-distance, 12 bit ADC, disc encoder magnet.*

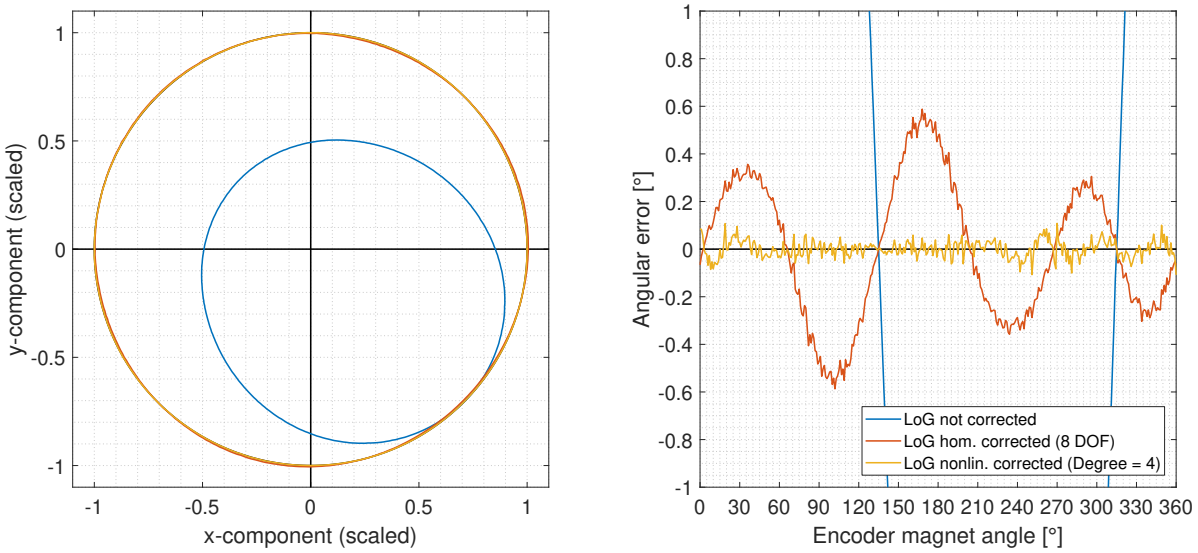


Figure 5.55: Investigation of angular error correction algorithms: misalignment compensation, worst SOA case. Left: xy-paths of uncorrected and corrected LoG algorithm. Right: corresponding angular errors. Both methods exhibits sufficient angular error correction performance for the fulfillment of angular accuracy requirement. Also here the nonlinear angular error correction algorithm exhibit a better performance and cancels the angular error completely. The discontinuities points of the nonlinear algorithm are negligible small as well. *Simulation setup: 15 x 15 array, -1 mm x- and 1 mm y-translation, 5° x-tilt, 2 mm z-distance, 12 bit ADC, disc encoder magnet. Corrections setups: amount of matching points = 181*4 for nonlinear correction (Degree 4), 360 for homography approach (DOF 8).*

Figures 5.56 and 5.57 show a comparison of DFT analysis results of uncorrected and corrected x-/y-signals. In this misalignment case, a **group of harmonics** occurred in the extracted x-/y-signals from the sensor array, which are canceled by applied algorithms differently:

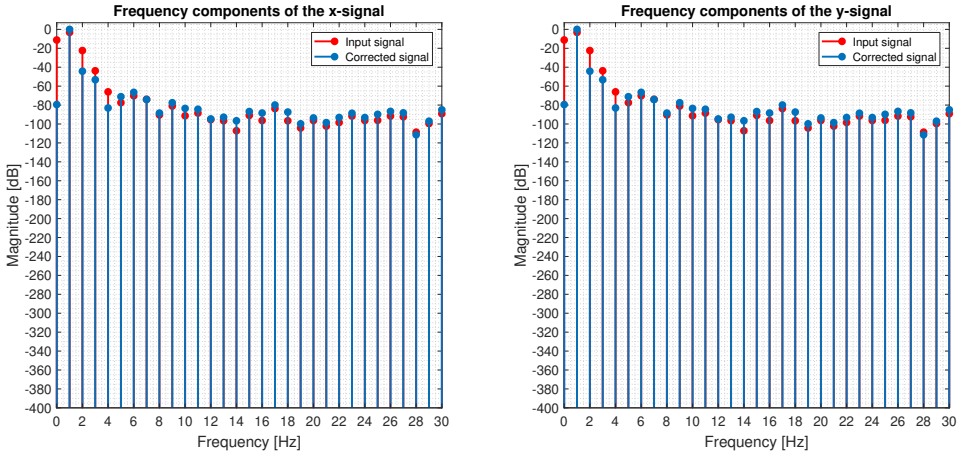


Figure 5.56: Investigation of linear angular error correction algorithm based on homography: cancelation of harmonics resulting from the misalignment of disc magnet. 2nd, 3rd and 4th harmonics are decreased. The noise level is increased only slightly due the correction of the amplitudes of the signals. *Correction setup: DOF = 8, amount of matching points = 360*.

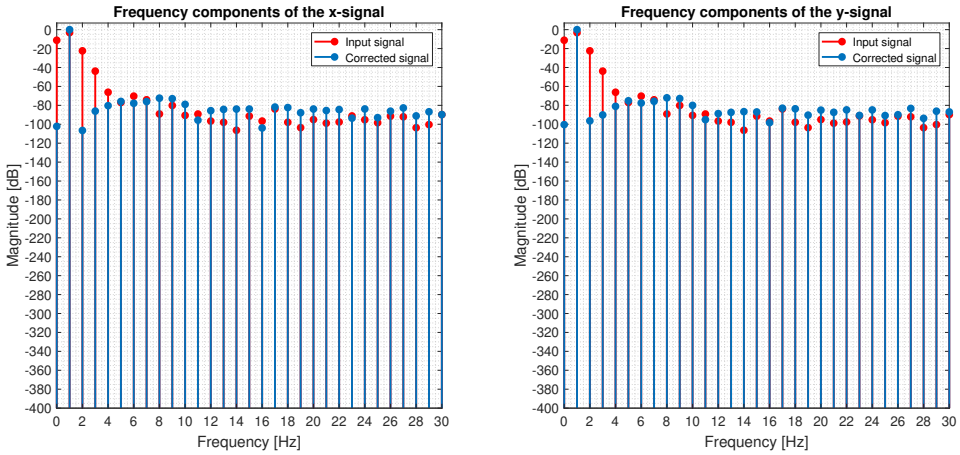


Figure 5.57: Investigation of nonlinear angular error correction algorithm based on geometric distortion compensation: cancelation of harmonics resulting from the misalignment of disc magnet. 2nd, 3rd and 4th and 6h occurred harmonics are suppressed stronger than by using the homography based algorithm. The noise level is more elevated due to amplitudes correction as well as discontinuities issued, but still under approximately 70 dB. *Correction setup: Degree = 4, amount of matching points = 181*4*.

As well as in case of the extreme misalignment test, the comparison of THDs and the maximum and mean errors of all algorithms were determined. The values are recorded in Table 5.25.

		Not corr.	Homography			Nonlinear correction		
			DOF 6	DOF 8 8 coeffs.	DOF 8 9 coeffs.	2nd ord.	4th ord.	6th ord.
THD x-signal [%]	*Ext. 2D-DFT*	7.5786	9.1414	0.5505	0.5513	0.1600	0.0720	0.0421
	LoG	7.6909	9.2265	0.5659	0.5662	0.1749	0.0624	0.0375
	Approx.	7.7010	9.2481	0.5529	0.5532	0.1738	0.0646	0.0378
THD y-signal [%]	*Ext. 2D-DFT*	7.8222	9.3611	0.5657	0.5647	0.2131	0.0688	0.0425
	LoG	7.6909	9.2265	0.5659	0.5653	0.1735	0.0626	0.0380
	Approx.	7.7010	9.2481	0.5529	0.5523	0.1724	0.0650	0.0384
Mean angular error [°]	*Ext. 2D-DFT*	6.2654	6.0913	0.6034	0.2521	0.3533	0.0321	0.0250
	LoG	6.2722	5.9805	0.6029	0.2561	0.3173	0.0276	0.0218
	Approx.	6.2615	6.0088	0.5872	0.2505	0.3224	0.0283	0.0222
Max angular error [°]	*Ext. 2D-DFT*	10.5382	6.0913	0.6034	0.6026	0.3533	0.1311	0.1011
	LoG	10.4102	5.9805	0.6029	0.6023	0.3173	0.1114	0.0896
	Approx.	10.4070	6.0088	0.5872	0.5866	0.3224	0.1216	0.0868

Table 5.25: Angular error correction algorithms test: Misalignment effect compensation, worst SOA case. Comparison of angular errors and THDs in % before and after angular error corrections. *Simulation setup: 15 x 15 array, 1 mm x- and 1 mm y-translation, 5° x-tilt, 2 mm z-distance, 12 bit ADC, disc encoder magnet. Corrections setups: amount of matching points = 181*4 for nonlinear correction (Degree 4), 360 for homography approach (DOF 8).*

In Appendix G.8.3 the mean and maximum angular error reduction results depending on amount of matching points as well as DFT analysis results of uncorrected and corrected x- and y-signals for both angular error correction algorithms are shown. The tendency of the error reduction depending on the amount of matching points is similar the the previous investigation. Extremely good reduction of harmonics using nonlinear angular error correction algorithm is achieved by a polynomial degree of 6.

Conclusions: Indeed, the developed angular error correction algorithms represent really powerful methods. The suitability of the angular error correction algorithms was proved on an extreme as well as the worst SOA mechanical misalignment case of the encoder magnets. The gained knowledge about the algorithm performance as well as the derived conclusions were confirmed by both investigations.
Both angular error correction methods exhibit sufficient angular error correction potential for the obtain of angular accuracy $\varphi_{err} \leq 1°$. By use of sphere encoder magnet system a perfect correction of misalignment effect can be obtained already by minimum required degrees, DOF 6 for homography based algorithm and polynomial degree of 2 for geometric distortion based algorithm. In case of application of disc encoder magnet, additional harmonic frequencies occur in the x-/y-signals what cannot be completely corrected by homography approach, but by nonlinear approach. For the homography approach no significant enhancement of angular error reduction tendency is observed from an amount of total matching points ≥ 30 and for nonlinear angular error correction algorithm $\geq 60 \cdot 4$.

5.8 Results summary, discussions and proposals

This section provides a summary of the obtained results from conducted investigations. Furthermore, the results are shortly discussed, proposals for application and implementation of algorithms are given and the set questions for this thesis are answered.

As a preliminary investigation of the system the gradients behavior of considered encoder magnets field were examined and important conclusions related to the acquisition of data were derived.
With regard to possible preprocessing step suitable candidates for this purpose were tested and their impact on deformation of measured field components associated with filtering was examined. According to the present problem appropriate filter parameters were proposed.
Furthermore, within the framework of optimizations the noise sensitivity of the LoG approach was investigated and its optimal bandwidths determined. It was found out that the excessive σ parameter leads to the identical noise robustness as the approximation algorithm.

In the context of performed investigations of algorithms for angle informations extraction and angular error correction important conclusions not only for the implementation proposal but also about system behavior and associated successions in the entirety were derived:
By use of inhomogeneous encoder magnet system and construction of a sensor array it will be indeed feasible to rely on the concept of curvature extraction for angle calculation and suppression of homogeneous field part for fulfillment of stray field suppression requirement. The concept will work not only for a sphere encoder magnet but also for in the series production present disc encoder magnet of a diameter of 6 mm and height of 4 mm. The simulated disc encoder magnet exhibits a field inhomogeneity which is sufficient for the field curvature based angle calculation. In accordance to SOA limitations by using a 12 bit ADC a sensor array size of 7 x 7 elements will be needed to meet the angular accuracy requirement of $\varphi_{err} \leq 1^{\circ}$. By use of a 14 bit ADC resolution the concept becomes applicable also for a theoretical minimum required sensor array size of 3 x 3 elements.
In addition to it, a clear border in the physical signal quality measured in signal-to-noise ratio in accordance to requirement for angular accuracy was found out. Within investigated array sizes of 3 x 3 to 15 x 15 elements the quality of sampled signals must lay between 70 and 60 dB SNR.
Furthermore, the impact of limitation of register widths of arithmetic units responsible for arithmetic operations was investigated, where the single LoG approach was taken into consideration. For the set ADC configuration of 12 bit and s1Q10 input value format which is used as a standard within the project, it was proved, that in that case the multiplier units must be capable to perform multiplications of values represented in two bits more than the sampled value at least, in order to meet the angular accuracy requirement.
An impact of sensor electronics tolerances was examined as well. With assumed mismatches for gain and offset deviations of operational amplifiers it becomes impossible to meet the angular accuracy requirement of $\varphi_{err} \leq 1^{\circ}$. Hence, a calibration procedure of individual sensor

array elements is definitely needed.
Acceptable limits for the mechanical misalignment of encoder magnet were found out, where the angular accuracy requirement without additional teach-in procedures is complied. The misalignment limits are modestly small. Depending on z-distance and tilt of the encoder magnet the acceptable range for x-/y-translation of the encoder magnet is between $\pm$ 0.13 mm and 0.4 mm.
By application of tracking algorithm extension the angular error caused by worst misalignment of encoder magnet within the SOA limitations could be maximally reduced by 94 %, what represents the maximum possible increase of angular accuracy in combination with teach-in phase in spatial domain within the examined sensor array size limits. Nevertheless, the remained angular error laid above the angular accuracy requirement of $\varphi_{err} \leq 1^{\circ}$. This leads to conclusion that the subsequent angular teach-in procedures within the set SOA limitations are absolutely necessary and resulting processing delay for angular error correction needs to be accepted.
The suitability of angular error correction algorithms associated with the angular teach-in procedure for the calibration phase in the final test as well as misalignment effect compensation at the EOL was proved. The angular error correction algorithms exhibited promising results with respect to the SOA and investigated harmonic distortion case. The angular accuracy requirement of $\varphi_{err} \leq 1^{\circ}$ is achieved by application of both methods. Homography based as well as nonlinear angular error correction algorithms corrected the misalignment effects of the sphere magnet completely and with the lowest computational effort. For the disc magnet the homography method exhibits a minor weakness with regard to cancelations of harmonics associated with observed effect of its mechanical misalignment. However, the performance of this method for this purpose was satisfied (DOF 8). It is powerful enough to reduce the angular error caused by mechanical misalignment of the encoder magnet and reduce the impact of harmonics. Nonlinear angular error correction algorithm based on geometric distortion cancelation behaved much more powerful and is able to correct the misalignment effects and the compensate the harmonics distortions almost completely. The discontinuities disadvantage of this method played a lower role. Its constant noise level associated with discontinuities in the corrected x- and y- signal lay in an SNR range of 70 dB - 80 dB (degree 4), which is outside the range of practical interest in the present case.

For the angle information extraction including the suppression of stray field all considered candidates in the main investigations: extended 2D-DFT, LoG convolution, single LoG and approximation showed very similar results and passed through all the tests. LoG convolution approach did not exhibit a better performance than the single LoG method and therefore was not considered in the extended tests due to its methodical similarity to the single LoG approach.

5.8.1 Algorithms implementation proposal

In the following, based on gained knowledge from investigations results, a proposal for algorithm implementations is given.

Angle information extraction

The fact that the extended 2D-DFT, single LoG and approximation approaches show identical angle calculation accuracy and noise stability results leads to the following decision: The decision for the proposal is made due to the minimum necessary amount of multiplications, what represents the implementation effort and computation complexity on the ASIC. Therefore, for the angle information extraction including stray field suppression it is proposed to use the single LoG approach and implement a state-machine for element-wise multiplication with coefficients of LoG mask of smallest bandwidth. For an array consisting of NxN sensor elements the amount of multiplications required for extraction of x- and y- signal for the subsequent angle calculation in the CORDIC counts to $2N^2$.

Tracking algorithm

For the localization of the point of maximum field strength within the array and make a use of algorithmic extension associated with the tracking method and movement of angle information extraction mask, it is proposed to implement the approximation algorithm including the reinsertion of sensor plane coordinates described in Section 3.26. This algorithm exhibits the best localization accuracy and robustness against noise interfering and is considered as the best suitable candidate for this purpose (see Section 5.7.3).
If it is desired to make a use of tracking extension and use a movable angle information extraction mask to compensate possible misalignment effects already in the spatial domain, it is suggested to keep the extraction mask size of $\geq$ 5 x 5 elements due to a significant impact of noise interference, according to results in Section 5.7.4.

Angular error correction

It was proved that the linear angular error correction algorithm based on homography approach of DOF 8 and 8 coefficients is sufficient to correct the angular errors caused by worst misalignment case within the SOA as well as extremely cases what are in range of practical relevance. In comparison to the nonlinear angular error correction algorithm it requires less implementation as well as computation effort. Hence, the linear angular error correction algorithm based on homography method is proposed as a most suitable candidate for implementation in accordance to the present problem. It is proposed to keep the amount of matching points greater than 30 in order to achieve a robust estimation and and a „settling“ of the angular error correction process, according to the observed results in Section 5.7.5. With respect to the application of this method for the present problem it was proved that there are absolutely no differences in

results obtained by DOF 8 with 8 and 9 coefficients of the H matrix. Therefore, no implementation of SVD algorithm is needed for the maximum applicability of this method. A standard least squares solver is fully satisfied. However, it is nevertheless proposed to implement the SVD algorithm[9], due to its numerical stability and well-conditioning [65]. In accordance to the concept described in Section 3.2.1, the teach-in phases are not time critical and therefore the required computational effort does not play a role in this proposal.
If it is aimed to get the maximum performance of angular accuracy of the system it is proposed to implement the nonlinear angular error correction algorithm based on geometric distortion compensation with 2D polynomials of 4th degree (8 coefficients). The amount of matching points for the teach-in operation shall be greater than 60 points for each 2D polynomial, so 240 matching points in total, in accordance to the results in Section 5.7.5. It needs to be mentioned that the implementation of this algorithm requires in comparison to the homography based algorithm a much more complex state machine, teach-in and operation procedure due to the splitting procedure of the operational xy-plane.

5.8.2 Proposal for encoder system

In the context of performed investigations it was shown in Section 5.7.5 that by application of a disc encoder magnet system the x-/y-signals extracted from the sensor array for the angle calculation are getting distorted by higher harmonics if a mechanical misalignment is existing. Therefore, to reduce the required computational effort needed for the subsequent compensation of misalignment effect and cancelation of harmonic distortions it is proposed to use exclusively a sphere encoder magnet system. In this case, according to performed investigations, all the angular errors caused by misalignment are completely cancelable by angular error correction algorithm based on homography of DOF 6.

[9] See more in Appendix B.2.3.

6 Summary and outlook

6.1 Summary

In the context of this thesis, signal processing algorithms for magnetic sensor arrays based on methods from image processing and approximation of the magnetic field were developed. Their practical feasibility has been confirmed based on simulation of a 1:1 replicated encoder magnet - sensor array system with assumed ideal sensor behavior.
In the beginning a deep understanding about the physical structure and functionality of magnetic sensors in automotive applications was provided and the requirements introduced.
Thereafter, a processing concept was introduced, consisting of a functional split into a teach-in phase and an operational phase. Algorithms for angle information extraction with included stray field suppression were developed based on knowledge gained from previous project work [75], [52], [25]. Two angular error correction methods for compensation of mechanical misalignment effects of the encoder magnet were developed: a) a spatial domain based method which implies a movement of the angle information extraction area within the sensor plane after localization of the projection point of the rotational encoder magnet axis and b) a spatial domain independent method which requires the application of additional rectification functions and angular teach-in procedure. Two algorithms can be applied for the second angular error correction method: a) a linear angular error correction algorithm based on projective transformation called homography and b) a nonlinear algorithm derived from geometric distortion compensation. All the formulations of estimations procedures, mathematical foundations and signal processing flows needed for the implementation of the algorithms were documented. The developed teach-in procedures were documented as well.
For the test and evaluation of algorithm results an evaluation program was written that can be used and extended as future work. A broad test plan was created based on the Safe Operating Area of the sensor and the investigation were carried out with full accordance to the plan. The capability of algorithms under real case conditions was investigated. Based on the obtained results, proposal for implementation and application of algorithms were provided.
A signal processing chain for a sensor array system was designed, consisting of misalignment and stray field detection, stray field suppression, angle calculation based on resolution of the spatial domain and system accuracy enhancement in combination with a teach-in phase.
By application of angular error correction algorithms, a significant improvement of system accuracy is achievable. Under assumptions for the simulation, it was shown that by use of a teach-in procedure, the angular accuracy requirement within the Safe Operating Area limitations of the sensor can be met.

To sum up, the most important questions regarding the feasibility were answered and in addition to it, important thoughts and tips for further work were given.

6.1.1 Results of algorithms

The direct extraction of curvature from the sensor plane won the challenge for the angle calculation. Extraction of the curvature in all directions by use of LoG filter with exceeded σ parameter delivered the same results as the approximation and evaluation of curvature coefficients of the 2D polynomial basis functions as well as the extended 2D-DFT approach. At the same time the direct curvature extraction represents the simplest way for the angle information extraction and requires the lowest computational effort. Notable is the fact that all considered algorithm candidates except the modified approximation approach exhibited the same angular accuracy. From the modified approach a better angular accuracy was expected. Unfortunately, the expectations were not met due to its significantly high noise sensitivity.
Approximation based algorithms exhibited the best robustness for the localization of projection point of the encoder magnet rotational axis. Their gradient based competitors did not show satisfied results.
The developed angular error correction algorithms represent very powerful features and are applicable for correction of misalignment effects as well as pre-calibration and reduction of total harmonic distortion.

6.1.2 Contribution to the project

The main principle used for angle information extraction described in Section 3.3.1 corresponds to the principles of algorithms developed by K. Jünemann and H. Pape [75], [52]. However, it was derived from another point of view.
Attempts to gain the information about the angle from 2D-DFT spatial frequencies were already carried out in the beginning of the ISAR Project by K-R. Riemschneider et al. [25], but not continued further. The evaluation of spatial frequencies for angle calculation was taken up again in the context of this work and has led to the clear goal for the angle calculation, but only in combination with dynamic global gradient correction[1] described in Section 3.5.2.
New was the application of image processing filters in spatial domain, e.g. the LoG filter for curvature extraction and the gradient filters for the localization of AMM points. The application of field approximation methods for the localization of AMM points was novel as well. Completely new was also the application of image processing algorithm on the virtual plane resulting from the paths of x-/y-signals of the sensor for the correction of angular errors. The applied projective geometry methods using homogeneous coordinates was split from the proposed approach of forward and inverse kinematics by K-R. Riemschneider and T. Schüthe [90].
The proposed 2D-DFT filtering algorithm including 2D-IDFT [81] was included into the evaluation software, despite the fact that it had no conceptional match to the proposed processing

[1] Also denoted as realignment of field components.

approach. It took a lot of effort to couple this approach into the software without a real added value for this thesis because it was not documented due to the unsatisfactory results.
Likewise the 2D-DFT approach, the consideration of the field weakening behavior by its inclusion into the approximation algorithm and the associated investigation as well as optimization were extremely time consuming.
Furthermore, an attempt for feature extraction from the spatial frequencies of 2D-DFT was carried out. Some results were found out but were also not included into this thesis.
In addition, a sensor system operation in the saturation mode was constructed in the software. An algorithm for stray field suppression was developed based on template matching with „origin magnetic picture“. Under the applied simulation conditions, the robustness of this algorithm was not sufficient and therefore not considered in this thesis.
Furthermore, hardware for demonstration purposes was designed for this thesis: a linear reference sensor array system consisting of 8 x 8 Hall sensors[2], shown in Appendix H. The acquisition of sensor signals by application of a additional module has been realized. Also, a connection to the terminal program has already been established. In addition, a demonstrator system consisting of nonmagnetic materials was constructed. Due to time constraints, the developed system could not be taken into the operation.

6.1.3 Obstacles and difficulties

The creation of evaluation software and the GUI was extremely time consuming due to a large grade of dependencies between the objects and functions.
Also, handling of boundary conditions for data extraction from the data arrays, i.e. the coordinate grids adaption, scalings depending on position of extraction masks etc. was a complex procedure.
A large part of this work was based on graphical evaluation and visual explanations. The creation of graphs and diagrams was extremely time consuming.
Furthermore, it was necessary to evaluate a huge amount of complex data arrays of multiple dimensions. The programming of the script files for performing of specified tests with parameter sweeps and results evaluation took a lot of effort.

[2] MLX90380LDC-BAB-100-SP-ND [26]

6.2 Outlook

6.2.1 Independent proof of results

The first short-term task which needs to be performed within the framework of project continuation is an independent proof of all obtained algorithm results.
In addition, the used magnetic simulator [75] shall be completely validated. For this purpose, the data of a FEM / FDM magnetic simulation shall be used, to make sure that the field data, especially of misaligned constructed magnets, are represented correctly.

6.2.2 Real data investigations

The algorithms tested in the main investigations have shown satisfying results in the simulation. However, this does not mean that the algorithms will behave in the same way when applied to real sensor array data. The physical constrains of sensor elements will definitely play a significant impact on the achievable angular accuracy. Therefore, the feasibility of the algorithms shall be tested on real measurement data, e.g. from a demonstrator. The developed linear reference system shown in Appendix H can be used for this purpose in the first step.

6.2.3 Software extension

The fixed-point arithmetic feature by coupling of functions from the MATLAB® Fixed-Point Designer Toolbox™ was not implemented completely due to time constraints. Only the front-end and the LoG method include this feature. Hence, in order to test the angular accuracy associated with the arithmetic operation limitations of the other algorithms and investigate the required register widths for estimation procedures, this functionality shall be fully implemented.

Furthermore, as a medium-term goal, a physical sensor model could be coupled into the software in order to test the algorithms on a more closely real data. A Stoner-Wohlfarth model or coupling of the field data into the micromagnetic[3] simulation where the xMR sensor array elements are represented on the micro scale level can be coupled into the software.

6.2.4 Algorithm extensions and improvements

Approximation algorithm

Additionally, for the angle information extraction algorithm based on approximation of magnetic field components the coefficients of chosen 2D polynomial functions which describe the xy-dependency[4] can be considered for the angle calculation, as already pointed to it in Section

[3] e.g. MicroMagnum, a freeware FDM micromagnetic simulator.
http://micromagnum.informatik.uni-hamburg.de/

[4] Coefficient p_{x3} of the function $f_{B_x}(x, y)$, and p_{y3} of function $f_{B_y}(x, y)$.

3.3.3. The coefficients can be included in the angle calculation by their mutual crossed consideration[5]. The consideration of the coefficients representing the xy-dependency may have a positive impact on angular accuracy and stability against noise, especially due the increased amplitude in that case. This finds the correspondence to the components of dipole moment in x- and y-direction [52], [75]. All moments would be fully considered in this case. This extension shall be tested with high priority within the context of project continuation.

2D-DFT approach

The spatial frequencies components at 45° (coordinate u = 1, v = 1) should be taken into account, in order to get a higher amplitude and derive more information about the measured field (see Figure 3.8). These frequencies can be taken into the calculation of the angle in arctangent function by multiplication by -1 and their reciprocative crossed coupling from 2D-DFT results. An investigation needs to be performed to determine how their initial offset and phase shift will affect the angle calculation. It might be possible that their impacts cancel themselves during angle calculation.

Nonlinear angular error correction method

For the nonlinear angular error correction algorithm, an improvement as well as the consideration of continuous models are proposed.

- **Improvement of the developed algorithm**
  The split of the operational plane represents the disadvantage of this algorithm. Its practical applicability shall be investigated very carefully. Indeed, the splitting of the plane may cause a problem if the conditions detailed in Section 3.7.2 are not met. Thus, it is proposed to improve this algorithm by consideration of additional transition points for distinguishing between operational areas. Alternatively, the case statement can be performed depending on the uncorrected angle after the offset correction of x- and y-signals. Therefore, the angle of the uncorrected x-/y-signals needs to be calculated in each step before the use of the rectification functions. In this way, the assignment of operational planes will be completely unambiguous.
  Moreover, one needs to ensure that the points at discontinuities areas are precisely sampled and not missed. To decrease the effect of discontinuities either even more points at these areas needs to be included in the estimation or weighting factors used.
- **Application of other distortion correction models**
  Furthermore, it is suggested to use models consisting of continuous functions, and couple them exclusively with the projective transformation based on homography as the preliminary correction. The Brown-Conrady model [94] which corrects both the radial and tangential distortions, is proposed as an appropriate candidate for this extension. Due

[5] I.e. for the x component the coefficients p_{y3}, p_{x4}, p_{x5}, and for the y component the coefficients p_{x3}, p_{y4}, p_{y5}.

to its property it is expected that this method will exhibit a more than satisfying result as well.

For the practical realization of determination of rectification function coefficients, the estimation methods proposed by Z. Zhang in [97] and R. Hartley [48] shall be considered.

6.2.5 Further algorithm approaches

Template Matching based approaches

In case of a sensor system operating in the saturation mode, exclusively the direction of local vectors resulting from the measured components in the xy-plane is getting changed if the sensor array is exposed to an external magnetic field. Due to this, the presence of a stray field will lead to a deformation of the shapes of the measured field. Therefore, the algorithms based on linear systems won't work in this case[6].
During this work a development of an algorithm for stray field suppression and angle calculation based on template matching with a „magnetic origin picture“ which is applicable for saturated systems was carried out. The template matching is based on correlation where the matching grade is calculated either by the mean square distance or the mean absolute distance [39]. This algorithm computes the differences of field components between the previously defined template and the input data and evaluates the values for the angle calculation which are located at the global minimum in the result picture. The correlation is performed for 360 angular positions, discretized into 1° steps. Unfortunately, this algorithm was not robust enough within the set physical framework and SOA, but it worked. Thus, the question needs to be reformulated to: What are the limits of physical conditions for this algorithm? The template matching idea shall be definitely taken up within the project.

Misalignment handling based on image moments calculations

Another basis for the possible misalignment detection or correction on the spatial domain is the consideration of static features, so called central moments[7]. Such methods are widely used in image processing for rotation independent pattern recognition, shape analysis and feature extraction [78], [43].
Here the information from the image acts as a two-dimensional random variable. In the moments calculation, the gray value intensity as well as the shape of the object in the image is considered. Distribution of the probability mass is used as a measure of existing symmetry [39]. For the present problem the third and fourth moments which described the skewness and kurtosis are of more interest. Also, here the information about expected field symmetries shall be used. For more information, refer to the work Luren Yang and Jan Flusser [43], [66]. Good explanations can also be found in [78, Chap. 11] and [39].

[6] See more in Appendix C.1
[7] Also called Hu moments, related to Ming-Kuei Hu, who described this measures.

Gradiometer principle

Another possibility for information evaluation is to use the gradiometer instead of magnetometer principle and perform a differential read out of the sensor signals with adapted range of the ADC (see discussion in Section 5.1). The determined gradient values within the SOA recorded in Section 5.1 can be used for the configuration of the ADC units. It this way, a much higher part of the dynamic range of the ADCs can be used which will lead to a higher angular accuracy due to a large increase of bit significance. The approaches for angle information extraction needs to be modified and operate on a lower derivative order in this case.

Adaptive angular error correction

In order to consider the aging effect of sensor hardware, the angular teach-in phase could be integrated into the operational phase, where a dynamic adaption of the coefficients for the rectification functions is executed. Teach-in procedures for 360° rotating systems are simply realizable by use of timers and a comparison of times between the multiple cycles of complete evaluation of the operational area. A more challenging issue is the dynamic compliance of the zero angle reference.

6.2.6 Further investigations

Investigation of the modified approximation algorithm

Based on results from preliminary investigations it was found out that the modified approximation algorithm for angle information extraction incl. stray field suppression exhibits unacceptable noise sensitivity[8] and was therefore excluded from the main investigations directly in the beginning. Either the high sensitivity problem is indeed associated with the division by additional terms included the denominator during the estimation process or the observed effects represent an implementation issue of the function. Unfortunately, the search of the origin of the occurred problem could not be covered in line with this thesis due to set time constraints. This issue shall be definitely investigated in the context of further work.

Approximation of dipole derivative functions

As a part of previous project work, an algorithm was developed, that differs from the direct approximation methods and direct curvature extraction [75, Chap. 5.5]. In this algorithm in the first step the local derivatives are determined numerically and in the second step the coefficients of partial derivative functions of the dipole equation are estimated. The angle is evaluated from the coefficients coupled with z-coordinate. Like the modified approximation approach, this algorithm requires a knowledge about the z-distance from the magnet and the application of sphere magnet, but it does not require the determination of the projection point of the encoder magnet rotation axis.

[8] Results are shown and discussed in Section 5.4 and Appendix G.4.

Due to a wider focus set for this thesis this algorithm was not taken into the investigations. It shall definitely be built into the evaluation software and investigated in the context of further algorithm comparisons.

Features extraction from 2D-DFT

Within the context of this work, an analysis of spatial frequencies was started. Due to time constraints, the analysis was not carried out completely. However, some features could be extracted, e.g. determined frequency pairs for the angle information calculation and, it was found out that the encoder magnet tilt causes a continuous phase shift of perpendicular spatial frequencies. There is a high potential for features extraction from the spatial content of 2D-DFT, especially in the evaluation of phase information. Possibly, compensation of misalignment effects is achievable, perhaps even without including of teach-in phases. The spatial frequencies behavior shall be further investigated.

6.2.7 Extended system proposal

In the context of the investigations in this thesis it was shown that the evaluation of the field curvature obtained from the spatial domain for angle calculation is extremely sensitive against the mechanical misalignment of the magnet encoder. Furthermore, the amplitudes of the signals which represent information about the extracted curvature from the field are decreased in the presence of a misalignment significantly, what was pointed out in Section 3.5, Figure 3.19 and Section 5.7.5 Figures 5.46 and 5.53. Instead of it, large global gradients occurs in the measured field components. Therefore, as a long term goal, it is proposed to construct a hybrid system and always switch to the evaluation of global gradients for the angle calculation in case of a detected misalignment of a large relativeness. The developed misalignment detection method described in Section 3.5.1 and the algorithms for tracking of the AMM points combined with the spatial teach-in procedure shall be used for this purpose. The threshold level for this transition shall be found out within the scope of further work.
Furthermore, it is proposed to:

- correct the angular error always by use of angular teach-in procedures, i.e. by creation of the virtual operational plane, estimation of coefficients of rectification functions and their subsequent continuous application in the operation phase of the sensor.
- use always at least the angular correction algorithm based on homography of 8 DOF.

Moreover, as an ultimate goal it is proposed to implement not only the main functionalities for angle information extraction, but also the estimation and misalignment detection procedures directly on the ASIC, in order to include the main features on one chip: a self calibrating chip system without additional need for relocation of computations on the μC or ECU, to ensure a complete self-sufficiency of the novel sensor array system. The feasibility of a practical implementation of the developed methods shall be examined by the **ISAR** research group in the future.

Bibliography

[1] *AMR sensors*. Institut für Elektrische Messtechnik und Grundlagen der Elektrotechnik, Technische Universität Braunschweig. – URL http://www.emg.tu-bs.de/forschung/mag_sens/amr_e.html. – Zugriffsdatum: 23.12.2018

[2] *Application Note: Noise Basics*. Noisecom. – URL https://www.noisecom.com/~/media/Noisecom/noisebasics.ashx. – Zugriffsdatum: 23.12.2018

[3] *Online article: Basics of Magnetoresistive (MR) Sensors*. TE Connectivity. – URL https://www.te.com/usa-en/industries/sensor-solutions/insights/intro-into-mr-sensor-applications.html. – Zugriffsdatum: 23.12.2018

[4] *Online article: Bestimmung des Stichprobenumfangs (engl: Determination of the sample size)*. Humboldt-Universität zu Berlin. – URL https://mars.wiwi.hu-berlin.de/mediawiki/mmstat3/index.php/Bestimmung_des_Stichprobenumfangs. – Zugriffsdatum: 23.12.2018

[5] *Online article: Colors of noise*. Wikipedia, the free encyclopedia. – URL https://en.wikipedia.org/wiki/Colors_of_noise. – Zugriffsdatum: 23.12.2018

[6] *Online article: HOW TO DETERMINE SAMPLE SIZE, DETERMINING SAMPLE SIZE*. iSixSigma, high-tech B2B media business. – URL https://www.isixsigma.com/tools-templates/sampling-data/how-determine-sample-size-determining-sample-size/. – Zugriffsdatum: 23.12.2018

[7] *Online article: Johnson-Nyquist noise*. Wikipedia, the free encyclopedia. – URL https://en.wikipedia.org/wiki/Johnson%E2%80%93Nyquist_noise. – Zugriffsdatum: 23.12.2018

[8] *Online article: Magnetic Position Sensors*. TE Connectiviy. – URL https://www.te.com/usa-en/products/sensors/position-sensors/magnetic-sensors.html?tab=pgp-story. – Zugriffsdatum: 23.12.2018

[9] *Online article: Noise Generation*. National Instruments. – URL http://zone.ni.com/reference/en-XX/help/370051V-01/cvi/libref/analysisconcepts/noise_generation/. – Zugriffsdatum: 23.12.2018

[10] *Online article: Q (number format)*. Wikipedia, the free encyclopedia. – URL https://en.wikipedia.org/wiki/Q_(number_format). – Zugriffsdatum: 23.12.2018

[11] *Online article: Sinus cosinus*. Wikipedia, the free encyclopedia. – URL https://de.wikipedia.org/wiki/Sinus_und_Kosinus. – Zugriffsdatum: 23.12.2018

[12] *Online article: Six Sigma. Lean Manufacturing and Six Sigma Definitions (Glossary, terms and definitions for Lean and Six Sigma)*. Organization: Business Performance Improvement (BPI). – URL http://leansixsigmadefinition.com/glossary/six-sigma/. – Zugriffsdatum: 23.12.2018

[13] *Online article: Tunnelmagnetoresistive Sensoren für praezise Wege und Winkel (engl: Tunnel magnetoresistive sensors for precise distances and angles)*. Elektronik Praxis. – URL https://www.elektronikpraxis.vogel.de/index.cfm?pid=7525&pk=625317&fk=1251387&type=article. – Zugriffsdatum: 23.12.2018

[14] *Online version of diploma thesis: Interaktion zwischen Mensch und Maschine (engl: Interaction between human and machine)*. Technische Hochschule Nürnberg Georg Simon Ohm (engl: Nuremberg Institute of Technology Georg Simon Ohm). – URL https://me.efi.th-nuernberg.de/interaktion/index.php5/Bideo. – Zugriffsdatum: 23.12.2018

[15] *Technical Note: Analog to Digital Conversion*. Measurement Computing (mccdaq.com). – URL https://www.mccdaq.com/PDFs/specs/Analog-to-Digital.pdf. – Zugriffsdatum: 23.12.2018

[16] *Technical Note: TMR Angle sensors*. TDK Corporation. – URL https://product.tdk.com/info/en/products/sensor/angle/tmr_angle/technote/tpo/index.html. – Zugriffsdatum: 23.12.2018

[17] *519-1992 - IEEE Recommended Practices and Requirements for Harmonic Control in Electrical Power Systems, Chapter 8*. Institute of Electrical and Electronics Engineers, 1993

[18] *DIN EN 61000-2-4 VDE 0839-2-4:2003-05, Electromagnetic compatibility (EMC), Environment - Compatibility levels in industrial plants for low-frequency conducted disturbances ((IEC 61000-2-4:2002); German version EN 61000-2-4:2002)*. DIN Deutsches Institut für Normung e. V. (engl: German Institute for Standardization), 2003

[19] *International Organization for Standardization, Industrial Standard ISO 26262-1:2011: Road Vehicles - Functional Safety*. International Organization for Standardization, 2011

[20] *AMR sensor KMZ60 product datasheet: Angle sensor with integrated amplifier (Rev. 2 -7, February 2014)*. NXP Semiconductors Germany GmbH, 2014

[21] *KMI17/4 Rotational speed sensor product datasheet: Angle sensor with integrated amplifier (Rev.1 -14, Septmber 2014)*. NXP Semiconductors Germany GmbH, 2014

[22] *International Organization for Standardization, Industrial Standard ISO 11452-8:2015: Road vehicles - Component test methods for electrical disturbances from narrowband radiated electromagnetic energy - Part 8: Immunity to magnetic field*. International Organization for Standardization, 2015

[23] *Mobile Product Catalog: Sensor Solutions for Automotive, Industrial and Consumer Applications (Order Number: B142-I0126-V1-7600-EU-EC-P, Date: 05/2015*. Infineon Technologies AG, 2015. – URL https://www.infineon.com/dgdl/Infineon-Sensor+Solutions+for+Automotive,+Industrial+and+Consumer+Applications-ABR-v01_00-EN.pdf?fileId=5546d46250cc1fdf01516335e36c3e01. – Zugriffsdatum: 23.12.2018

[24] *Online article: MR Sensor Technology, Introduction to TMR Magnetic Sensors*. Multi-Dimension Technology Co.,Ltd., 2015. – URL http://www.dowaytech.com/en/1776.html. – Zugriffsdatum: 23.12.2018

[25] *Project description: Signalverarbeitung für Integrated Sensor-Arrays basierend auf dem Tunnel-Magnetoresistiven Effekt für den Einsatz in der Automobilelektronik (engl: Signal processing for integrated sensor arrays based on the tunnel magnetoresistive effect for application in automotive electronics). Acronym: ISAR*. Bundesministerium für Bildung und Forschung (engl: Federal Ministry of Education and Research), 2016

[26] *MLX90380 - Triaxis Resolver data sheet (REVISION 9 - 13 MARCH 2018, 3901090380)*. Melexis NV Corporation, 2018

[27] *Online article: Melexis announces next-generation Triaxis position sensor IC)*. Melexis NV Corporation, 2018. – URL https://www.melexis.com/en/news/2018/26mar2018-next-generation-triaxis-position-sensor-ic. – Zugriffsdatum: 23.12.2018

[28] *Patent: ROTATORY ENCODER WITH SHIELDED MAGNET (United States Patent, No.: US 9,892,836 B2. Feb. 13, 2018)*. Infineon Technologies AG (Inventor: Udo Ausserlechner), 2018

[29] ALEX HUBERT, Rudolf S.: *Book: Magnetic Domains, the Analysis of Magnetic Microstructures*. Springer, 1998. – ISBN 978-3-540-85054-0

[30] ARTMANN, Uwe: *Online artcle: So entstehen Verzeichnung und Vignettierung (engl: This is how distortion and vignetting occur)*. PC MAGAZIN, 2012. – URL https://www.pc-magazin.de/ratgeber/verzeichnung-und-vignettierung-krumm-schief-1250376.html. – Zugriffsdatum: 23.12.2018

[31] BRUNTON, Proctor: *Singular Value Decomposition (SVD) and Principal Components Analysis (PCA)*. URL http://faculty.washington.edu/sbrunton/me565/pdf/CHAPTER1.pdf. – Zugriffsdatum: 23.12.2018, 2015

[32] BY DEREK REDMAYNE, Alison S.: *Online article: Understanding the effect of clock jitter on high-speed ADCs*. EE Times (Linear Technology Corp.). – URL https://www.eetimes.com/document.asp?doc_id=1272469. – Zugriffsdatum: 23.12.2018

[33] C. H. SMITH, T. Dogaru S. T. S.: *Scientific publication: EDDY-CURRENT TESTING WITH GMR MAGNETIC SENSOR ARRAYS (published in Review of Progress in Quantitative Nondestructive Evaluation, Vol. 2323)*. American Institute of Physics, Melville, NY, 2003, 2003

[34] CARL H. SMITH, Robert W. S.: *CHIP-SIZE MAGNETIC SENSOR ARRAYS*. NVE Corporation, Eden Prairie, MN, 2002

[35] CHRISTIANO GAVA, Gabriele B.: *Lecture materials: 2D projective transformations (homographies)*. The University of Kaiserslautern, 2011. – URL https://ags.cs.uni-kl.de/fileadmin/inf_ags/3dcv-ws11-12/3DCV_WS11-12_lec04.pdf. – Zugriffsdatum: 23.12.2018

[36] CNRS, Coriolan TIUSAN U.: *Tunneling Magnetoresistance (TMR) in Magnetic Tunnel Junctions (MTJ)*. UTCN - CNRS. – URL https://slidex.tips/download/tunneling-magnetorezistance-tmr-in-magnetic-tunnel-junctions-mtj. – Zugriffsdatum: 23.12.2018

[37] COLLINS, Robert: *Lecture materials Computer Vision I: Planar Homographies*. Penn State, College of Engineering, Electrical Engineering and Computer Science Faculty. – URL http://www.cse.psu.edu/~rtc12/CSE486/lecture16.pdf. – Zugriffsdatum: 23.12.2018

[38] DACHS, Susanne: *Technical magazine: Elektronik, Fachmedium für industrielle Anwender und Entwickler. Chapter: Praezise Winkelsensoren für Industrieanwendungen, Mehr messen mit Magneten (engl: Electronics, specialist for industrial users and developers. Chapter: Precise angle sensors for industrial applications, more measuring with magnets). Number 24, 22. November 2018*. VDE/VDI-Gesellschaft Mikroelektronik, Mikrosystem- und Feintechnik, 2018

[39] DAHLKEMPER, Jörg: *Lecture materials: Angewandte Industrielle Bildverarbeitung (AIBV) (engl: Applied industrial image processing)*. Hamburg University of Applied Sciences (HAW Hamburg), 2018

[40] DIETER SUESS, Claas Abert Thomas Schrefl Hubert Brückl et al.: *Topologically Protected Vortex Structures to Realize Low-Noise Magnetic Sensors*. University of Vienna, Christian Doppler Laboratory, Faculcy of Physics, Physics of functional materials, Infineon Technologies AG, 2017

[41] DREW A. HALL, Kofi A. A. Makinwa ShanX.Wang Boris M.: *Scientific publication: A 256 Pixel Magnetoresistive Biosensor Microarray in 0.18 micrometer CMOS*. IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 48, NO. 5, MAY 2013, 2013

[42] ENGELN-MÜ, Gisela: *Book: Numerik-Algorithmen, 10. Auflage (engl. Numeric algorithms)*. Springer

[43] FLUSSER, Jan: *Scientific publication: On the independence of rotation moment invariants*. Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic (Permagon Press, Pattern Recognition 33 (2000) 1405 1410)

[44] GEORGI TODOROV NIKOLOV, Stefan Valentinov V. ; NIKOLOVA, Boyanka M.: *Magnetic Fields Measurement with AMR Sensors*. ANNUAL JOURNAL OF ELECTRONICS, 2009 ISSN 1313-1842

[45] GREG HAGER, S. Seitz K G.: *2D Projective Geometry*. Johns Hopkins University. – URL `http://www.cs.jhu.edu/~hager/teaching/cs461/Notes/Lect5-Homography.pdf`. – Zugriffsdatum: 23.12.2018

[46] GRONAU, Norbert: *Online article: Prozessverbesserung mittels Six Sigma, Anwendung am Beispiel einer Grossserienfertigung (engl: Process improvement using Six Sigma, using the example of a large-scale production)*. GITO mbH Verlag. – URL `https://productivity-management.de/node/427`. – Zugriffsdatum: 23.12.2018

[47] GUYOL, Robert: *AN-1314 APPLICATION NOTE: AMR Angle Sensors*. Analog devices, 2014

[48] HARTLEY, Richard I.: *Scientific publication: In Defense of the Eight-Point Algorithm*. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 1997

[49] JOHN MADDOCK, Hubert Holin Xiaogang Zhang Bruno Lalande Johan Rade Gautam Sewani Thijs van den Berg Benjamin S.: *Online article: Normal (Gaussian) Distribution*. Boost C++ libraries. – URL `https://www.boost.org/doc/libs/1_53_0/libs/math/doc/sf_and_dist/html/math_toolkit/dist/dist_ref/dists/normal_dist.html`. – Zugriffsdatum: 23.12.2018

[50] JOHN R. BARRY, David G. M.: *Book: Digital Communication*. Springer Verlag. – ISBN 978-0792375487

[51] JÜNEMANN, Klaus: *Script : Magnetostatik, Chapter 1 DAS DIPOLFELD (engl: Magnetostatics, Chapter 1 THE DIPOLE FIELD)*. Hamburg University of Applied Sciences (HAW Hamburg)

[52] JÜNEMANN, Klaus: *ISAR Presentation: Winkelberechnung mit der Methode der kleinsten Quadrate (engl: Angle calculation using least squares method)*. Hamburg University of Applied Sciences (HAW Hamburg), 2017

[53] KARL-RAGMAR RIEMSCHNEIDER, Thorben S.: *ISAR Presentation: SPATIAL FREQUENCY FILTERING AND STRAY FIELD COMPENSATION USING 2D-DFT ALGORITHM*. Hamburg University of Applied Sciences (HAW Hamburg), 2017

[54] KARPFINGER, Christian: *Book: Höhere Mathematik in Rezepten (engl. Higher mathematics in Recipes)*. Springer Spektrum, 2014. – ISBN 978-3-662-43810-7

[55] KESTER, Walk: *Tutorial materials: Understand SINAD, ENOB, SNR, THD, THD + N, and SFDR so You Don't Get Lost in the Noise Floor (MT-003 TUTORIAL)*. Analog Devices. – URL https://www.analog.com/media/en/training-seminars/tutorials/MT-003.pdf. – Zugriffsdatum: 23.12.2018

[56] KESTER, Walk: *Taking the Mystery out of the Infamous Formula, SNR = 6.02N + 1.76dB, and Why You Should Care*. Analog Devices, 2008. – URL http://www.analog.com/media/en/training-seminars/tutorials/MT-001.pdf. – Zugriffsdatum: 23.12.2018

[57] KLAUS DIETMAYER, Marcus W.: *Application Note AN00023: Contactless Angle Measurement using KMZ41 and UZZ9000 (24 March 2000)*. Philips Semiconductors

[58] KÖLZER, Hans P.: *Script: Elektronik 3 (E4 / EL3) (engl: electronics 3*. Hamburg University of Applied Sciences (HAW Hamburg), 2013

[59] KÖLZER, Hans P.: *Lecture materials: Bildverarbeitung und Mustererkennung (engl: Image processing and pattern recognition)*. Hamburg University of Applied Sciences (HAW Hamburg), 2018

[60] KONDUL, Ada: *Master thesis: Signalverarbeitung für ein magnetisches Sensor-Array als digitaler Chipentwurf (engl: Signal processing for a magnetic sensor array as a digital chip design)*. Hamburg University of Applied Sciences (HAW Hamburg), 2018

[61] KREY, Martin: *Systemarchitektur und Signalverarbeitung für die Diagnose von magnetischen ABS-Sensoren (engl: System Architecture and Signal Processing for the Diagnosis of Magnetic Speed Sensors)*. Helmut Schmidt Universität, Universität der Bundeswehr Hamburg, 2015

[62] KRIEGMAN, David: *Lecture materials Computer Vision I: Homography Estimation*. UC San Diego, Jacobs School of Engineering, 2007. – URL https://cseweb.ucsd.edu/classes/wi07/cse252a/homography_estimation/homography_estimation.pdf. – Zugriffsdatum: 23.12.2018

[63] LARRY ZITNICK (RESEARCHER) LARRYZ@MICROSOFT.COM, http://larryzitnick.org/): *Computer vision slides: Image Geometric Transformations, CSE P 576*. University of Washington / Facebook AI Research in Menlo Park. – URL https://courses.cs.washington.edu/courses/csep576/11sp/pdf/Transformations.pdf. – Zugriffsdatum: 23.12.2018

[64] LATTMANN, Thomas: *Bachelor thesis: Chipimplementation einer zweidimensionalen Fouriertransformation für die Auswertung eines Sensor-Arrays (engl: Chip implementation*

of a two-dimensional Fourier transformation for the evaluation of a sensor array). Hamburg University of Applied Sciences (HAW Hamburg), 2017

[65] LEE, Do Q.: *Scientific publication: NUMERICALLY EFFICIENT METHODS FOR SOLVING LEAST SQUARES PROBLEMS*. The University of Chicago, Department of Mathematics, 2012

[66] LUREN YANG, Fritz A.: *Scientific publication: FAST AND EXACT COMPUTATION OF CARTESIAN GEOMETRIC MOMENTS USING DISCRETE GREEN'S THEOREM*. Image Processing Laboratory, Department of Informatics, University of Oslo (Permagon Press), 1995

[67] MAASS, Jochen: *Lecture script: Nichtlineare Regelungstechnik (engl: Nonlinear control)*. Hamburg University of Applied Sciences (HAW Hamburg), 2015

[68] MARAUSKA, Stephan: *ISAR project partner*

[69] MARAUSKA, Stephan: *Hall-Mikromagnetometrie strominduzierter Domaenenwandbewegung in Nanodraehten*. 2010

[70] MATZ, Hartmut: *ISAR project partner*

[71] MOLER, Cleve: *Professor SVD*. MathWorks, 2006. – URL `https://de.mathworks.com/company/newsletters/articles/professor-svd.html`. – Zugriffsdatum: 23.12.2018

[72] MORSE, Bryan: *Lecture materials. Title: Low-Pass Filtering, CS 450: Introduction to Digital Signal and Image Processing*. Brigham Young University Computer Science. – URL `https://canvas.instructure.com/courses/136140/files/1304766/download?verifier=cTwwL1NOg2GxHApU0uXAarsChyxLoEhCCd0NlSoX&wrap=1`. – Zugriffsdatum: 23.12.2018

[73] OYRER, Heinz: *Slides: Magnetic Position Sensors (published by Senior Manager Global Marketing)*. ams AG, 2015. – URL `https://www.slideshare.net/HeinzOyrer/ams-magnetic-positionsensorsslidedeck201502251`. – Zugriffsdatum: 23.12.2018

[74] P P FREITAS, S C. ; CARDOSO, F: *Magnetoresistive sensors*. JOURNAL OF PHYSICS: CONDENSED MATTER, 2007

[75] PAPE, Hauke: *Bachelorthesis: Simulation und Auswertung von Permanentmagneten für magnetoresistive Sensor-Arrays (engl: Simulation and evaluation of permanent magnets for magnetoresistive sensor arrays*. Hamburg University of Applied Sciences (HAW Hamburg), 2017

[76] PAVEL RIPKA, Alois T.: *Modern Sensors Handbook*. ISTE Ltd, 2007. – ISBN 978-1-905209-66-8

[77] PURGATHOFER, Werner: *Geometrische Transformationen (engl: geometric transformations*. URL https://www.cg.tuwien.ac.at/courses/CG1/textblaetter/02%20Geometrische%20Transformationen.pdf. – Zugriffsdatum: 23.12.2018

[78] RAFAEL C. GONZALEZ, Richard E. W.: *Book: Digital Image Processing. 4th Edition, Global Edition*. Pearson, 2018. – ISBN 10: 1-292-22304-9 / 13: 978-1-292-22304-9

[79] RAUSCHER-SCHEIBE, Annabella: *Discrete Fourier-transform table from lecture materials: Signale- und Systeme 2 (engl: Signals and systems 2)*. Hamburg University of Applied Sciences (HAW Hamburg), 2013

[80] REICHARDT, Jürgen: *Book: Lehrbuch Digitaltechnik, Eine Einführung mit VHDL (engl: Textbook Digital Technology, An Introduction to VHDL)*. Oldenbourg, 2013. – ISBN 978-3-486-72765-4

[81] RIEMSCHNEIDER, Karl-Ragmar: Hamburg University of Applied Sciences (HAW Hamburg)

[82] S. MATHE, A.D. J.: *Tutorial: Planar homographies)*. University of Toronto, Computer Science Faculty, 2004. – URL http://www.cs.toronto.edu/~jepson/csc2503/tutorial2.pdf. – Zugriffsdatum: 23.12.2018

[83] SAUVAGERD, Ulrich: *Lecture materials: Digital Signal Processing on signal processors and Discrete Fourier Transform*. Hamburg University of Applied Sciences (HAW Hamburg), 2016

[84] SCHMIDT-WALTER, Kories: *Book: Taschenbuch der Elektrotechnik (engl: Handbook about Electrical Engineering)*. Verlag Harri Deutsch, 2006. – ISBN 978-3-8171-1793-2

[85] SLATTER, Rolf: *Journal article: Tunnelmagnetoresistive Sensoren für die Antriebstechnik (Sensitec)*. ELEKTRONIKPRAXIS, 2017. – URL https://www.sensitec.com/fileadmin/sensitec/About_Sensitec/Public_Relations/Expert_Publications/SENSITEC_PN-2017-07_TMR-Technologie_von_Sensitec_DE.pdf. – Zugriffsdatum: 23.12.2018

[86] SLAWOMIR TUMANSKI, Slawomir B.: *MAGNETIC SENSOR ARRAY FOR INVESTIGATIONS OF MAGNETIC FIELD DISTRIBUTION*. Journal of ELECTRICAL ENGINEERING, VOL 57. NO 8/S, 2006

[87] STORK, Thomas: *Automotive FAE Event 2010 presentation: Speed Sensors (Smart Sensor Systems Group)*. NXP Semiconductors Germany GmbH, 2010

[88] SZELISKI, Richard: *Book: Computer Vision: Algorithms and Applications*. Springer, 2010. – URL http://szeliski.org/Book/

[89] THOMAS DUNKER, Sebastian L.: *Scientific publication: (Forum Bildverarbeitung) Sensormodell und Kalibrierung für einen IR-Streifenlichtsensor (engl: Sensor Model and Calibration for an IR Strip Light Sensor (Forum Image Processing)*. Fraunhofer Institut für Fabrikbetrieb und -automation, IFF (engl: Fraunhofer Institute for Factory Operation and Automation (IFF)), 2012

[90] THORBEN SCHÜTHE, Karl-Ragmar R.: *ISAR Presentation: KINEMATICS, -FORWARD AND INVERSE-*. Hamburg University of Applied Sciences (HAW Hamburg), 2017

[91] TILLE, Thomas: *Book: Automobil-Sensorik. Ausgewählte Sensorprinzipien und deren automobile Anwendung. (engl: Automotive sensors. Selected sensor principles and their automotive application)*. Springer Verlag, 2016. – ISBN 978-3-662-48943-7

[92] TUMANKI, Slawomir: *Modern magnetic field sensors - a review (R. 89 NR 10/2013)*. Warsaw University of Technology, PRZEGLAD ELEKTROTECHNICZNY, 2013 ISSN 0033-2097

[93] TUMANSKI, Slawomir: *Book: Thin Film Magnetoresistive Sensors. Chapter: GMR Sensors*. – ISBN 9780750307024

[94] VILLIERS, Ronelle G. Jason P. de: *Scientific publication: Centi-pixel accurate real-time inverse distortion correction*. Council for Scientific and Industrial Research, Meiring Naude St, Pretoria, South Africa (Proc. of SPIE Vol. 7266 726611-1), 2008

[95] WEISS, Pierre: *La variation du ferromagnetisme du temperature*. Comptes Rendus, 1906

[96] WINFRIED GEHRKE, et. a.: *Book: Digitaltechnik: Grundlagen, VHDL, FPGAs, Mikrocontroller (engl: Digital technology: basics, VHDL, FPGAs, microcontrollers)*. Springer-Verlag, 2013. – ISBN 978-3-662-49730-2

[97] ZHANG, Zhengyou: *Scientific publication: Flexible Camera Calibration By Viewing a Plane From Unknown Orientations*. Microsoft Research, One Microsoft Way, Redmond, WA 98052-6399, USA

List of Tables

2.1 Comparison of magnetic sensing technology parameters. 29
2.2 Technical application conditions and tolerances. 31
2.3 Mechanical sensor die tolerances. 31
2.4 Stray field specification and requirements for system accuracy. 32

3.1 Rating of the algorithms for angle information extraction. 61

4.1 Dimensions and amount of dipoles of simulated encoder magnets. 105
4.2 Limitations of simulation parameters and their variation step sizes. 110
4.3 Limitations of simulation parameters and their variation step sizes: SNR. 110
4.4 Limitations of simulation parameters and their variation step sizes: ADC resolution and arithmetic accuracy. 111

5.1 Field strengths, specified field gradients and their percentage reduction in dependence of the z-distance. 136
5.2 Optimized exponents n for the disc magnet and remained MEFC in %. 151
5.3 MEFC in % of field components of the disc magnet without modification of approximation. 151
5.4 MEFC in % of field components of the sphere magnet without modification of approximation. 151
5.5 Proof of concept results. Sensitivity against quantization noise. 163
5.6 Limits for an angular accuracy $\varphi_{err} \leq 1^\circ$ for the algorithms in dependence of the array size. Results of approximation algorithm. 167
5.7 Applied limits for gain and offset deviations for the proof of concept test. 167
5.8 Maximum and mean angular errors depending on the register width for the fractional part of applied s1Qn fixed point format. 173
5.9 Maximum angular errors depending on the register width for the fractional part of appplied s1Qn fixed point format. 175
5.10 Corners of misalignment parameters space within the SOA 176
5.11 Acceptable x- and y-translation misalignment areas depending on z-distance and x-tilt associated with the angular accuracy requirement $\varphi_{err} \leq 1^\circ$. *Simulation setup: 15 x 15 array, 14 bit ADC resolution.* 179
5.12 Randomly chosen misalignment cases for the test of array size and misalignment effect dependency. 179

5.13 Array size and misalignment effect dependency test results. Change of mean angular error for 3 applied misalignment cases. 180
5.14 Applied mechanical misalignment parameters of encoder magnet for the test of tracking algorithms. 181
5.15 Applied mechanical misalignment parameter of sphere encoder magnet for extended algorithms test. 187
5.16 Direct comparison of mean and maximum angular errors of non extended and AMM point tracking extended algorithms. 189
5.17 Relative reduction of maximal angular error of AMM point tracking extended algorithms in dependence of sensor array size. 189
5.18 Limits for an angular accuracy $\varphi_{err} \leq 1^{\circ}$ for AMM point tracking extended algorithms. Results of approximation algorithm. 190
5.19 Assumed values for the magnitudes and phases of superimposed harmonics for the harmonics cancelation test. 193
5.20 Angular error correction algorithms test: Harmonics cancelation. Comparision of THDs. 198
5.21 Applied mechanical misalignment conditions of encoder magnets for the angular error correction test: extreme misalignment case. 199
5.22 Angular error correction algorithms test: Misalignment effect compensation, extreme case. 204
5.23 Angular error correction algorithms test: Misalignment effect compensation, extreme case. 205
5.24 Applied mechanical misalignment conditions of encoder magnets for the angular error correction test: worst SOA misalignment case. 207
5.25 Angular error correction algorithms test: Misalignment effect compensation, worst SOA case. 210

G.1 Proof of concept results. Subblock 1.0 (1). 297
G.2 Proof of concept results. Subblock 1.0 (2). 298
G.3 Proof of concept results. Subblock 1.0 (3). 299
G.4 Proof of concept results. Subblock 1.0 (4). 300
G.5 Proof of concept results. Subblock 1.2 (1). 312
G.6 Proof of concept results. Subblock 1.2 (2). 313

List of Figures

2.1 Automotive applications for magnetic field sensors. 17
2.2 Rotation measurements by magnetic positions sensors. 18
2.3 Simplified circuit diagram for a double Wheatstone bridge AMR sensor 19
2.4 Illustration of AMR effect. 22
2.5 A typical resistance curve of an AMR permalloy stripe. 23
2.6 GMR trilayer and spin-valve structure and the corresponding hysteresis curves. 25
2.7 Comparison of GMR and TMR stack structures and output voltages. 27
2.8 Principle of magnetic field sensing by a Hall element. 28
2.9 Illustration of the stray field suppression capability of a sensor array. 33

3.1 Functional split of the total processing approach. 37
3.2 Developed processing flow. 38
3.3 Schematic representation of physical signal flows from the sensor elements. . . 39
3.4 Calibration stages of the sensor array system. 40
3.5 Progression of $\mathbf{B_x}$ components measured by the sensor array in the xy-plane. . 42
3.6 Progression of $\mathbf{B_y}$ components measured by the sensor array in the xy-plane. . 42
3.7 Spatial frequency components of 2D-DFT applied separated on $\mathbf{B_x}$ and $\mathbf{B_y}$ data. 45
3.8 Progression of analyzed spatial frequency components over 360°. 46
3.9 Signal flow diagram of 2D-DFT based algorithm. 47
3.10 Comparison of simulated and approximated field components. 50
3.11 Progression of analyzed coefficients of chosen basis functions. 51
3.12 Signal flow diagram of approximation based algorithm. 51
3.13 LoG filter kernel and its frequency response. 57
3.14 Direct curvature extraction possibility by application of LoG filter. 57
3.15 Signal flow diagram of LoG convolution based algorithm. 58
3.16 Illustration of median filter algorithm. 62
3.17 Median rank filter compared to mean filter. 63
3.18 Butterworth low pass-filter. 64
3.19 Misalignment handling pre-analysis. 66
3.20 Realignment of field components. 67
3.21 Schematically representation of tracking algorithm working principle. 71
3.22 Results of preliminary test in frame of tracking algorithms development. 73
3.23 Signal flow diagram of Butterworth low pass filter based tracking algorithm. . . 74
3.24 Signal flow diagram of Laplacian of Gaussian filter based tracking algorithm. . . 76

3.25 Signal flow diagram of derivative of Gaussians filter based tracking algorithm. . 77
3.26 Signal flow diagram of approximation based tracking algorithm. 79
3.27 Signal flow diagram of simplified appr. and analytical based tracking algorithm. 81
3.28 Sine and cosine components of a unit circle projected to the angular axis. . . . 82
3.29 Line preserving projection of information of two arbitrary planes. 85
3.30 Chosen coordinates sets p and p' and obtained planar image projection. 85
3.31 The set of possible 2D planar transformations. 86
3.32 Teach-in procedure of the angular error correction algorithm. 91
3.33 Homography camera experiment. 93
3.34 Two geometric distortion types and their correction 94
3.35 Functionality of geometric distortion cancelation algorithm 96
3.36 Split of the operational area required by nonlin. angular error correction algorithm. 97
3.37 Schematically representation of stray field component interaction. 101
3.38 Signal flow chart of the whole algorithmic chain. 103

4.1 Created encoder magnets in the simulation and the sensor xy-plane. 106
4.2 Coordinate system, units and initial values in the simulation. 107
4.3 Simulated vector field in the created sensor array plane. 108
4.4 Illustration of GUI data flow. 113
4.5 Proof of ADC function functionality *sample_components_signed*. 118
4.6 Gaussian probability density functions. 120
4.7 Effect of a white Gaussian noise. 120
4.8 Comparison of quant. and white Gaussian noise impacts on angular accuracy (1). 123
4.9 Comparison of quant. and white Gaussian noise impacts on angular accuracy (2). 124

5.1 Strength of $\mathbf{H_x}$ components of simulated encoder magnets at z-distance of 2 mm. 134
5.2 Cross sections in the simulated xy-plane in x-direction at y-coordinate = 0 mm. 135
5.3 Delineation of the gradient calculation from $\mathbf{H_x}$ components at defined $\mathbf{H_y}$ cross section. 136
5.4 Deformation of field components caused by Butterworth low pass filtering. . . . 139
5.5 Mean error of field components in % between ideal and filtered components caused by application of Butterworth low pass filter. 140
5.6 Chosen Butterworth low pass filter kernel for preprocessing in spatial frequency domain (1). 141
5.7 Chosen Butterworth low pass filter kernel for preprocessing in spatial frequency domain (2). 142
5.8 Comparison of filtering effects gained from filtering of a not padded and zero padded signals in spatial frequency domain. 143
5.9 Deformation of field components caused by nonlinear filtering with median filter. 145
5.10 Mean error of field components in % interfered by white Gaussian noise and components filtered by median filter in dependence of SNR. 146
5.11 LoG filter kernels of 15 x 15 size with different σ parameters. 147

5.12 LoG filter kernels of 15 x 15 size with different σ parameters transformed into spatial frequency domain. 148
5.13 Optimization of LoG filter bandwidth. 149
5.14 Optimization of exponent n for the disc magnet for z-distance 2 mm to 5 mm and array size of 15 x 15 elements. 150
5.15 Preliminary investigation: misalignment test 1. 153
5.16 Preliminary investigation: quantization test 1. 154
5.17 Preliminary investigation: quantization test 2. 154
5.18 Investigation plan: created total test plan for execution of investigations of developed algorithms. 158
5.19 Investigation plan: Content of block 1. 159
5.20 Investigation plan: Content of block 2. 161
5.21 Investigation plan: Content of block 4. 162
5.22 Proof of concept investigation: Effects of quantization noise. 164
5.23 Investigated impact of noise on the angular accuracy of algorithms. 166
5.24 Histogram of distribution of applied gains and offsets within set limits. 168
5.25 Impact of gain and offset deviations of fictive sensor electronics elements on the measured field components. 168
5.26 Effects of interfering of sensor signals by white Gaussian noise. 169
5.27 Impact of insufficient bit width for the arithmetic operations (resulting from initial s1Q10 format). 172
5.28 Increase of angular accuracy by enlarging of the bit width for the fractional part of airthmetical operations registers (test 1). 173
5.29 Increase of angular accuracy by enlarging of the bit width for the fractional part of airthmetical operations registers (test 2). 174
5.30 Misalignment test results. Results of all considered algorithms. 177
5.31 Misalignment test results. Results of approximation approach. 178
5.32 Expected localized AMM points within the sensor array area for the test of tracking algorithms. 182
5.33 Initially localized AMM points by considered tracking algorithms. 183
5.34 Mean distance deviation for all localized AMM points for the sphere magnet data. 184
5.35 Mean distance deviation for localized mean AMM point for the sphere magnet data. 185
5.36 Localized AMM points by considered tracking algorithms at 35 dB SNR. 186
5.37 Comparison of maximum and mean angular errors of non extended and AMM point tracking extended algorithms. 188
5.38 Investigated impact of noise on the angular accuracy of AMM point tracking extended algorithms. 191
5.39 FFT analysis of the signals of the AMR sensor for harmonics cancelation test. . 193
5.40 Investigation of linear angular error correction algorithm based on homography approach: harmonics cancelation. 194

5.41 Investigation of nonlinear angular error correction algorithm based on geometric distortion compensation: harmonics cancelation. 195
5.42 Investigation of linear angular error correction algorithm based on homography approach: harmonics cancelation (2). 196
5.43 Investigation of nonlinear angular error correction algorithm based on geom. dist. compensation: harmonics cancelation (2). 196
5.44 Investigation of linear angular error correction algorithm based on homography approach: harmonics cancelation (3). 197
5.45 Investigation of nonlinear angular error correction algorithm based on geometric distortion compensation: harmonics cancelation (3). 197
5.46 Investigation of angular error correction algorithm: misalign. compensation, extreme case. 199
5.47 Investigation of angular error correction algorithm: misalign. compensation, extreme case (2). 200
5.48 Investigation of linear angular error correction algorithm based on homography approach: misalignment compensation, extreme case. 201
5.49 Investigation of linear angular error correction algorithm based on homography approach: misalignment compensation, extreme case (1). 202
5.50 Investigation of nonlinear angular error correction algorithm based on geometric distortion cancelation: misalignment compensation, extreme case. 202
5.51 Investigation of nonlinear angular error correction algorithm based on geom. dist. cancelation: misalignment compensation, extreme case (2). 203
5.52 Investigation of angular error correction algorithms: misalignment effect compensation, extreme case. 206
5.53 Investigation of angular error correction algorithm: misalignment compensation, worst SOA case. 207
5.54 Investigation of angular error correction algorithm: misalignment compensation, worst SOA case (2). 208
5.55 Investigation of angular error correction algorithms: misalignment compensation, worst SOA case (3). 208
5.56 Investigation of linear angular error correction algorithm based on homography: misalignment compensation, worst SOA case. 209
5.57 Investigation of nonlinear angular error correction algorithm based on geometric distortion compensation: misalignment compensation, worst SOA case. 209

C.1 Comparison of field components measured by a not saturated (linear) and a saturated sensor array ystem. 250
C.2 Application of algorithms on saturated sensor array system. 251
C.3 Application of developed template matching algorithm on saturated sensor array system. 251
C.4 Correlation results of performed template matching algorithm test (3D view). . . 252

C.5 Correlation results of performed template matching algorithm test (2D view). . . 253
C.6 Application of Template Matching algorithm on saturated sensor array system in case of stray field interference. 254
C.7 2D-DFT based algorithm for angle information extraction for complex input signals: spatial frequency components. 255
C.8 2D-DFT based algorithm for angle information extraction for complex input signals: progression of spatial frequency components. 255
C.9 Signal flow diagram of 2D-DFT based algorithm for complex input signals. . . . 256

D.1 Created Graphical user interface for visualization and evaluation purposes. . . 257
D.2 GUI control and evaluation panels. 258
D.3 UML activity diagram of created evaluation software, part 1. 277
D.4 UML activity diagram of created evaluation software, part 2. 278
D.5 Ideal n bit ADC transfer function and its quantization noise. 280
D.6 Six sigma process limits in the population standard deviation. 282
D.7 Two's complement number circles. 283
D.8 1Q10-format data register structure. 283

E.1 Comparison of filtering effects gained from filtering of a not padded and zero padded signals in spatial frequency domain. 284
E.2 Mean error of disc magnet field components in % interfered by white Gaussian noise and components filtered by median filter in dependence of SNR. 285

F.1 Optimization of exponent n for the sphere magnet 286
F.2 Optimization of exponent n for the disc magnet 287
F.3 Optimized LoG filter coefficients for 3x3 kernel size 288
F.4 Optimized LoG filter coefficients for 5x5 kernel size 288
F.5 Optimized LoG filter coefficients for 7x7 kernel size 288
F.6 Optimized LoG filter coefficients for 9x9 kernel size 289
F.7 Optimized LoG filter coefficients for 11x11 kernel size 289
F.8 Optimized LoG filter coefficients for 13x13 kernel size 290
F.9 Optimized LoG filter coefficients for 15x15 kernel size 290

G.1 Application of algorithms and evaluation of their results in GUI: stray field robustness test. 291
G.2 Application of algorithms and evaluation of their results in GUI: 2D-DFT extension. 292
G.3 Application of algorithms and evaluation of their results in GUI: misalignment sensitivity test. 293
G.4 Application of algorithms and evaluation of their results in GUI: quantization test 1. 294
G.5 Application of algorithms and evaluation of their results in GUI: quantization test 2. 295
G.6 Application of algorithms and evaluation of their results in evaluation GUI: misalignment compensation. 296

G.7 1° border for mean and maximum angular errors of extended 2D-DFT algorithm for angle information extraction. Encoder magnet: sphere. 5 mm z-distance. . . 301
G.8 1° border for mean and maximum angular errors of approximation based algorithm for angle information extraction. Encoder magnet: sphere. 5 mm z-distance. 302
G.9 1° border for mean and maximum angular errors of single LoG algorithm for angle information extraction. Encoder magnet: sphere. 5 mm z-distance. . . . 303
G.10 1° border for mean and maximum angular errors of LoG convolution for angle information extraction. Encoder magnet: sphere. 5 mm z-distance. 304
G.11 1° border for mean and maximum angular errors of extended 2D-DFT algorithm for angle information extraction. Encoder magnet: disc. 5 mm z-distance. . . . 305
G.12 1° border for mean and maximum angular errors of approximation based algorithm for angle information extraction. Encoder magnet: disc. 5 mm z-distance. 306
G.13 1° border for mean and maximum angular errors of single LoG algorithm for angle information extraction. Encoder magnet: disc. 5 mm z-distance. 307
G.14 1° border for mean and maximum angular errors of LoG convolution for angle information extraction. Encoder magnet: disc. 5 mm z-distance. 308
G.15 Bar plot for mean and maximum angular errors of extended 2D-DFT algorithm for angle information extraction. Encoder magnet: disc. 5 mm z-distance. . . . 309
G.16 Bar plot for mean and maximum angular errors of single LoG algorithm for angle information extraction. Encoder magnet: disc. 5 mm z-distance. 310
G.17 Bar plot for mean and maximum angular errors of approximation based algorithm for angle information extraction. Encoder magnet: disc. 5 mm z-distance. . . . 311
G.18 Maximum angular errors of algorithms at z-distance = 2 mm, x-tilt = 0°. 314
G.19 Maximum angular errors of algorithms at z-distance = 2 mm, x-tilt = 5°. 315
G.20 Maximum angular errors of algorithms at z-distance = 5 mm, x-tilt = 0°. 316
G.21 Maximum angular errors of algorithms at z-distance = 5 mm, x-tilt = 5°. 317
G.22 Array size and misalignment effect dependency investigation (subblock 2.1): case 1. 318
G.23 Array size and misalignment effect dependency investigation (subblock 2.1): case 2. 318
G.24 Array size and misalignment effect dependency investigation (subblock 2.1): case 3. 318
G.25 Mean distance deviation for all localized AMM points for the disc magnet data. . 319
G.26 Mean distance deviation for localized mean AMM point for the disc magnet data. 319
G.27 1° border for mean and maximum angular errors of extended 2D-DFT algorithm for angle information extraction combined with AMM tracking feature. Encoder magnet: sphere. 5 mm z-distance. 320
G.28 1° border for mean and maximum angular errors of single LoG algorithm for angle information extraction combined with AMM tracking feature. Encoder magnet: sphere. 5 mm z-distance. 321

G.29 1° border for mean and maximum angular errors of approximation based algorithm for angle information extraction combined with AMM tracking feature. Encoder magnet: sphere. 5 mm z-distance. 322
G.30 Bar plots for mean and maximum angular errors of extended 2D-DFT algorithm for angle information extraction combined with AMM tracking feature. Encoder magnet: sphere. 5 mm z-distance. 323
G.31 Bar plots for mean and maximum angular errors of single LoG algorithm for angle information extraction combined with AMM tracking feature. Encoder magnet: sphere. 5 mm z-distance. 324
G.32 Bar plots for mean and maximum angular errors of approximation based algorithm for angle information extraction combined with AMM tracking feature. Encoder magnet: sphere. 5 mm z-distance. 325
G.33 Harmonics cancelation test: linear algorithm. 326
G.34 Harmonics cancelation test: linear algorithm (2). 326
G.35 Harmonics cancelation test: linear algorithm (3). 326
G.36 Harmonics cancelation test: nonlinear algorithm. 327
G.37 Harmonics cancelation test: nonlinear algorithm (2). 327
G.38 Harmonics cancelation test: nonlinear algorithm (3). 327
G.39 Misalignment compensation: extreme case. Linear correction. Errors. 328
G.40 Misalignment compensation: extreme case. Linear correction (2). Errors. . . . 328
G.41 Misalignment compensation: extreme case. Nonlinear correction. Errors. . . . 329
G.42 Misalignment compensation: extreme case. Nonlinear correction (2). Errors. . . 329
G.43 Misalignment compensation: extreme case. Nonlinear correction (3). Errors. . . 330
G.44 Misalignment compensation: extreme case. Nonlinear correction (4). Errors. . . 330
G.45 Misalignment compensation: : extreme case. Linear correction. Harmonics, sphere magnet. 331
G.46 Misalignment compensation: extreme case. Linear correction (2). Harmonics, sphere magnet. 331
G.47 Misalignment compensation: extreme case. Linear correction (3). Harmonics, sphere magnet. 331
G.48 Misalignment compensation: extreme case. Linear correction. Harmonics, disc magnet. 332
G.49 Misalignment compensation: extreme case. Linear correction (2). Harmonics, disc magnet. 332
G.50 Misalignment compensation: extreme case. Linear correction (3). Harmonics, disc magnet. 332
G.51 Misalignment compensation: extreme case. Nonlinear correction. Harmonics, sphere magnet. 333
G.52 Misalignment compensation: extreme case. Nonlinear correction (2). Harmonics, sphere magnet. 333

G.53 Misalignment compensation: extreme case. Nonlinear correction (3). Harmonics, sphere magnet. 333
G.54 Misalignment compensation: extreme case. Nonlinear correction. Harmonics, disc magnet. 334
G.55 Misalignment compensation: extreme case. Nonlinear correction (2). Harmonics, disc magnet. 334
G.56 Misalignment compensation: extreme case. Nonlinear correction (3). Harmonics, disc magnet. 334
G.57 Misalignment compensation: worst SOA case. Linear correction. GUI results, sphere magnet. 335
G.58 Misalignment compensation: worst SOA case. Linear correction (2). GUI results, sphere magnet. 335
G.59 Misalignment compensation: worst SOA case. Nonlinear correction. GUI results, sphere magnet. 336
G.60 Misalignment compensation: worst SOA case. Nonlinear correction (2). GUI results, sphere magnet. 336
G.61 Visualization of mechanical misalignment of disc encoder magnet. Worst SOA misalignment case: 5° x-tilt and 2 mm z-distance. 337
G.62 Misalignment compensation: worst SOA case. Linear correction. GUI results, disc magnet. 338
G.63 Misalignment compensation: worst SOA case. Linear correction (2). GUI results, disc magnet. 338
G.64 Misalignment compensation: worst SOA case. Nonlinear correction. GUI results, disc magnet. 339
G.65 Misalignment compensation: worst SOA case. Nonlinear correction (2). GUI results, disc magnet. 339
G.66 Misalignment compensation: worst SOA case. Linear correction. Errors, disc magnet. 340
G.67 Misalignment compensation: worst SOA case. Linear correction. Harmonics, disc magnet. 341
G.68 Misalignment compensation: worst SOA case. Linear correction (2). Harmonics, disc magnet. 341
G.69 Misalignment compensation: worst SOA case. Linear correction (3). Harmonics, disc magnet. 341
G.70 Misalignment comp.: worst SOA case. Nonlin. corr. Errors, disc magnet. 342
G.71 Misalignment comp.: worst SOA case. Nonlin. corr. Harmonics, disc magnet. . 343
G.72 Misalignment comp.: worst SOA case. Nonlin. corr. (2). Harmonics, disc magnet. 343
G.73 Misalignment comp.: worst SOA case. Nonlin. corr. (3). Harmonics, disc magnet. 343

H.1 Sensor array demonstration system. 344

Appendix A: Task description

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Hochschule für Angewandte Wissenschaften Hamburg
Department Informations- und Elektrotechnik
Prof. Dr.-Ing. Karl-Ragmar Riemschneider

14. Mai 2018

Masterthesis Oleg Petrak

Signal processing algorithms for magnetic sensor arrays

Motivation

Magnetic sensors are of particular importance in automotive electronics, e.g. for contactless detection of speed and angle information in the rotating mechanical systems. These types of sensors are versatilely used in engine electronics, anti-lock braking systems and in the applications of comfort area.

Hamburg University of Applied Sciences works in cooperation with partners on the research project „ISAR - signal processing for integrated sensor arrays based on the tunnel magnetoresistive effect for use in automotive electronics“, which is funded by the German Federal Ministry of Education and Research. In this project, signal processing and system architecture for a new type of sensor array shall be developed.

The application of a sensor array makes a new kind of magnetic field aquisition possible. In this way the useful information about the field of a permanent magnet is represented by its spatial position. The spatial resolution obtained by a sensor array provides a possibility for error correction and accuracy improvement. In addition to that a detection and compensation of stray field is aspired.

Objectives

In the masterthesis of Oleg Petrak signal processing algorithms for magnetic sensor arrays shall be developed. The following three targets are aspired in the frame of thesis:

- precise angle calculation
- angular error correction
- stray field suppression

An increase of the system accuracy should be proved in comparison to a standard single-sensor system. The robustness of the algorithms should be maximized.
By application of the algorithms the sensor array system must exhibit an immunity against misalignment effects such as translation and tilt of the encoder magnet. For this purpose a teach-in phase could be included in the system.
Magnetic simulation software should be used, which was developed as part of the preliminary work in the ISAR project.

As a first facultative option in addition to the defined tasks the developed algorithms should be implemented in VHDL and a design of dedicated chip modules shall be performed. The objective is the estimation of implementation effort (e.g. cell count, chip area, clock cycles).
As a second option a contribution for concept for demonstrator could be included in the thesis. The demonstration system is planed for direct visualization of the algorithm results based on a board of discrete sensor chips. For these purposes additional student and/or team members should be involved.

Tasks definition

In the frame of the masterthesis of Mr. Petrak the following work packages shall be carried out:

1. **Introduction**
   - Short technology explanation of AMR-, GMR- and TMR-based sensors, with respect to application related aspects
   - Requirements to angular sensors in automotive applications, fixing of important specification data
   - Comparison of single-sensor system and sensor array system: explanation of advantages and algorithmic potential
2. **Algorithms development**
   - Functional targets:
     - Angle information extraction
     - Stray field suppression and optional detection of stray fields
     - Detection / Correction of misalignment effects
   - Approaches:
     - **Teach-in phase**, e.g. angular/misalignment error correction/detection algorithms, resource limitation not primary, software implementation
     - **Operational phase**, e.g. stray suppression, optimized for speed and resources, ASIC implementation
   - Investigations of algorithms and their optimization
   - Pre-selection of algorithms
3. **Tests of algorithms**
   - Establishment of a test plan for the simulation: definition of the parameter space, variation step size and simulation limits, spherical magnet (dipol), disc magnet
   - Definition of constraints for the algorithm tests such as noise, arithmetical accuracy, offsets, etc.
   - Test of algorithms with simulated and selected data measured from a sensor matrix (if applicable): prove of robustness under application oriented conditions
   - Comparison of results and decision for an algorithm for further implementation
4. **Optional: VHDL implementation and design of chip modules**
   - Implementation of chosen algorithm for angle calculation inclusive stray field suppression
   - Implementation of angular error correction functionalities
   - Synthetization and timing simulations
   - Estimation of ASIC area demand, cell count and clock cycles
5. **Summary and outlook**
   - Evaluation and rating of results
   - Open points and suggestions for continuation or algorithm extensions or improvements
   - Conclusion and contribution to the total project
6. **Attachment**
   - Program codes with implemented algorithms
   - Documentation of simulations and evaluation results
   - Tabular and graphical overview of the algorithms results in detail
   - Short guidance/manual of the toolchain details for continuation

Documentation

The developed algorithmic solutions shall be explained comprehensibly and carefully documented for future use and implementation. The mathematical chain needed for processing of the array signals shall be fully explained and described in detail. All simulation results must be evaluated to a meaningful extent. The realized solutions and the results must be critically evaluated. Approaches to improvements and further work shall be mentioned.

Appendix B: Least squares method

For this thesis, as described in Sections 3.3.3 and 3.7, optimization methods based on linear regression or rather linear curve fitting for estimation of model and functions parameters are used. For the optimization criteria the standard approach called Least Squares Method was chosen: the minimization of the sum of squared errors between the measurement and the model fit. The relevant and most used solving methods of the least squares problem, their required effort and the alternatives are summarized and briefly discussed in this chapter. For more information about the linear regression problem and least squares method refer to [Chap. 19][54]. Furthermore, it is proposed to refer to the research work about numerics from Do Q Lee in the publication [65].

B.1 Problem definition

Suppose the interest is in a observed output value β which have a linear dependence on the observed input values α_1, α_2, ... α_n of a system or an experiment [65]. The problem is to find the unknown model parameters x_1, x_2, ... x_n of a model which describes the dependence between β and observed values α_1, α_2, ... α_n, where the amount of measurements data set is more than the amount of model parameters. The experiment data and the model can be formulated into an overdetermined linear equation system[1]:

$$\mathbf{Ax} = \mathbf{b} - \mathbf{r} \tag{B.1}$$

where **A** is an $m\ x\ n$ matrix contained the α variables or the input values, **b** is a $m\ x\ 1$ vector including the observed output values and **x** is $n\ x\ 1$ vector with the model parameters. Vector **r** represents the errors between the model and the measurement for each data set, called as residuals. The goal of the procedure is to get a solution of the overdetermined system and thus estimate the model parameters x_1, x_2, ... x_n in the best way, what is achieved it the squared Euclidean norm of the vector **r**, formulated as dependent of model parameters vector **x**, becomes minimum [65]:

$$min||\mathbf{r}(\mathbf{x})||_2^2 = min||\mathbf{b} - \mathbf{Ax}||_2^2 \tag{B.2}$$

The solution of **x** is called in this case the Least Squares Solution [65].

[1] All elements in matrices and vectors are $\in \mathbb{R}$

B.2 Least squares solving methods

B.2.1 Direct normal equations solution

The Euclidean norm of existent errors of the formulated Equation system B.1 is at minimum and thus the optimal solution for the model parameter vector **x** is obtained, if

$$\mathbf{A}^T(\mathbf{b} - \mathbf{A}\mathbf{x}) = 0 \tag{B.3}$$

This is the formalism, which says that the difference vector $\mathbf{r}(\mathbf{x}) = (\mathbf{b} - \mathbf{A}\mathbf{x})$ is at minimum if **Ax** is the orthogonal projection of **b** onto column space of the matrix **A** [65]. This is represented by

$$(\mathbf{A}^T\mathbf{A})\mathbf{x} = \mathbf{A}^T\mathbf{b} \tag{B.4}$$

what is the normal equation system. In the simplest way, the vector **x** can be obtained in that case by use of the inverse of $(\mathbf{A}^T\mathbf{A})$:

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \tag{B.5}$$

The matrices $(\mathbf{A}^T\mathbf{A})$ represent a regular $n\ x\ n$ matrix. Hence, the calculation of $(\mathbf{A}^T\mathbf{A})^{-1}$ can be achieved by Gaussian Elimination algorithm[2]. To make sure that the algorithm can always be started the pivoting procedure needs to be included into the decomposition. See more in [42, Chap. 4.5].
A much better approach to solve the normal equation system is the application of Cholesky factorization, what is numerically more suitable and is faster than the standard Gaussian Elimination. Cholesky factorization involves symmetric, positive definite **A** matrices and requires approx. $mn^2 + (1/3)n^3$ flops [65]. The algorithm is described in [65, Section 4.3].

B.2.2 QR factorization

By using the QR factorization method the linear curve fitting problem can be solved much more stable because the performed factorization does not affecting the condition of matrix A [42]. The QR factorization uses orthogonal transformations for the reduction of the least squares problem to a triangular system [65].
The matrix **A** from an overdetermined linear equation system consisting of $m\ x\ n$ elements is decomposed into two matrices: a) an orthogonal matrix $m\ x\ m$ **Q** and an upper triangular $m\ x\ n$ matrix **R**. For the factorization procedure the Gram-Schmidt or Householder transformation method are used.

[2] known as LR decomposition in German and LU decomposition in English literature

Algorithm

The algorithm for the solution of the model parameter vector x by using the QR factorization is as follows [65]:

1. Performing of **QR** factorization of the matrix **A**
2. Computation of $\mathbf{d} = \mathbf{Q}^T\mathbf{b}$
3. Solving of $\mathbf{Rx} = \mathbf{d}$ by backwards substitution.

The required computational effort for QR factorization counts to $2mn^2$ flops [65]. The QR factorization via Householder transformations is promising more stability in comparison to Gram-Schmidt method. A good explanation about the QR factorization using Householders transformations can be found in [42, Chap. 4] as well as in [Chap. 19][54].

B.2.3 Singular value decomposition (SVD)

The singular value decomposition is a relative new tool in the numeric mathematics [71]. It is an universal tool and is used besides parameter estimation i.e. also in image processing as a compression technique [31].
The matrix **A** represented by its SVD is:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{B.6}$$

where the matrix **U** is a unitary $m\ x\ m$ matrix, **V** is unitary $n\ x\ n$ matrix and $\mathbf{\Sigma}$ is a $m\ x\ n$ diagonal matrix. The general solution for a linear equation system $\mathbf{Ax} = \mathbf{b}$ using the SVD is [71]:

$$\mathbf{x} = \mathbf{V}\Sigma^{-1}\mathbf{U}^T\mathbf{b} \tag{B.7}$$

The algorithm for minimum norm solution of the least squares problem based on SVD method is documented in [65, Section 6.1].
The numerical effort of an SVD counts to approximately $2mn^2 + 11n^3$ flops. The SVD algorithm implies usually much more effort but exhibits excellent stability performance [31]. The SVD of a matrix is always existing and can be computed stably. The perturbations in matrix **A** are not amplified by the calculation of SVD because the orthogonal matrices obtained by SVD preserve the 2-norm [65]. In comparison to QR factorization it can also handle rank deficiency of matrices where the QR method may fail [65]. For more information about SVD refer also to [Chap. 42][54].

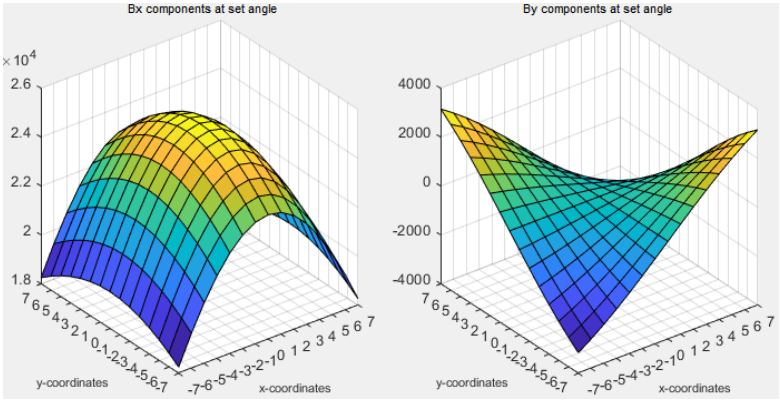
B.3 Least squares alternatives

Since the outliers in the measurement data and among the data points correspondences by using the least squares method are weighed quadratically, alternative optimization approach can be considered in line with further project continuation. The following methods are the most important and can be applied on the present problem:

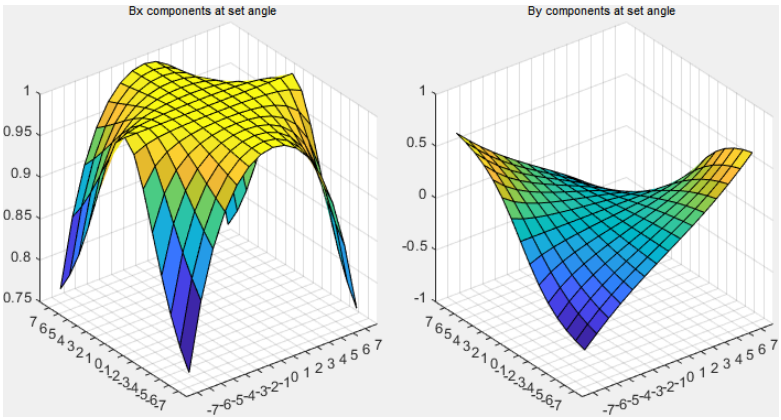
- **Least absolute deviations (LAD / LAE method)**
  Minimization of least absolute errors rather than the least square errors: $min \sum_{n=1}^{N} |r_i|$
- **Maximum-Likelihood method (M-estimator)**
  Here a probability density function / penalty function $p(\mathbf{r})$ is applied on the residuals instead of squaring their errors [88],[97].
- **Least median of squares (LMS)**
  Finds the median value of residual $||r_i||^2$ values and applies an order static filtering of "bad„ residuals. Selection process are repeated and the data samples with the smallest median residual are kept as solution [88].
- **Random Sample Consensuns (RANSAC)**
  Determination of the consensus set and subsequent operation on subsets of the measured values defined by this value. Statistically likely good sets of measurement inliers are used for the estimation procedure [88].

Appendix C: Development of algorithms

C.1 Saturated and not saturated sensor array comparison



(a)



(b)

Figure C.1: Comparison of field components measured by a not saturated (linear) and a saturated system in the simulation. a) $\mathbf{B_x}$ and $\mathbf{B_y}$ components measured by a linear sensor array. b) $\mathbf{B_x}$ and $\mathbf{B_y}$ components measured by a sensor array system operating in the saturation mode. The magnitude information is not included in the measured components and stays for each angular position of the encoder magnet the same. The parabolic behavior is not present in b). $\mathbf{B_x}$ components in b) form a cross-shaped hull (A saturated sensor system was created in the evaluation software which scales the magnitudes to 1.) *Simulation setup: 15 x 15 array, 2 mm z-distance, ideal alignment, sphere magnet.*

C.2 Application of Template Matching algorithm on a saturated sensor array system

See Appendix D for explanation of the control and visualizations panels of GUI evaluation software, which is shown in this section.

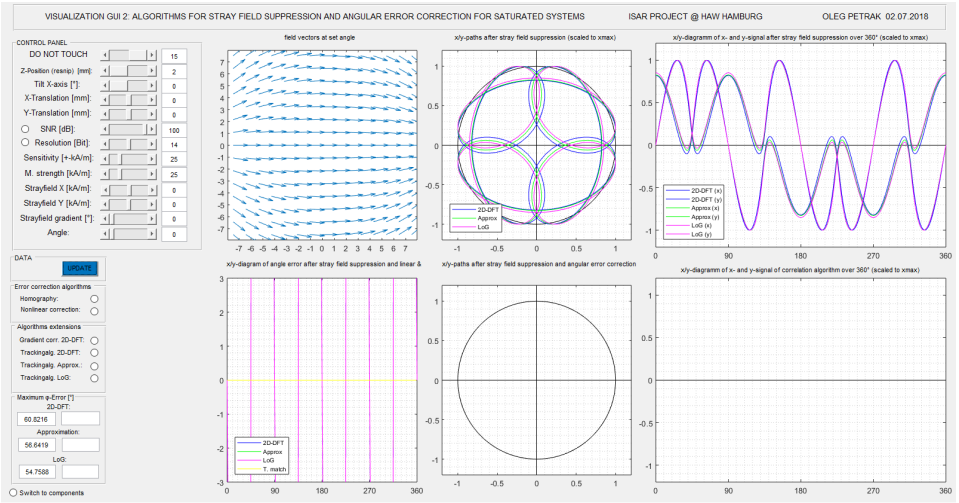


Figure C.2: Application of algorithms on saturated sensor array system. The standard algorithm which requires on linear sensor behavior are not applicable on sensor array systems working in saturation. The progression of extracted x- and y- signals is not sinusoidal.

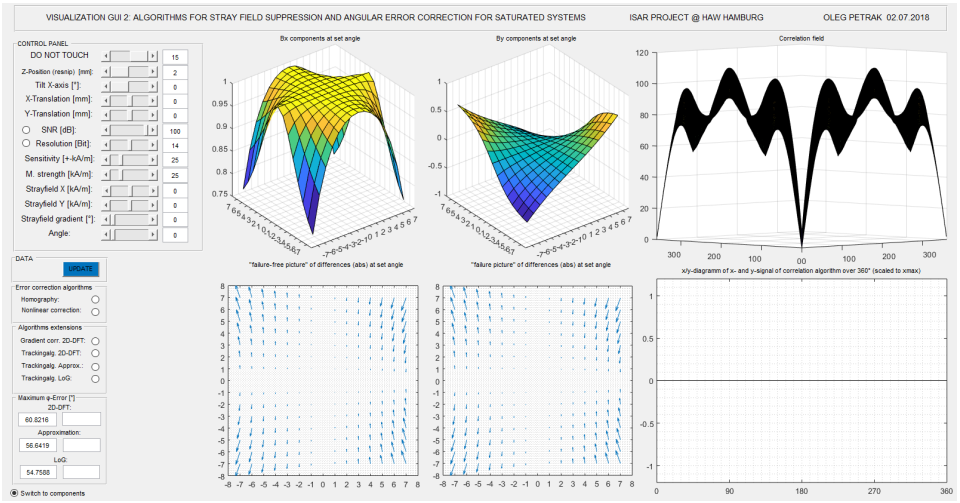
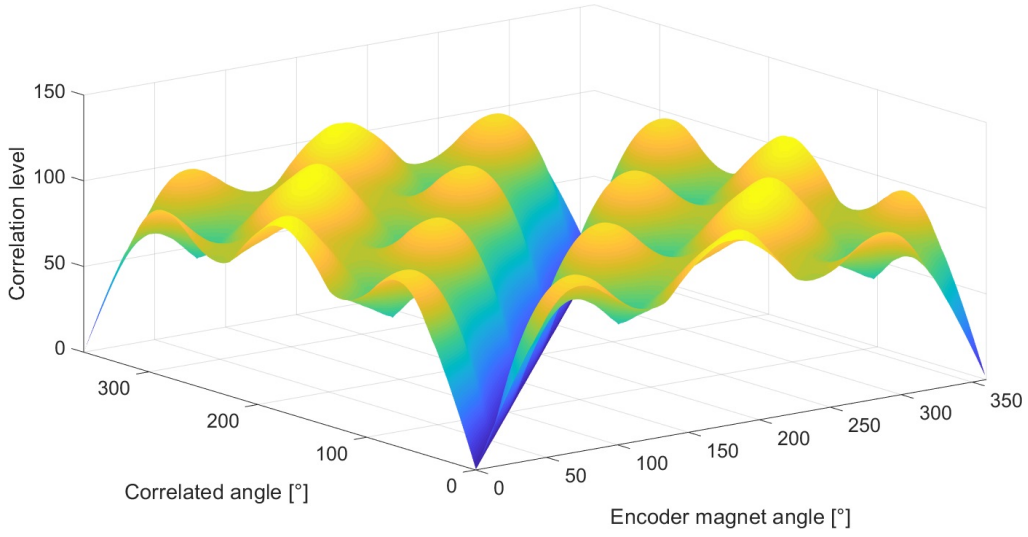
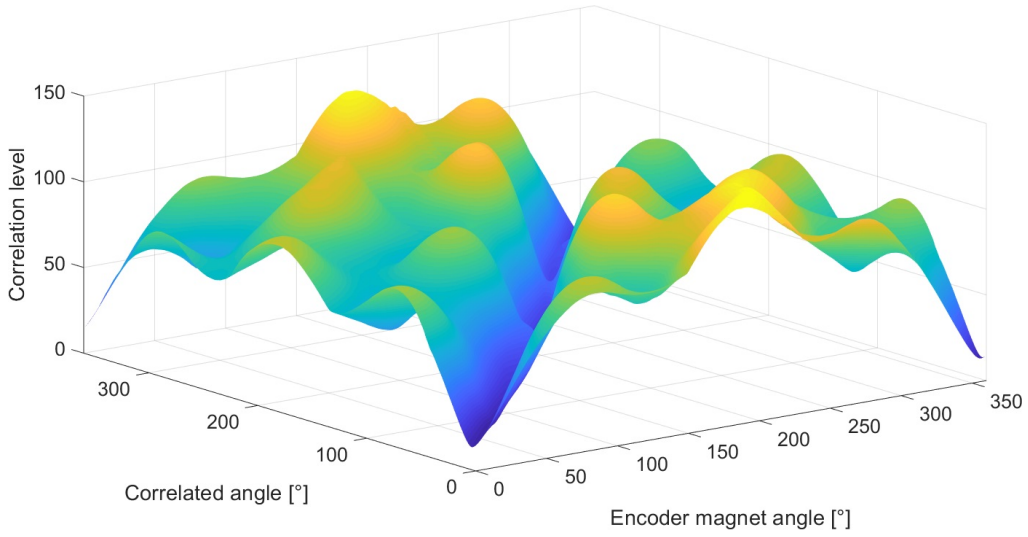


Figure C.3: Application of developed template matching algorithm on saturated sensor array system. In the window 3 (upper graph right) the obtained correlation result with a „magnetic origin picture“ is depicted. In the windows 5 and 6 the difference of field components between a „failure-free picture“ (ideal) and a „failure picture“ which is interfered by a stray field or distorted by a misalignment at a chosen angle is shown.

The correlation results are shown below. Figure C.4(a) depicts the correlation results for ideal conditions and C.4(b) the correlation result obtained in the presence of a homogeneous stray field of 3 kA/m in x- and 1 kA/m in y-direction.

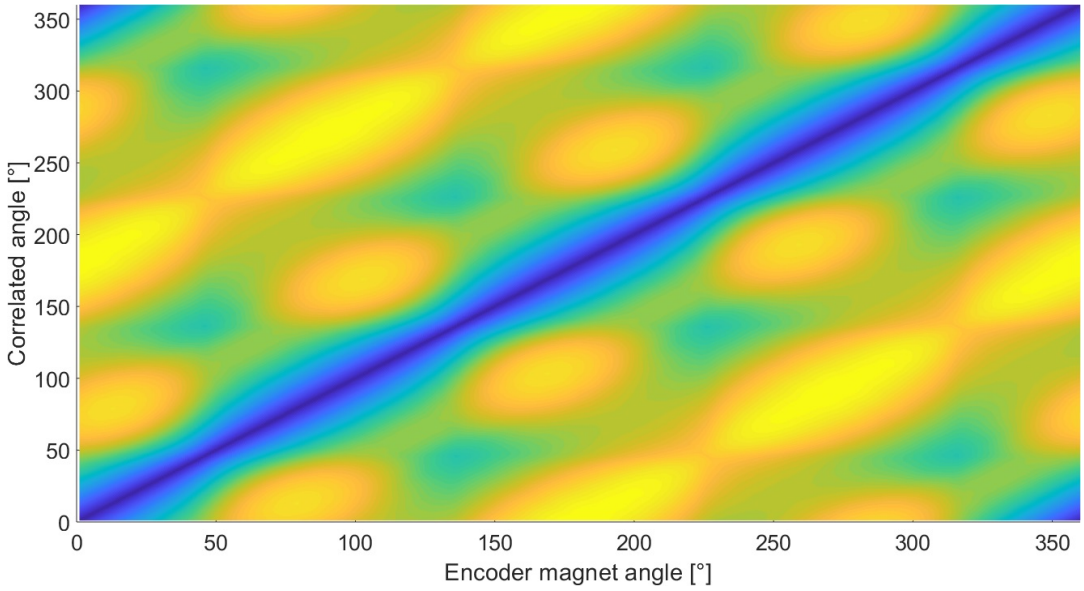


(a)

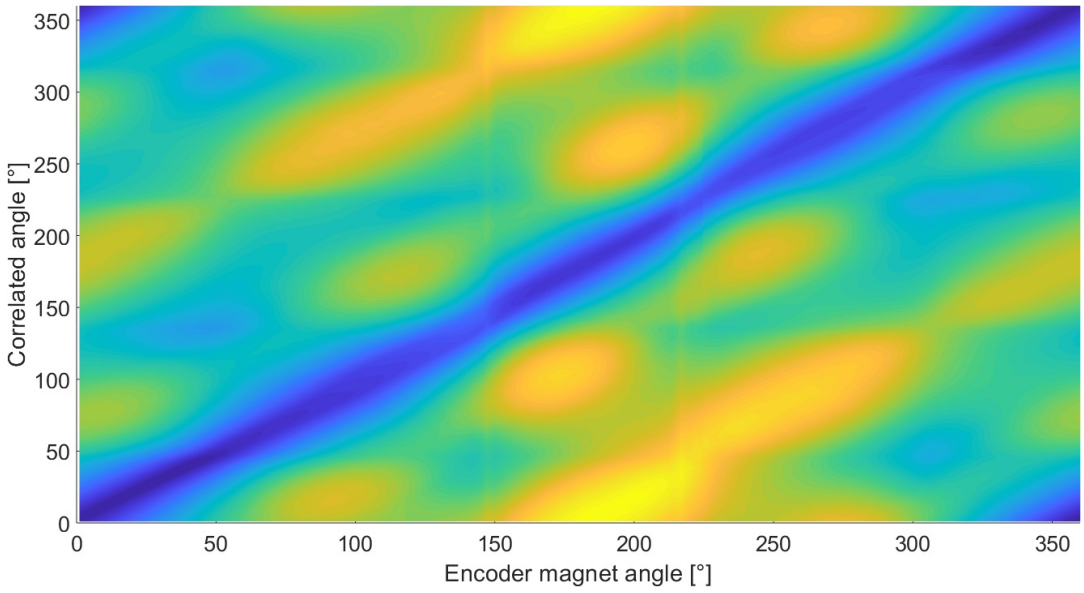


(b)

Figure C.4: Correlation results of performed template matching algorithm test (3D view). a) Ideal correlation image. A clear minimum exists, which corresponds to the angular position of the encoder magnet. The local minima form a straight line. b) Correlation image in presence of specified stray field. Clear minima are still existent. However the local minima line is distorted significantly. The correlated angle obtains an error which is outside the angular requirement specification.

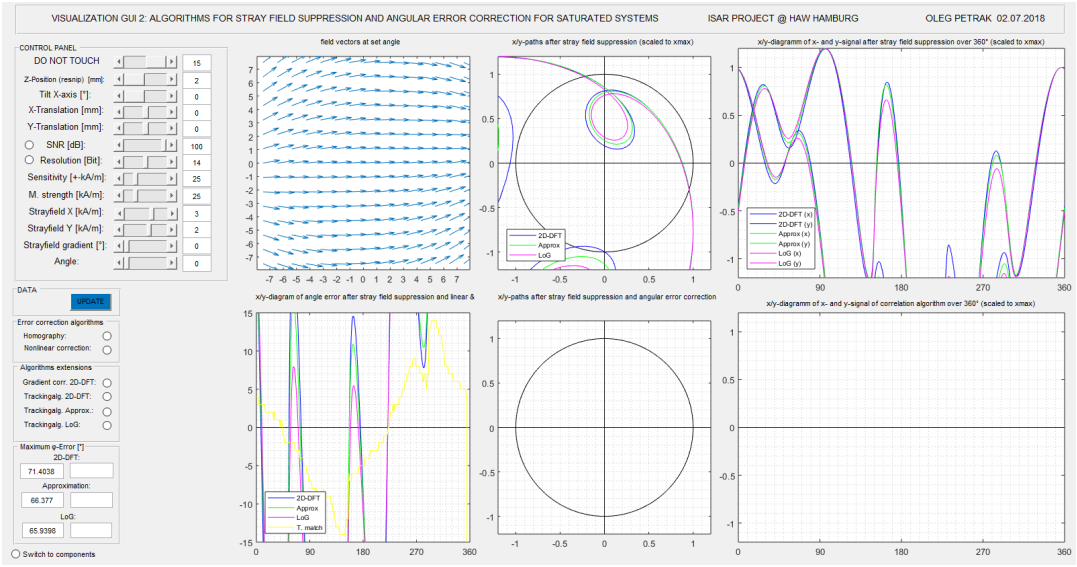


(a)

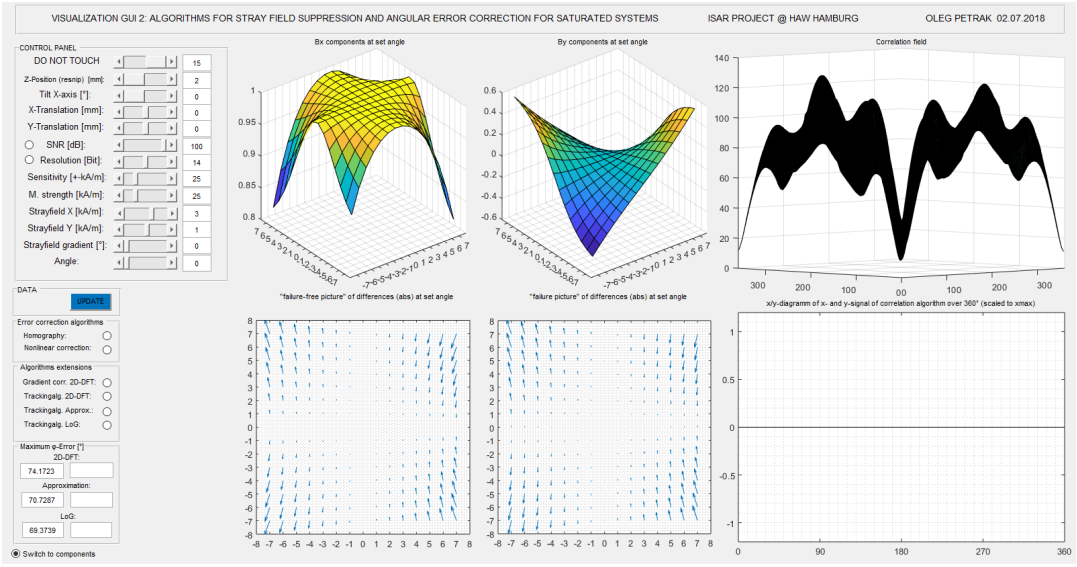


(b)

Figure C.5: Correlation results of performed template matching algorithm test (2D view). a) Ideal correlation image. b) Correlation image in presence of specified stray field. Unfortunately, the distortion of the straightness of the line represent the local minima is too high. The accuracy of the correlated angle is outside of the requirement $\varphi_{err} \leq 1^{\circ}$ (see angular error progression in Figure C.6(a)).



(a)



(b)

Figure C.6: Application of Template Matching algorithm on saturated sensor array system in case of stray field interference. a) Consider the window 6 (left lower graph): progression of caused angular error by interfered stray field of 3 kA/m in x and 1 kA/m in y-direction is shown. The maximum angular error amounts to $\approx 14^{\circ}$. b) Field components $\mathbf{B_x}$ and $\mathbf{B_y}$ at 0° angular position of encoder magnet (windows 1 and 2). Obtained correlation result is shown in window 3 (upper graph right). In the windows 5 and 6 the difference of field components between a „failure-free picture“ (ideal) and a „failure picture“ (here stray field interfered) is shown. The shape formed by field components $\mathbf{B_x}$ and $\mathbf{B_y}$ is distorted (compare to Figure C.3) due the change of the direction of local vectors caused by interfered stray field.

C.3 2D-DFT algorithm for angle information extraction for complex input signals

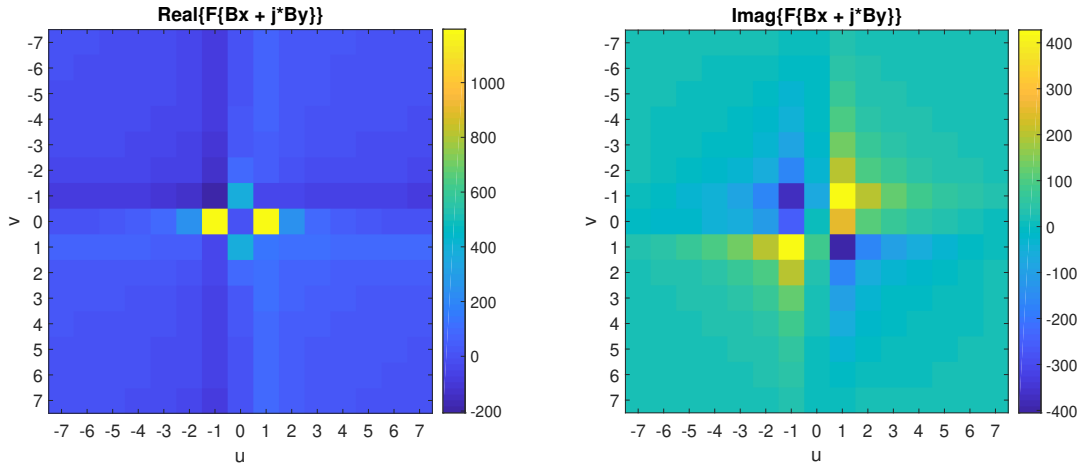


Figure C.7: Spatial frequency components of 2D-DFT applied on Bx- and j*By-data (encoder magnet angle = 0°). Frequency components are shifted to the center by using the command *fftshift()*. Due to complex input signals the real components do not exhibit a complete mirror symmetry. Also the point symmetry is not completely existent in the imaginary parts (see tiny color differences in the blue points).

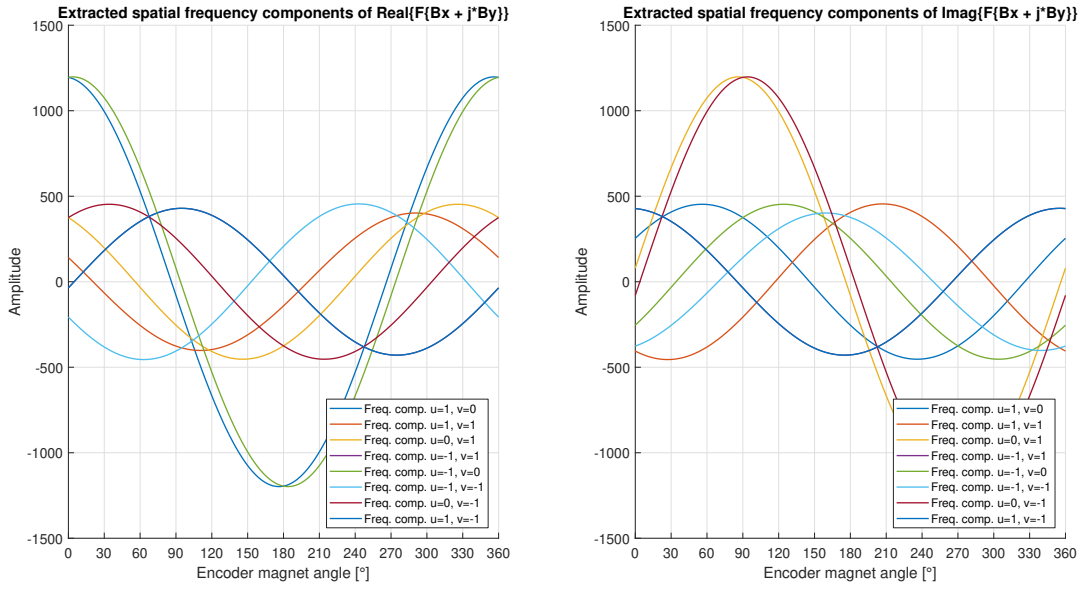


Figure C.8: Progression of spatial frequency components over 360° rotation of encoder magnet. All frequencies vary sinusoidally. The highest amplitudes posse the spatial frequency components Re{F(1,0)} and Re{F(-1,0)} as well as Im{F(0,1)} and Im{F(0,-1)}. An initial phase shift between these frequency components is present. However, it is suspected that this phase shift won't affect the angle calculation if the corresponding components are accumulated. Other combinations can also be considered for the angle calculation.

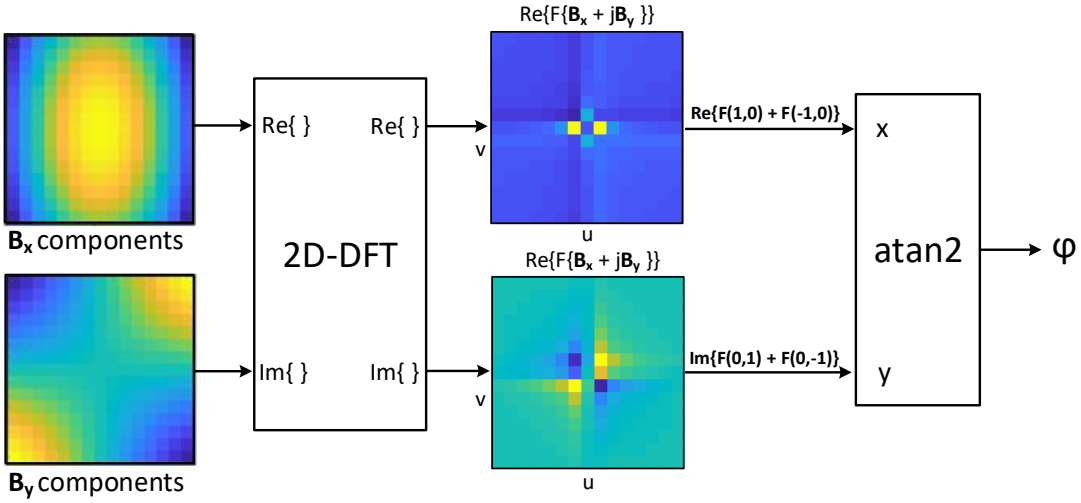


Figure C.9: Signal flow diagram of 2D-DFT based algorithm for angle calculation including stray field suppression which treats the $\mathbf{B_y}$ components as imaginary inputs. The angle is obtained by summation of corresponding first spatial frequencies exclusively in u- and v-direction.

As a conclusion, the formulation for the angle calculation by using the 2D-DFT algorithm for complex input signals results to:

$$\varphi = arctan\left(\frac{Im\{F(0,1)+F(0,-1)\}}{Re\{F(1,0)+F(-1,0)\}}\right) \tag{C.1}$$

where $Im\{F(0,1)+F(0,-1)\}$ and $Re\{F(1,0)+F(-1,0)\}$ are the corresponding the first opposite spatial frequencies in u and v direction of measured field components $\mathbf{B_x}$ and $j\mathbf{B_y}$.

Appendix D: Evaluation software

The developed evaluation software for linear and for saturated sensor systems can be found in the attached DVD-ROM. The simulation data, the actual version of used magnetic simulator [75] and the script files for generated SOA data are located on the attached DVD-ROM as well.

D.1 Front-end

D.1.1 Graphical User Interface: overview

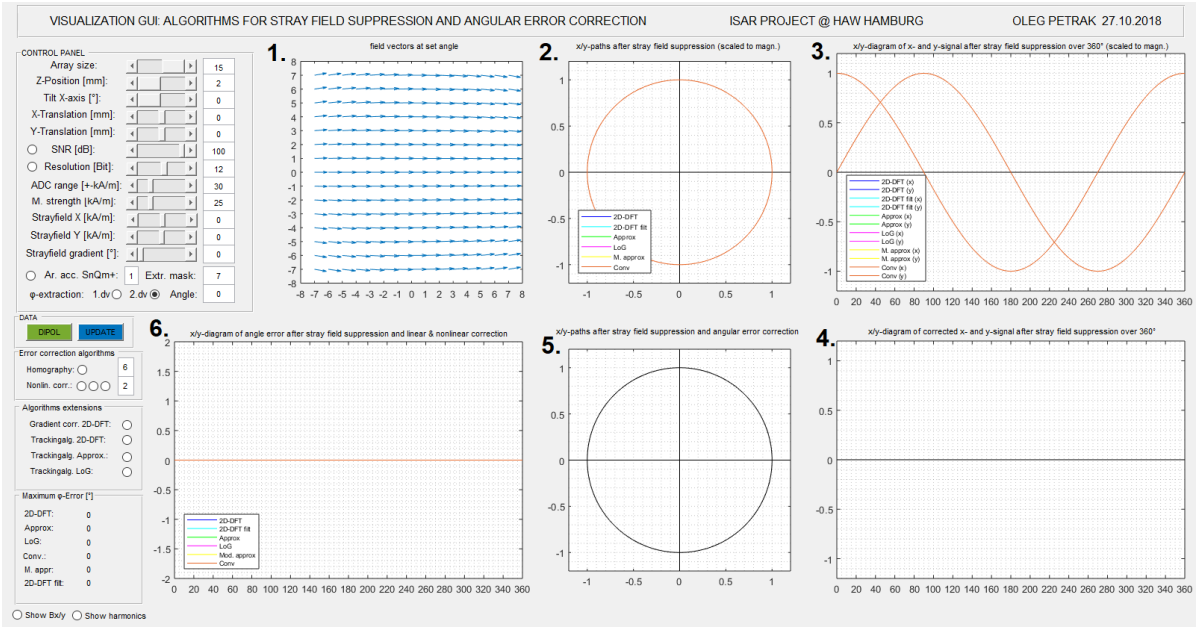
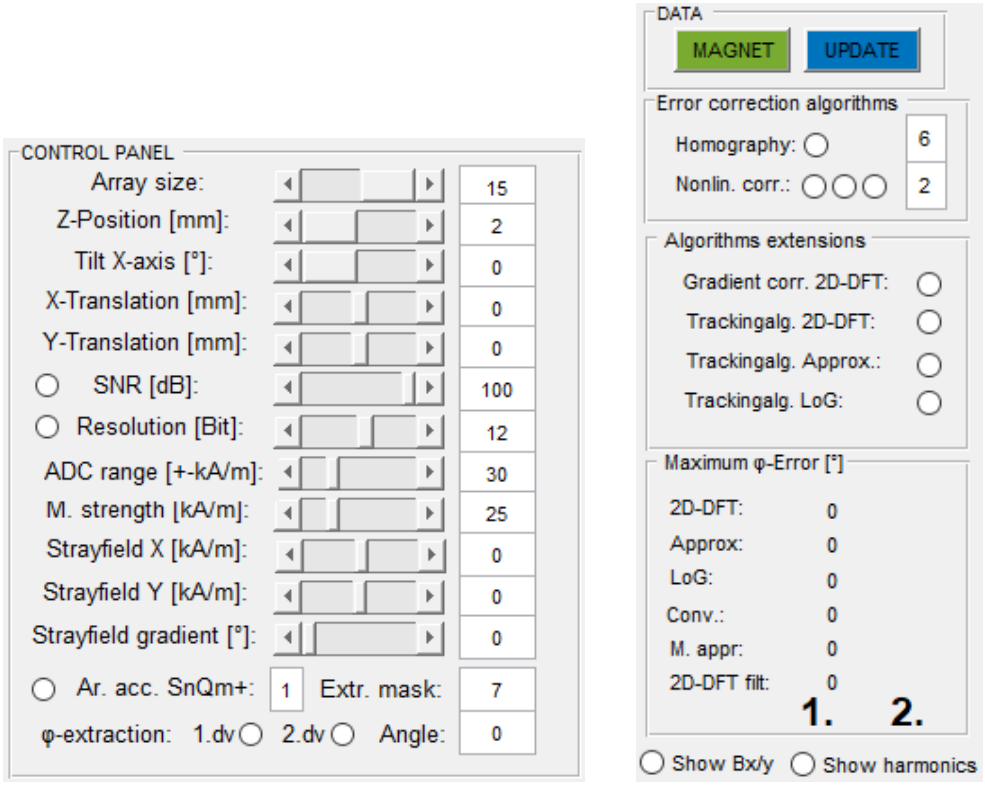


Figure D.1: Graphical user interface for visualization and evaluation purposes created in MATLAB®. In window 1. the field vectors measured by the sensor array are visualized. Window 2. shows the x-/y-signal paths of angle information extraction algorithms. In Window 3. the x-/y-signals over a rotation of encoder magnet over 360°. In Windows 4. and 5. the corresponding content but after the application of angular error correction algorithms is shown. So, 2. and 5. as well as 3. and 4. are directly comparable. In the Window 6. all angular errors (not corrected and corrected) over the full encoder magnet are shown.

D.1.2 Graphical User Interface: control and evaluation panels



(a) Control panel.

(b) Evaluation and Control panel 2.

Figure D.2: Control and evaluation panels.
(a) Control panel for input parameter variations, such as: mechanical misalignment of encoder magnet, noise level and ADC resolution and its dynamic range, strength of the encoder magnet and additional stray fields. The array size and size of the movable extraction mask is controlled in this panel as well. Also the parameter for the fraction part for arithmetic operation in S1Q10(+) format is controlled here. In addition to it the evaluation of the field curvature (2.dv) or global gradient (1.dv, not implemented) can be chosen in this panel.
(b) Evaluation panel with control buttons for update of the software. In this panel the encoder magnet can be chosen and a controlling of parameters for angular error correction algorithms are given: degree of correction and the correction methods (multiple methods for nonlinear approach. Approach based on geometric distortion is the first one. Other approaches were not covered / documented within the work). At the bottom the maximum angular error caused by mechanical misalignment and their corrections can be viewed. Left column are the angular error without correction (1.), right column (2.) what is fade out becomes visible by activation of angular error correction algorithms. Additional button for analysis of harmonics and visualization of measured field components are present.

D.2 Back-end

D.2.1 Main GUI update function

```
%%------------------------------------------------------------------------%%
%------||---||-------||-----||----------||-----||---||---||---||-------%
%------||---||-------||--||--||----------||-------||---||-----||---||-------%
%-----|||||||-----||-----||----||--||--||---------|||||||-----|||||||-------%
%------||----||---|||||||||-----||-||-||---------||----||-----||----||-------%
%------||----||--||----------||---||--||----------||----||-----||----||-------%
%%------------------------------------------------------------------------
%-------------||----------|||||-------||-----------|||||-------------------%
%---------------||---------|||-----------||--||---------||-----||-----------%
%-------------||---------|||||-----------||-----||-----|||||-----------------%
%-------------||-----------|||||-------|||||||-------||--||-----------------%
%-------------||--------|||||-------||-----------||---||-------------------%
%%------------------------------------------------------------------------%%
%%
%%------------------------------------------------------------------------%%
% EVALUATION SOFTWARE FOR ALGORITHMS FOR STRAY FIELD SUPPRESSION AND
% ANGULAR ERROR CORRECTION
%
% Author:           Oleg Petrak
% Version:            V2.0
% Created:            07.2018
% Last update:      25.12.2018
%
%%------------------------------------------------------------------------%%
...
%%------------------------------------------------------------------------
function UPDATE_BUTTON_Callback(hObject, eventdata, handles)
%%------------------------------------------------------------------------%%
% Main update function of the created evaluation software
%%------------------------------------------------------------------------%%
% if no magnet is chosen, choose sphere magnet and set info in the GUI
if ( strcmp(get(handles.MAGNET_BUTTON,'String'),'MAGNET') )
    set(handles.MAGNET_BUTTON,'String',handles.MAGNETSHAPE);
    handles.DATANAME_string = strcat('SOAR_COMPONENTS_',handles.MAGNETSHAPE,'_Z_POS_');
    handles.FLAG_MAGNETSHAPE = 1;
end
% set flag tothe second derivative if no DV flag is chosen
if ( handles.DV1_FLAG == 0 && handles.DV2_FLAG == 0)
    handles.DV2_FLAG = 1;
    set(handles.DV2_BUTTON,'Value',1);
end
%------------------------------ LOAD DATA-------------------------------
handles.SOAR_DATA_string = create_data_string(handles);
handles.SIMDATA = load_data(handles);
% extract the data from area of itnereset depending on the set x- and  y-translation
handles.X_VECTOR_EXTRACT = [(floor(handles.Total_simulation_array_size/2)+1-floor(handles.Array_size/2)):1:(floor(handles.
    Total_simulation_array_size/2)+1+floor(handles.Array_size/2))] + handles.Translat_y_steps ;
handles.Y_VECTOR_EXTRACT = [(floor(handles.Total_simulation_array_size/2)+1-floor(handles.Array_size/2)):1:(floor(handles.
    Total_simulation_array_size/2)+1+floor(handles.Array_size/2))] + handles.Translat_x_steps ;
% store the values in the main buffer
[handles.MAGFIELD_BX_BUFFER, handles.MAGFIELD_BY_BUFFER] = extract_sensor_values(handles);
% save the values into second buffer for further processing. Here, the
% buffer is depending on the chosen amount of sensor elements on the die.
[handles.MAGFIELD_BX, handles.MAGFIELD_BY] = decimate_sensor_values(handles, handles.Interpolation_steps_inbetween);
%-----------------------------END LOAD DATA-------------------------------
% add stray field interference
[handles.MAGFIELD_BX_plus_strayfield, handles.MAGFIELD_BY_plus_strayfield] = add_strayfield(handles);
handles.MAGFIELD_BX_operated = handles.MAGFIELD_BX_plus_strayfield;
handles.MAGFIELD_BY_operated = handles.MAGFIELD_BY_plus_strayfield;
% add noise
if ( handles.Noise_button )
  [handles.MAGFIELD_BX_noise, handles.MAGFIELD_BY_noise] = add_noise(handles);
    handles.MAGFIELD_BX_operated = handles.MAGFIELD_BX_noise;
    handles.MAGFIELD_BY_operated = handles.MAGFIELD_BY_noise;
end
% add gains and offsets deviation to the sensor array elements
if( handles.FLAG_ADD_GAIN_VARIATIONS )
  [handles.MAGFIELD_BX_operated, handles.MAGFIELD_BY_operated] = add_gain_variations(handles);
end
if( handles.FLAG_ADD_OFFSET_VARIATIONS )
  [handles.MAGFIELD_BX_operated, handles.MAGFIELD_BY_operated] = add_offset_variations(handles);
end
%% Digitalization of values
if ( handles.Sampling_button && handles.S1QN_button ) % if sampling and fixed-point aritmetic button are active:
   % sample and convert the data to the fixed-point format.
    handles.MAGFIELD_BX_sampled = sample_components_signed(handles.MAGFIELD_BX_operated, handles.Resolution, -handles.Sensitivity, handles
      .Sensitivity, handles.Array_size_extracted);
    handles.MAGFIELD_BY_sampled = sample_components_signed(handles.MAGFIELD_BY_operated, handles.Resolution, -handles.Sensitivity, handles
      .Sensitivity, handles.Array_size_extracted);
    handles.MAGFIELD_BX_operated = handles.MAGFIELD_BX_sampled;
    handles.MAGFIELD_BY_operated = handles.MAGFIELD_BY_sampled;
```

```
78      [handles.MAGFIELD_BX_operated, handles.MAGFIELD_BY_operated] = normalize_data_to_S1QN(handles);
79  elseif ( handles.Sampling_button ) %otherwise sample only, if sampling button is active.
80      handles.MAGFIELD_BX_sampled = sample_components_signed(handles.MAGFIELD_BX_operated, handles.Resolution, -handles.Sensitivity, handles
            .Sensitivity, handles.Array_size_extracted);
81      handles.MAGFIELD_BY_sampled = sample_components_signed(handles.MAGFIELD_BY_operated, handles.Resolution, -handles.Sensitivity, handles
            .Sensitivity, handles.Array_size_extracted);
82      handles.MAGFIELD_BX_operated = handles.MAGFIELD_BX_sampled;
83      handles.MAGFIELD_BY_operated = handles.MAGFIELD_BY_sampled;
84  end
85  %% Proprocessing...
86  if( handles.FLAG_MEDIAN_FILTERING )
87    [handles.MAGFIELD_BX_operated, handles.MAGFIELD_BY_operated] = median_filt(handles);
88  end
89  if( handles.FLAG_MEAN_FILTERING )
90    [handles.MAGFIELD_BX_operated, handles.MAGFIELD_BY_operated] = mean_filt(handles);
91  end
92  if( handles.FLAG_BUTTERWORTH_FILTERING )
93    [handles.MAGFIELD_BX_operated, handles.MAGFIELD_BY_operated] = Butterworth_filt(handles);
94  end
95  if( handles.FLAG_GAUSS_FILTERING )
96    [handles.MAGFIELD_BX_operated, handles.MAGFIELD_BY_operated] = Gauss_filt(handles);
97  end
98  %% Data extraction for algorithms extension  point algorithms
99  % Localization of the AMM points and extraction of the data for angle information from the set area of covered by the extraction mask
100 [handles.AMM_x_coordinates, handles.AMM_y_coordinates, handles.AMM_warnings] = track_AMM_point(handles);
101 [handles.MAGFIELD_BX_operated_AMM, handles.MAGFIELD_BY_operated_AMM] = extract_sensor_values_from_AMM(handles);
102 % Calculation of the x- and y-signals by the algorithms for angle information extraction:
103 [handles.x_values_DFT, handles.y_values_DFT] = DFT_xy_calculation(handles, handles.DFT_components_width); % 2D-DFT based algorithm
104 [handles.x_values_LOG, handles.y_values_LOG] = LOG_xy_calculation(handles); % Single LoG
105 [handles.x_values_APPROX, handles.y_values_APPROX] = Approx_xy_calculation(handles); % standard approximation
106 [handles.x_values_Dipol, handles.y_values_Dipol] = Flat_approx_xy_calculation(handles); % modified approximation (K. Juenemann, H. Pape
        equation)
107 [handles.FFT2D_ARRAY_FILT, handles.x_values_DFT_FILT, handles.y_values_DFT_FILT] = DFT_filtering_and_xy_calculation(handles); % 2D-DFT
        filtering algorithm (K-R Riemschneider, T.Schuethe)
108 [handles.ARRAY_BX_Conv_filt, handles.ARRAY_BY_Conv_filt, handles.x_values_Conv, handles.y_values_Conv] = Convolution_xy_calculation(
        handles);
109 % Misalignment correction by homography approach
110 [handles.x_values_DFT_hom_corrected, handles.y_values_DFT_hom_corrected, handles.H_DFT] = homography_correction(handles.x_values_DFT/(max(
        handles.x_values_DFT)),handles.y_values_DFT/(max(handles.y_values_DFT)),handles.X_VECTOR_FAILUREFREE, handles.Y_VECTOR_FAILUREFREE
        ,[1:1:361],handles.homography_order);
111 [handles.x_values_APPROX_hom_corrected, handles.y_values_APPROX_hom_corrected, handles.H_APPROX] = homography_correction(handles.
        x_values_APPROX/(max(handles.x_values_APPROX)),handles.y_values_APPROX/(max(handles.y_values_APPROX)),handles.X_VECTOR_FAILUREFREE,
        handles.Y_VECTOR_FAILUREFREE,[1:1:361],handles.homography_order);
112 [handles.x_values_LOG_hom_corrected, handles.y_values_LOG_hom_corrected, handles.H_LOG] = homography_correction(handles.x_values_LOG/(max(
        handles.x_values_LOG)),handles.y_values_LOG/(max(handles.y_values_LOG)),handles.X_VECTOR_FAILUREFREE, handles.Y_VECTOR_FAILUREFREE
        ,[1:1:361],handles.homography_order);
113 [handles.x_values_Conv_hom_corrected, handles.y_values_Conv_hom_corrected, handles.H_Conv] = homography_correction(handles.x_values_Conv/(
        max(handles.x_values_Conv)),handles.y_values_Conv/(max(handles.y_values_Conv)),handles.X_VECTOR_FAILUREFREE, handles.
        Y_VECTOR_FAILUREFREE,[1:1:361],handles.homography_order);
114 [handles.x_values_DFT_FILT_hom_corrected, handles.y_values_DFT_FILT_hom_corrected, handles.H_DFT_FILT] = homography_correction(handles.
        x_values_DFT_FILT/(max(handles.x_values_DFT_FILT)),handles.y_values_DFT_FILT/(max(handles.y_values_DFT_FILT)),handles.
        X_VECTOR_FAILUREFREE, handles.Y_VECTOR_FAILUREFREE,[1:1:361],handles.homography_order);
115 [handles.x_values_Dipol_hom_corrected, handles.y_values_Dipol_hom_corrected, handles.H_Dipol] = homography_correction(handles.
        x_values_Dipol/(max(handles.x_values_Dipol)),handles.y_values_Dipol/(max(handles.y_values_Dipol)),handles.X_VECTOR_FAILUREFREE,
        handles.Y_VECTOR_FAILUREFREE,[1:1:361],handles.homography_order);
116 if ( handles.Homography_corr )
117     %Additional correction by nonlinear correction in case if homography correction in the pre-step
118     [handles.x_values_DFT_nonlin_corrected, handles.y_values_DFT_nonlin_corrected] = nonlinear_correction_geometric_distortion(handles.
          x_values_DFT_hom_corrected,handles.y_values_DFT_hom_corrected,handles.X_VECTOR_FAILUREFREE, handles.Y_VECTOR_FAILUREFREE,181,handles
          .nonlinear_correction_order);
119     [handles.x_values_APPROX_nonlin_corrected, handles.y_values_APPROX_nonlin_corrected] = nonlinear_correction_geometric_distortion(
          handles.x_values_APPROX_hom_corrected,handles.y_values_APPROX_hom_corrected,handles.X_VECTOR_FAILUREFREE, handles.
          Y_VECTOR_FAILUREFREE,181,handles.nonlinear_correction_order);
120     [handles.x_values_LOG_nonlin_corrected, handles.y_values_LOG_nonlin_corrected] = nonlinear_correction_geometric_distortion(handles.
          x_values_LOG_hom_corrected,handles.y_values_LOG_hom_corrected,handles.X_VECTOR_FAILUREFREE, handles.Y_VECTOR_FAILUREFREE,181,handles
          .nonlinear_correction_order);
121     [handles.x_values_Conv_nonlin_corrected, handles.y_values_Conv_nonlin_corrected] = nonlinear_correction_geometric_distortion(handles.
          x_values_Conv_hom_corrected,handles.y_values_Conv_hom_corrected,handles.X_VECTOR_FAILUREFREE, handles.Y_VECTOR_FAILUREFREE,181,
          handles.nonlinear_correction_order);
122     [handles.x_values_DFT_FILT_nonlin_corrected, handles.y_values_DFT_FILT_nonlin_corrected] = nonlinear_correction_geometric_distortion(
          handles.x_values_DFT_FILT_hom_corrected,handles.y_values_DFT_FILT_hom_corrected,handles.X_VECTOR_FAILUREFREE, handles.
          Y_VECTOR_FAILUREFREE,181,handles.nonlinear_correction_order);
123     [handles.x_values_Dipol_nonlin_corrected, handles.y_values_Dipol_nonlin_corrected] = nonlinear_correction_geometric_distortion(handles
          .x_values_Dipol_hom_corrected,handles.y_values_Dipol_hom_corrected,handles.X_VECTOR_FAILUREFREE, handles.Y_VECTOR_FAILUREFREE,181,
          handles.nonlinear_correction_order);
124 else
125     % Direct correction by nonlinear algorithm without homography correction in the pre-step
126     [handles.x_values_DFT_nonlin_corrected, handles.y_values_DFT_nonlin_corrected] = nonlinear_correction_geometric_distortion(handles.
          x_values_DFT/(max(handles.x_values_DFT)),handles.y_values_DFT/(max(handles.y_values_DFT)),handles.X_VECTOR_FAILUREFREE, handles.
          Y_VECTOR_FAILUREFREE,181,handles.nonlinear_correction_order);
127     [handles.x_values_APPROX_nonlin_corrected, handles.y_values_APPROX_nonlin_corrected] = nonlinear_correction_geometric_distortion(
          handles.x_values_APPROX/(max(handles.x_values_APPROX)),handles.y_values_APPROX/(max(handles.y_values_APPROX)),handles.
          X_VECTOR_FAILUREFREE, handles.Y_VECTOR_FAILUREFREE,181,handles.nonlinear_correction_order);
128     [handles.x_values_LOG_nonlin_corrected, handles.y_values_LOG_nonlin_corrected] = nonlinear_correction_geometric_distortion(handles.
          x_values_LOG/(max(handles.x_values_LOG)),handles.y_values_LOG/(max(handles.y_values_LOG)),handles.X_VECTOR_FAILUREFREE, handles.
          Y_VECTOR_FAILUREFREE,181,handles.nonlinear_correction_order);
```

```
129     [handles.x_values_Conv_nonlin_corrected, handles.y_values_Conv_nonlin_corrected] = nonlinear_correction_geometric_distortion(handles.
          x_values_Conv/(max(handles.x_values_Conv)),handles.y_values_Conv/(max(handles.y_values_Conv)),handles.X_VECTOR_FAILUREFREE, handles.
          Y_VECTOR_FAILUREFREE,181,handles.nonlinear_correction_order);
130     [handles.x_values_DFT_FILT_nonlin_corrected, handles.y_values_DFT_FILT_nonlin_corrected] = nonlinear_correction_geometric_distortion(
          handles.x_values_DFT_FILT/(max(handles.x_values_DFT_FILT)),handles.y_values_DFT_FILT/(max(handles.y_values_DFT_FILT)),handles.
          X_VECTOR_FAILUREFREE, handles.Y_VECTOR_FAILUREFREE,181,handles.nonlinear_correction_order);
131     [handles.x_values_Dipol_nonlin_corrected, handles.y_values_Dipol_nonlin_corrected] = nonlinear_correction_geometric_distortion(handles
          .x_values_Dipol/(max(handles.x_values_Dipol)),handles.y_values_Dipol/(max(handles.y_values_Dipol)),handles.X_VECTOR_FAILUREFREE,
          handles.Y_VECTOR_FAILUREFREE,181,handles.nonlinear_correction_order);
132 end
133 % Calculation of angular errors of all considered algorithms
134 [handles.Dipol_Error_normal, handles.DFT_Error_normal, handles.DFT_FILT_Error_normal, handles.LOG_Error_normal, handles.
          APPROX_Error_normal, handles.Conv_Error_normal,...
135  handles.Dipol_Error_hom_corr, handles.DFT_Error_hom_corr, handles.DFT_FILT_Error_hom_corr, handles.APPROX_Error_hom_corr, handles.
          LOG_Error_hom_corr, handles.Conv_Error_hom_corr,...
136  handles.Dipol_Error_nonlin_corr, handles.DFT_Error_nonlin_corr, handles.DFT_FILT_Error_nonlin_corr, handles.APPROX_Error_nonlin_corr,
          handles.LOG_Error_nonlin_corr, handles.Conv_Error_nonlin_corr,...
137  handles.Dipol_Error_nonlin_corr_2, handles.DFT_Error_nonlin_corr_2, handles.DFT_FILT_Error_nonlin_corr_2, handles.
          APPROX_Error_nonlin_corr_2, handles.LOG_Error_nonlin_corr_2, handles.Conv_Error_nonlin_corr_2] = calculate_angular_errors(handles);
138 % Update GUI storage
139 guidata(hObject,handles);
140 % Save the handles into the workspace
141 assignin('base', 'handles', handles);
142 % Update visuaalization
143 update_GUI_plots(handles);
```

Listing D.1: Main GUI update function: UPDATE_BUTTON_Callback(hObject, eventdata, handles)

D.2.2 Angle information extraction

2D-DFT algorithm for angle information extraction incl. stray field suppression

```
1  function [x_values_DFT, y_values_DFT] = DFT_xy_calculation(handles, WIDTH)
2  %%------------------------------------------------------------------------%%
3  % 2D-DFT based algorithm for angle information extraction.
4  %
5  % Author:          Oleg Petrak
6  % Last update:     25.12.2018
7  %
8  % INPUT: handles data base and value for the amount of spatial freq. comp.
9  % in x- and y- direction for consideration in the angle calculation (WIDTH).
10 % OUTPUT: x- and y- signals needed for angle calculation.
11 %%------------------------------------------------------------------------%%
12
13 % Consider the scaling 1/N^2:
14 SCALING_FACTOR = handles.Array_size_extracted^2;
15
16 if (handles.Alg_extension_DFT_2 == 1) % operation on data from extraction mask (tracking extension):
17     if ( handles.Alg_extension_DFT_1 ) % include the the global gradient correction (realignment of field components):
18         for Angle = 0:1:360
19             % PERFORM FFT
20             %handles.FFT2D_ARRAY_X_TRACKING(:,:,Angle+1) = Fourier_transformation_2D(gradient_correction(handles.MAGFIELD_BX_operated_AMM
       (:,:,Angle+1),handles.Spatial_filtermask),handles.FI_FLAG);
21             %handles.FFT2D_ARRAY_Y_TRACKING(:,:,Angle+1) = Fourier_transformation_2D(gradient_correction(handles.MAGFIELD_BY_operated_AMM
       (:,:,Angle+1),handles.Spatial_filtermask),handles.FI_FLAG);
22             handles.FFT2D_ARRAY_X_TRACKING(:,:,Angle+1) = fft2(gradient_correction(handles.MAGFIELD_BX_operated_AMM(:,:,Angle+1),handles.
       Spatial_filtermask))/SCALING_FACTOR;
23             handles.FFT2D_ARRAY_Y_TRACKING(:,:,Angle+1) = fft2(gradient_correction(handles.MAGFIELD_BY_operated_AMM(:,:,Angle+1),handles.
       Spatial_filtermask))/SCALING_FACTOR;
24         end
25     else % otherwise operate on standard input data:
26         for Angle = 0:1:360
27             % PERFORM FFT
28             %handles.FFT2D_ARRAY_X_TRACKING(:,:,Angle+1) = Fourier_transformation_2D(handles.MAGFIELD_BX_operated_AMM(:,:,Angle+1),handles
       .FI_FLAG);
29             %handles.FFT2D_ARRAY_Y_TRACKING(:,:,Angle+1) = Fourier_transformation_2D(handles.MAGFIELD_BY_operated_AMM(:,:,Angle+1),handles
       .FI_FLAG);
30             handles.FFT2D_ARRAY_X_TRACKING(:,:,Angle+1) = fft2(handles.MAGFIELD_BX_operated_AMM(:,:,Angle+1))/SCALING_FACTOR;
31             handles.FFT2D_ARRAY_Y_TRACKING(:,:,Angle+1) = fft2(handles.MAGFIELD_BY_operated_AMM(:,:,Angle+1))/SCALING_FACTOR;
32         end
33     end
34     % Calculation of x- and y-signal for the angle calculation:
35     for Angle = 0:1:360
36         handles.FFT2D_ARRAY_X_TRACKING(1,1,Angle+1) =0;
37         % Taking the set amount of spatial frequencies in x- and y-direction from Bx components:
38         handles.BX_DFT_SINUSCOMPONENTS(Angle+1) = sum(imag(handles.FFT2D_ARRAY_X_TRACKING(1:(WIDTH),1,Angle+1))) + sum(imag(handles.
       FFT2D_ARRAY_X_TRACKING(1,1:(WIDTH),Angle+1)));
39         handles.BX_DFT_COSINUSCOMPONENTS(Angle+1) = sum(real(handles.FFT2D_ARRAY_X_TRACKING(1:(WIDTH),1,Angle+1))) + sum(real(handles.
       FFT2D_ARRAY_X_TRACKING(1,1:(WIDTH),Angle+1)));
40         handles.FFT2D_ARRAY_Y_TRACKING(1,1,Angle+1) =0;
41         % Taking the set amount of spatial frequencies in x- and y-direction from By components:
```

```
42          handles.BY_DFT_SINUSCOMPONENTS(Angle+1) = sum(imag(handles.FFT2D_ARRAY_Y_TRACKING(1:(WIDTH),1,Angle+1))) +  sum(imag(handles.
        FFT2D_ARRAY_Y_TRACKING(1,1:(WIDTH),Angle+1)));
43          handles.BY_DFT_COSINUSCOMPONENTS(Angle+1) = sum(real(handles.FFT2D_ARRAY_Y_TRACKING(1:(WIDTH),1,Angle+1))) +  sum(real(handles.
        FFT2D_ARRAY_Y_TRACKING(1,1:(WIDTH),Angle+1)));
44      end
45  else % operation on full array data
46      if ( handles.Alg_extension_DFT_1 )  % include the the global gradient correction (realignment of field components):
47          for Angle = 0:1:360
48              % PERFORM FFT
49              handles.FFT2D_ARRAY_X(:,:,Angle+1) = fft2(gradient_correction(handles.MAGFIELD_BX_operated(:,:,Angle+1),handles.
        Array_size_extracted))/SCALING_FACTOR;
50              handles.FFT2D_ARRAY_Y(:,:,Angle+1) = fft2(gradient_correction(handles.MAGFIELD_BY_operated(:,:,Angle+1),handles.
        Array_size_extracted))/SCALING_FACTOR;
51              %handles.FFT2D_ARRAY_X(:,:,Angle+1) = Fourier_transformation_2D(gradient_correction(handles.MAGFIELD_BX_operated(:,:,Angle+1),
        handles.Array_size_extracted),handles.FI_FLAG);
52              %handles.FFT2D_ARRAY_Y(:,:,Angle+1) = Fourier_transformation_2D(gradient_correction(handles.MAGFIELD_BY_operated(:,:,Angle+1),
        handles.Array_size_extracted),handles.FI_FLAG);
53          end
54      else % otherwise operate on standard input data:
55          for Angle = 0:1:360
56              % PERFORM FFT
57              %handles.FFT2D_ARRAY_X(:,:,Angle+1) = Fourier_transformation_2D((handles.MAGFIELD_BX_operated(:,:,Angle+1)),handles.FI_FLAG);
58              %handles.FFT2D_ARRAY_Y(:,:,Angle+1) = Fourier_transformation_2D((handles.MAGFIELD_BY_operated(:,:,Angle+1)),handles.FI_FLAG);
59              handles.FFT2D_ARRAY_X(:,:,Angle+1) = fft2((handles.MAGFIELD_BX_operated(:,:,Angle+1)))/SCALING_FACTOR;
60              handles.FFT2D_ARRAY_Y(:,:,Angle+1) = fft2((handles.MAGFIELD_BY_operated(:,:,Angle+1)))/SCALING_FACTOR;
61          end
62      end
63      % Calculation of x- and y-signal for the angle calculation:
64      for Angle = 0:1:360
65          handles.FFT2D_ARRAY_X(1,1,Angle+1) =0;
66          % Taking the set amount of spatial frequencies in x- and y-direction from Bx components:
67          handles.BX_DFT_SINUSCOMPONENTS(Angle+1) = sum(imag(handles.FFT2D_ARRAY_X(1:(WIDTH),1,Angle+1))) + sum(imag(handles.FFT2D_ARRAY_X
        (1,1:(WIDTH),Angle+1)));
68          handles.BX_DFT_COSINUSCOMPONENTS(Angle+1) = sum(real(handles.FFT2D_ARRAY_X(1:(WIDTH),1,Angle+1))) + sum(real(handles.FFT2D_ARRAY_X
        (1,1:(WIDTH),Angle+1)));
69          handles.FFT2D_ARRAY_Y(1,1,Angle+1) =0;
70          % Taking the set amount of spatial frequencies in x- and y-direction from Bx components:
71          handles.BY_DFT_SINUSCOMPONENTS(Angle+1) = sum(imag(handles.FFT2D_ARRAY_Y(1:(WIDTH),1,Angle+1))) +  sum(imag(handles.FFT2D_ARRAY_Y
        (1,1:(WIDTH),Angle+1)));
72          handles.BY_DFT_COSINUSCOMPONENTS(Angle+1) = sum(real(handles.FFT2D_ARRAY_Y(1:(WIDTH),1,Angle+1))) +  sum(real(handles.
        FFT2D_ARRAY_Y(1,1:(WIDTH),Angle+1)));
73      end
74  end
75  % Calculate and output the x- and y-signals:
76  x_values_DFT =  -(handles.BX_DFT_COSINUSCOMPONENTS + handles.BX_DFT_SINUSCOMPONENTS);
77  y_values_DFT =  -(handles.BY_DFT_COSINUSCOMPONENTS + handles.BY_DFT_SINUSCOMPONENTS);
78  end
```

Listing D.2: 2D-DFT based algorithm for angle information extraction

```
1  function [OUTPUT_MATRIX] = gradient_correction(INPUT_MATRIX,ARRAY_SIZE)
2  %%-------------------------------------------------------------------------%%
3  % Function for subtraction of global gradient in x and y direction from
4  % the sensor array data (Bx or By).
5  %
6  % Author:          Oleg Petrak
7  % Last update:     25.12.2018
8  %
9  % INPUT: Data array (Bx or By).
10 % OUTPUT: Data array free of global gradient in x- and y-direction.
11 %%-------------------------------------------------------------------------%%
12
13 %% Creation of meshgrids
14 ARRAY_WIDTH  = fix( ARRAY_SIZE / 2);
15 MESH = [-ARRAY_WIDTH:1:ARRAY_WIDTH];
16 [X,Y] = meshgrid(MESH);
17 %% Estimation of the 2D linear equation and subtraction of the global gradient
18 POL_COEFFS = field_approximation_middle_shifted_glob_gradients_only(INPUT_MATRIX);
19 %% Global gradient subtraction and output of realigned field components:
20 OUTPUT_MATRIX = INPUT_MATRIX - POL_COEFFS(2).*X - POL_COEFFS(3).*Y;
21 end
```

Listing D.3: Extension for the 2D-DFT based algorithm (global gradient correction / realignment of field components)

```
1  function [P_VECTOR] = field_approximation_middle_shifted_glob_gradients_only(INPUT_FIELD)
2  %%-------------------------------------------------------------------------%%
3  % Function for estimation of the coefficients of 2D linear equation
4  % Coordinates origin is in the center of the sensor array.
5  %
6  % Author:          Oleg Petrak
7  % Last update:     25.12.2018
```

```
%
% INPUT: Data array (Bx or By).
% OUTPUT: Coefficients of 2D linear equation.
%%------------------------------------------------------------------------%%
%% A has the form:
%[ 1 x1 y1
%  1 x2 y2
%  1 x3 y3
%...]
%% Creation of meshgrids
FIELD_WIDTH  = fix(length(INPUT_FIELD) / 2);
MESH = [-FIELD_WIDTH:1:FIELD_WIDTH];
%% Vectors allocation
X_VECTOR = zeros(1,numel(INPUT_FIELD))';
Y_VECTOR = zeros(1,numel(INPUT_FIELD))';
b = zeros(1,numel(INPUT_FIELD))';
ONES_VECTOR = ones(1,length(INPUT_FIELD(1,:))*length(INPUT_FIELD(:,1)))';
%% Filling of the vectors x and y with meshgrid coordinates
for n = 1:1:length(INPUT_FIELD(:,1))
    X_VECTOR((1:1:length(INPUT_FIELD(:,1)))+(n-1)*length(INPUT_FIELD(:,1))) = MESH';
end
for n = 1:1:length(INPUT_FIELD(:,1))
    Y_VECTOR((1:1:length(INPUT_FIELD(:,1)))+(n-1)*length(INPUT_FIELD(:,1))) = ones(length(INPUT_FIELD(:,1)),1).*MESH(n);% = 1:1:length(
      INPUT_FIELD(:,1));
end
%% Creation of the matrix A
A = [ONES_VECTOR,X_VECTOR, Y_VECTOR];
%% Filling of the vector b with input values corresponding to the coordinates in the matrix A
for n=0:1:length(INPUT_FIELD(1,:))-1
    for m = 0:1:length(INPUT_FIELD(:,1))-1
        b(m +1 + n*length(INPUT_FIELD(1,:))) = INPUT_FIELD(n+1,m+1);
    end
end
%% Solve by using SVD:
P_VECTOR = A\b;
end
```

Listing D.4: Function for estimation of 2D linear equations

Approximation algorithm for angle information extraction incl. stray field suppression

```
function [x_values_APPROX,y_values_APPROX] = Approx_xy_calculation(handles)
%%------------------------------------------------------------------------%%
% Approximation based algorithm for angle information extraction.
%
% Author:          Oleg Petrak
% Last update:     25.12.2018
%
% INPUT: handles data base.
% OUTPUT: x- and y- signals needed for angle calculation.
%%------------------------------------------------------------------------%%
% memory allocation for all polynomial coefficients
p_vec_BX = zeros(6,361);
p_vec_BY = zeros(6,361);
if (handles.Alg_extension_APPROX == 1)  % operation on data from extraction mask (tracking extension)
   for Angle = 0:1:360
    % use the middle shifted approximation in order to set the coordinate
    % origin to the middle of the array.
    p_vec_BX(:,Angle+1) = field_approximation_middle_shifted(handles.MAGFIELD_BX_operated_AMM(:,:,Angle+1));
    p_vec_BY(:,Angle+1) = field_approximation_middle_shifted(handles.MAGFIELD_BY_operated_AMM(:,:,Angle+1));
    %p_vec_BX(:,Angle+1) = field_approximation(handles.MAGFIELD_BX_operated_AMM(:,:,Angle+1));
    %p_vec_BY(:,Angle+1) = field_approximation(handles.MAGFIELD_BY_operated_AMM(:,:,Angle+1));

    % store the coefficients of interest
    handles.p4_BX_VEC(Angle+1)= p_vec_BX(4,Angle+1);
    handles.p6_BX_VEC(Angle+1)= p_vec_BX(6,Angle+1);
    handles.p4_BY_VEC(Angle+1)= p_vec_BY(4,Angle+1);
    handles.p6_BY_VEC(Angle+1)= p_vec_BY(6,Angle+1);
    end
else  % standard operation (complete array)
   for Angle = 0:1:360
    % use the middle shifted approximation in order to set the coordinate
    % origin to the middle of the array.
    p_vec_BX(:,Angle+1) = field_approximation_middle_shifted(handles.MAGFIELD_BX_operated(:,:,Angle+1));
    p_vec_BY(:,Angle+1) = field_approximation_middle_shifted(handles.MAGFIELD_BY_operated(:,:,Angle+1));
    %p_vec_BX(:,Angle+1) = field_approximation(handles.MAGFIELD_BX_operated(:,:,Angle+1));
    %p_vec_BY(:,Angle+1) = field_approximation(handles.MAGFIELD_BY_operated(:,:,Angle+1));

    % store the coefficients of interest
    handles.p4_BX_VEC(Angle+1)= p_vec_BX(4,Angle+1);
    handles.p6_BX_VEC(Angle+1)= p_vec_BX(6,Angle+1);
    handles.p4_BY_VEC(Angle+1)= p_vec_BY(4,Angle+1);
    handles.p6_BY_VEC(Angle+1)= p_vec_BY(6,Angle+1);
```

```
43      end
44  end
45  % Calculate and output the x- and y-signals:
46  x_values_APPROX = -(handles.p6_BX_VEC + handles.p4_BX_VEC);
47  y_values_APPROX = -(handles.p6_BY_VEC + handles.p4_BY_VEC);
48  end
```

Listing D.5: Approximation based algorithm for angle information extraction

```
1  function [P_VECTOR] = field_approximation_middle_shifted(INPUT_FIELD)
2  %%-------------------------------------------------------------------------%%
3  % Function for estimation of the coefficients of 2D polynomial basis
4  % functions for field approximation. Coordinates origin is in the center of
5  % the sensor array.
6  %
7  % Author:          Oleg Petrak
8  % Last update:     25.12.2018
9  %
10 % INPUT: Data array (Bx or By).
11 % OUTPUT: Coefficients of 2D polynomial.
12 %%-------------------------------------------------------------------------%%
13
14 %% Design matrix A has the form:
15 %[ 1 x1 y1 x1^2 x1*y1 y1^2
16 %  1 x2 y2 x2^2 x2*y2 y2^2
17 %  1 x3 y3 x3^2 x3*y3 y3^2
18 %...]
19 % The center coordinate is in the center of the matrix
20 % Uneven structure
21
22 %% Creation of meshgrids
23 FIELD_WIDTH  = fix(length(INPUT_FIELD) / 2);
24 MESH = [-FIELD_WIDTH:1:FIELD_WIDTH];
25 %% Vectors allocation
26 X_VECTOR = zeros(1,numel(INPUT_FIELD))';
27 Y_VECTOR = zeros(1,numel(INPUT_FIELD))';
28 ONES_VECTOR = ones(1,length(INPUT_FIELD(1,:))*length(INPUT_FIELD(:,1)))';
29 b = zeros(1,numel(INPUT_FIELD))';
30 %% Filling of the vectors x and y with meshgrid coordinates
31 for n = 1:1:length(INPUT_FIELD(:,1))
32     X_VECTOR((1:1:length(INPUT_FIELD(:,1)))+(n-1)*length(INPUT_FIELD(:,1))) = MESH';
33 end
34 for n = 1:1:length(INPUT_FIELD(:,1))
35     Y_VECTOR((1:1:length(INPUT_FIELD(:,1)))+(n-1)*length(INPUT_FIELD(:,1))) = ones(length(INPUT_FIELD(:,1)),1).*MESH(n);
36 end
37 %% Creation of vectors for xy and squared coordinates
38 X_SQ2_VECTOR = (X_VECTOR.*X_VECTOR);
39 Y_SQ2_VECTOR = (Y_VECTOR.*Y_VECTOR);
40 X_Y_VECTOR = (X_VECTOR.*Y_VECTOR);
41 %% Creation of the matrix A
42 A = [ONES_VECTOR,X_VECTOR, Y_VECTOR, X_SQ2_VECTOR, X_Y_VECTOR, Y_SQ2_VECTOR];
43 %% Filling of the vector b with input values corresponding to the coordinates in the matrix A
44 for n=0:1:length(INPUT_FIELD(1,:))-1
45     for m = 0:1:length(INPUT_FIELD(:,1))-1
46         b(m+1 + n*length(INPUT_FIELD(1,:))) = INPUT_FIELD(n+1,m+1);
47     end
48 end
49 %% Solve by using SVD:
50 P_VECTOR = A\b;
51 end
```

Listing D.6: Function for estimation of 2D plynomial basis functions with coordinates origin at the center

Modified approximation algorithm for angle information extraction incl. stray field suppression

```
1  function [x_values_APPROX, y_values_APPROX] = Flat_approx_xy_calculation(handles)
2  %%-------------------------------------------------------------------------%%
3  % Modified approximation  based algorithm for angle information extraction
4  % This algorithms operates always on the information from the extraction mask (tracking).
5  %
6  % Author:          Oleg Petrak
7  % Last update:     25.12.2018
8  %
9  % INPUT: handles data base.
10 % OUTPUT: x- and y- signals needed for angle calculation.
11 %%-------------------------------------------------------------------------%%
12
13 %SCALING_VALUE = max(max(handles.MAGFIELD_BX_operated_AMM(:,:,1)));
14 z0 = handles.ZPos; % z0 is the z-distance to the magnet (in mm units)
```

```
15 %% Determine the coordinate in the array with optimized n exponents
16 switch handles.Array_size_extracted
17     case 15
18         Array_size_coord = 7;
19     case 13
20         Array_size_coord = 6;
21     case 11
22         Array_size_coord = 5;
23     case 9
24         Array_size_coord = 4;
25     case 7
26         Array_size_coord = 3;
27     case 5
28         Array_size_coord = 2;
29     case 3
30         Array_size_coord = 1;
31     otherwise
32         Array_size_coord = 7;
33 end
34 switch handles.Spatial_filtermask
35     case 15
36         Spatial_mask_size_coord = 7;
37     case 13
38         Spatial_mask_size_coord = 6;
39     case 11
40         Spatial_mask_size_coord = 5;
41     case 9
42         Spatial_mask_size_coord = 4;
43     case 7
44         Spatial_mask_size_coord = 3;
45     case 5
46         Spatial_mask_size_coord = 2;
47     case 3
48         Spatial_mask_size_coord = 1;
49     otherwise
50         Spatial_mask_size_coord = 7;
51 end
52 % and use the corresponding value of the n exponent what determined by optimization procedure
53 if (strcmp(handles.MAGNETSHAPE ,'DIPOL'))
54     switch z0
55         case 4
56             %n = 0.0136;
57             n = handles.n_optimized_dipol(1,Array_size_coord,Spatial_mask_size_coord);
58         case 5
59             n = handles.n_optimized_dipol(2,Array_size_coord,Spatial_mask_size_coord);
60         case 6
61             n = handles.n_optimized_dipol(3,Array_size_coord,Spatial_mask_size_coord);
62         case 7
63             n = handles.n_optimized_dipol(4,Array_size_coord,Spatial_mask_size_coord);
64         case 8
65             n = handles.n_optimized_dipol(5,Array_size_coord,Spatial_mask_size_coord);
66         case 9
67             n = handles.n_optimized_dipol(6,Array_size_coord,Spatial_mask_size_coord);
68         case 10
69             n = handles.n_optimized_dipol(7,Array_size_coord,Spatial_mask_size_coord);
70         case 11
71             n = handles.n_optimized_dipol(8,Array_size_coord,Spatial_mask_size_coord);
72         case 12
73             n = handles.n_optimized_dipol(9,Array_size_coord,Spatial_mask_size_coord);
74         otherwise
75             n = handles.n_optimized_dipol(1,Array_size_coord,Spatial_mask_size_coord);
76     end
77 elseif (strcmp(handles.MAGNETSHAPE ,'DISC'))
78     switch z0
79         case 4
80             n = handles.n_optimized_disc(1,Array_size_coord,Spatial_mask_size_coord);
81         case 5
82             n = handles.n_optimized_disc(2,Array_size_coord,Spatial_mask_size_coord);
83         case 6
84             n = handles.n_optimized_disc(3,Array_size_coord,Spatial_mask_size_coord);
85         case 7
86             n = handles.n_optimized_disc(4,Array_size_coord,Spatial_mask_size_coord);
87         case 8
88             n = handles.n_optimized_disc(5,Array_size_coord,Spatial_mask_size_coord);
89         case 9
90             n = handles.n_optimized_disc(6,Array_size_coord,Spatial_mask_size_coord);
91         case 10
92             n = handles.n_optimized_disc(7,Array_size_coord,Spatial_mask_size_coord);
93         case 11
94             n = handles.n_optimized_disc(8,Array_size_coord,Spatial_mask_size_coord);
95         case 12
96             n = handles.n_optimized_disc(9,Array_size_coord,Spatial_mask_size_coord);
97         otherwise
98             n = handles.n_optimized_disc(1,Array_size_coord,Spatial_mask_size_coord);
99     end
100 end
101 %% Prepate the coordinate meshgirds
102 WIDTH  = fix(length(handles.MAGFIELD_BX_operated(:,:,1)) / 2);
```

```
MESH = [-WIDTH:1:WIDTH];
MESH = MESH / WIDTH * handles.Edge_length_of_sensor_array / 2;
STEP_SIZE = MESH(end)-MESH(end-1); % needed for adherance of the physical dimensions
WIDTH2  = fix(length(handles.MAGFIELD_BX_operated_AMM(:,:,1)) / 2);
MESH = [-WIDTH2:1:WIDTH2];
MESH = MESH / WIDTH2 * (STEP_SIZE * fix(length(MESH)/2));
[X,Y] = meshgrid(MESH); % meshgrid is of physical relevance to the z-distance (mm units)
%print coefficient n:
n
%% Solving procedure
    for Angle = 0:1:360
     % create design matrix DM (K.üJnemann code part, adapted):
     f1 = @(x,y) 1./((z0^2 + (x).^2 + (y).^2).^n);
     f2 = @(x,y) (x)./((z0^2 + (x).^2 + (y).^2).^n);
     f3 = @(x,y) (y)./((z0^2 + (x).^2 + (y).^2).^n);
     f4 = @(x,y) (y).^2 ./((z0^2 + (x).^2 + (y).^2).^n);
     f5 = @(x,y) (x).*(y)./((z0^2 + (x).^2 + (y).^2).^n);
     f6 = @(x,y) (x).^2 ./((z0^2 + (x).^2 + (y).^2).^n);
     DM = [f1(X(:), Y(:)), f2(X(:), Y(:)), f3(X(:), Y(:)), ...
         f4(X(:), Y(:)), f5(X(:), Y(:)), f6(X(:), Y(:))]; % (no homogeneous part is included in this design matrix)
 %    DM = [f1(X(:), Y(:)), f2(X(:), Y(:)), f3(X(:), Y(:)), ...
 %         f4(X(:), Y(:)), f5(X(:), Y(:)), f6(X(:), Y(:)), ...
 %        ones(size(X(:)))]; % (in case if additional coefficients for homogeneous part in the design matrix is used. In this case the
      quantization effect increases drastically.)
   Bx = handles.MAGFIELD_BX_operated_AMM(:,:,Angle+1);% / SCALING_VALUE;
     By = handles.MAGFIELD_BY_operated_AMM(:,:,Angle+1);% / SCALING_VALUE;
     %p_vec_BX = linsolve(DM'*DM, DM'*Bx(:));
     %p_vec_BY = linsolve(DM'*DM, DM'*By(:));
     % Solve by using SVD:
     p_vec_BX = DM\Bx(:);
     p_vec_BY = DM\By(:);
     % Save the coefficients progression:
     handles.p4_BX_VEC(Angle+1)= p_vec_BX(4);
     handles.p6_BX_VEC(Angle+1)= p_vec_BX(6);
     handles.p4_BY_VEC(Angle+1)= p_vec_BY(4);
     handles.p6_BY_VEC(Angle+1)= p_vec_BY(6);
     % Calculate field components for visualization if necessary
     %Flat_behaviour_field = (X.^2 + Y.^2 + z0^2 ).^n;
     %APPROXIMATED_FIELD_BX_FLAT = ( p_vec_BX(1) + p_vec_BX(2)*X + p_vec_BX(3)*Y + p_vec_BX(4)*Y.^2 + p_vec_BX(5)*X.*Y  + p_vec_BX(6)*X.^2
        ) ./ Flat_behaviour_field;
     %APPROXIMATED_FIELD_BX_FLAT = ( p_vec_BX(1) + p_vec_BX(2)*X + p_vec_BX(3)*Y + p_vec_BX(4)*Y.^2 + p_vec_BX(5)*X.*Y  + p_vec_BX(6)*X.^2
        ) ./ Flat_behaviour_field + p_vec_BX(7) ;
    end
% show the field components at particular encoder magnet angle:
%figure(); surf(handles.MAGFIELD_BX_operated_AMM(:,:,1))
%figure(); surf(APPROXIMATED_FIELD_BX_FLAT);
%ERROR = ((APPROXIMATED_FIELD_BX_FLAT - handles.MAGFIELD_BX_operated_AMM(:,:,1)) ./ handles.MAGFIELD_BX_operated_AMM(:,:,1) ) * 100
% Calculate and output the x- and y-signals:
x_values_APPROX = -(handles.p6_BX_VEC + handles.p4_BX_VEC);
y_values_APPROX = -(handles.p6_BY_VEC + handles.p4_BY_VEC);
end
```

Listing D.7: Modified approximation based algorithm for angle information extraction

LoG convolution for angle information extraction incl. stray field suppression

```
function [ARRAY_BX_Conv_filt, ARRAY_BY_Conv_filt, x_values_Conv, y_values_Conv] = Convolution_xy_calculation(handles)
%%-------------------------------------------------------------------------%%
% Convolution based algorithm for angle information extraction.
%
% Author:          Oleg Petrak
% Last update:     25.12.2018
%
% INPUT: handles data base.
% OUTPUT: x- and y- signals needed for angle calculation and data arrays
% with extracted local curvatures from Bx and By field components.
%%-------------------------------------------------------------------------%%

% Calculate the reduced mask size (in this setup always 2 elements smaller than the total array size)
ARRAY_SIZE = handles.Array_size_extracted;
if (ARRAY_SIZE == 3)
    MASK_SIZE = ARRAY_SIZE;
else
    MASK_SIZE = ARRAY_SIZE-2;
end
% Create the loG filter mask for the convolution
CONV_LOG_FILTER = Laplacian_of_Gaussian_filter('log',MASK_SIZE,handles.LOG_FILTER_BANDWIDTH_TOTAL);
CONV_LOG_FILTER = CONV_LOG_FILTER / max(max((abs(CONV_LOG_FILTER))));
% Operation on first derivative:
if (handles.DV1_FLAG == 1 || handles.DV2_FLAG == 0)
    %(NOT IMPLEMENTED). Here the global gradient needs to be extracted.
elseif (handles.DV1_FLAG == 0 || handles.DV2_FLAG == 1)
    for Angle = 0:1:360
```

```
28          % Perform the convolution of the Bx and By data arrays with the LoG mask (alternatively imfilter()-function can be used)
29          CURVATURE_FIELD_BX = conv2((handles.MAGFIELD_BX_operated(:,:,Angle+1)),CONV_LOG_FILTER,'same');
30          CURVATURE_FIELD_BY = conv2((handles.MAGFIELD_BY_operated(:,:,Angle+1)),CONV_LOG_FILTER,'same');
31          % cut the area of interest out of the resulting picture (bondaries..)
32          CURVATURE_FIELD_BX = CURVATURE_FIELD_BX(fix(MASK_SIZE/2)+1:1:end-fix(MASK_SIZE/2),fix( MASK_SIZE/2)+1:1:end-fix(MASK_SIZE/2));
33          CURVATURE_FIELD_BY = CURVATURE_FIELD_BY(fix(MASK_SIZE/2)+1:1:end-fix(MASK_SIZE/2),fix( MASK_SIZE/2)+1:1:end-fix(MASK_SIZE/2));
34          % store the extracted curvature images (local curvatures)
35          handles.ARRAY_BX_Conv_filt(:,:,Angle+1) = CURVATURE_FIELD_BX;
36          handles.ARRAY_BY_Conv_filt(:,:,Angle+1) = CURVATURE_FIELD_BY;
37          % Summation of corresponding local curvatures to one x- and y-value
38          handles.x_values_Conv(Angle+1) = sum(sum(CURVATURE_FIELD_BX));
39          handles.y_values_Conv(Angle+1) = sum(sum(CURVATURE_FIELD_BY));
40      end
41  end
42  % Output of the x- and y-signals and the curvature images:
43  x_values_Conv  = -handles.x_values_Conv;
44  y_values_Conv  = -handles.y_values_Conv;
45  ARRAY_BX_Conv_filt = handles.ARRAY_BX_Conv_filt;
46  ARRAY_BY_Conv_filt = handles.ARRAY_BY_Conv_filt;
47  end
```

Listing D.8: LoG convolution algorithm

Single LoG algorithm for angle information extraction incl. stray field suppression

```
1   function [x_values_LOG,y_values_LOG] = LOG_xy_calculation(handles)
2   %%-------------------------------------------------------------------------%%
3   % Single LoG based algorithm for angle information extraction.
4   %
5   % Author:          Oleg Petrak
6   % Last update:     25.12.2018
7   %
8   % INPUT: handles data base.
9   % OUTPUT: x- and y- signals needed for angle calculation.
10  %%-------------------------------------------------------------------------%%
11
12  if (handles.FI_FLAG ) % Create fixed-point objects and performed fixed-point arithmetics:
13      % Create fixed-point LoG filter masks:
14      handles.LOG_FILTER_TRACKING =  fi(handles.LOG_FILTER_TRACKING, handles.myFI.Signedness, handles.myFI.Word_length, handles.myFI.
          Fraction_length);
15      handles.LOG_FILTER_TOTAL =  fi(handles.LOG_FILTER_TOTAL, handles.myFI.Signedness, handles.myFI.Word_length, handles.myFI.
          Fraction_length);
16      % Set control for arithmetic operations:
17      handles.LOG_FILTER_TRACKING.fimath = handles.myFI_math;
18      handles.LOG_FILTER_TOTAL.fimath = handles.myFI_math;
19      if (handles.Alg_extension_LOG == 1)  % operation on data from extraction mask (tracking extension):
20          for Angle = 0:1:360
21              % accumulation procedure as fixed-point not covered completely here...
22              % Store the x- and y- signals:
23              handles.x_values_LOG(Angle+1) = double(sum(sum(handles.LOG_FILTER_TRACKING .* handles.MAGFIELD_BX_operated_AMM(:,:,Angle+1))))
          ;
24              handles.y_values_LOG(Angle+1) = double(sum(sum(handles.LOG_FILTER_TRACKING .* handles.MAGFIELD_BY_operated_AMM(:,:,Angle+1))))
          ;
25          end
26      else % otherwise operate on standard input data:
27          for Angle = 0:1:360
28              % point-wise multp. (MULTIPLICATION OPERATION)
29              BUFFER_BX = handles.LOG_FILTER_TOTAL .* handles.MAGFIELD_BX_operated(:,:,Angle+1);
30              BUFFER_BY = handles.LOG_FILTER_TOTAL .* handles.MAGFIELD_BY_operated(:,:,Angle+1);
31              %rescaling to [-2,2) (part in case if scaling of Bx and By is performed)
32              %scale_factor = max(max((abs([BUFFER_BX BUFFER_BY]))));
33              %scale_factor = 2 / scale_factor;
34              %BUFFER_BX = BUFFER_BX * scale_factor;
35              %BUFFER_BY = BUFFER_BY * scale_factor;
36              % casting back to s1Qn
37              BUFFER_BX = fi(BUFFER_BX, handles.myFI.Signedness, handles.myFI.Word_length, handles.myFI.Fraction_length); % casting back /
          comeback to register width
38              BUFFER_BY = fi(BUFFER_BY, handles.myFI.Signedness, handles.myFI.Word_length, handles.myFI.Fraction_length); % casting back /
          comeback to register width
39              % multiplicaiton accumulator (ACCUMULATION OPERATION)
40              BUFFER_BX_DOUBLE = double(BUFFER_BX);
41              BUFFER_BY_DOUBLE = double(BUFFER_BY);
42              % perform summation of values for x- and y-signal extraction:
43              x_signal  = sum(sum(BUFFER_BX_DOUBLE));
44              y_signal  = sum(sum(BUFFER_BY_DOUBLE));
45              % 2 * 15 * 15 = 450 maximum value of accu: s9Q10
46              % part in case if scaling of Bx and By is performed:
47              %x_signal = fi(x_signal, handles.myFI.Signedness, 20, 10);
48              %y_signal = fi(y_signal, handles.myFI.Signedness, 20, 10);
49              % part in case if NO scaling of Bx and By is performed:
50              x_signal = fi(x_signal, handles.myFI.Signedness, (handles.myFI.Fraction_length + 10), handles.myFI.Fraction_length);
51              y_signal = fi(y_signal, handles.myFI.Signedness, (handles.myFI.Fraction_length + 10), handles.myFI.Fraction_length);
52              %(back)scaling to [-2,2)
```

```
53              scale_factor =  2 / (max(abs([x_signal y_signal])));
54              x_signal = x_signal * scale_factor;
55              y_signal = y_signal * scale_factor;
56             % casting back to s1Q10
57              x_signal = fi(x_signal, handles.Signedness, handles.Word_length, handles.Fraction_length); % casting back / comeback to
        register width
58              y_signal = fi(y_signal, handles.Signedness, handles.Word_length, handles.Fraction_length); % casting back / comeback to
        register width
59             % Store the x- and y- signals:
60              handles.x_values_LOG(Angle+1)  = double((x_signal));
61              handles.y_values_LOG(Angle+1)  = double((y_signal));
62          end
63      end
64  else % otherwise operate normal without fixed-point arithmetics
65      if (handles.Alg_extension_LOG == 1)  % operation on data from extraction mask (tracking extension):
66          for Angle = 0:1:360
67              % perform summation of values for x- and y-signal extraction:
68              handles.x_values_LOG(Angle+1) = sum(sum(handles.LOG_FILTER_TRACKING .* handles.MAGFIELD_BX_operated_AMM(:,:,Angle+1)));
69              handles.y_values_LOG(Angle+1) = sum(sum(handles.LOG_FILTER_TRACKING .* handles.MAGFIELD_BY_operated_AMM(:,:,Angle+1)));
70          end
71      else % otherwise operate on standard input data:
72          for Angle = 0:1:360
73              % perform summation of values for x- and y-signal extraction:
74              handles.x_values_LOG(Angle+1) = sum(sum(handles.LOG_FILTER_TOTAL .* handles.MAGFIELD_BX_operated(:,:,Angle+1)));
75              handles.y_values_LOG(Angle+1) = sum(sum(handles.LOG_FILTER_TOTAL .* handles.MAGFIELD_BY_operated(:,:,Angle+1)));
76          end
77      end
78  end
79  % Output the x- and y-signals:
80  x_values_LOG =  -handles.x_values_LOG;
81  y_values_LOG =  -handles.y_values_LOG;
82  end
```

Listing D.9: Sngle LoG algorithm

D.2.3 Tracking algorithms

```
1   function [AMM_x_values_mean, AMM_y_values_mean, AMM_x_values, AMM_y_values, warning_values] = track_AMM_point_2(handles)
2   %%-------------------------------------------------------------------------%%
3   % Function for application of trackings algoritms. These algortihms
4   % localize the points of absolute maximum field strenght within the sensor
5   % array x-y-plane.
6   %
7   % Author:          Oleg Petrak
8   % Last update:     25.12.2018
9   %
10  % INPUT: handles data base.
11  % OUTPUT: Mean AMM point x- and y-coordinate, x- and y-coordinates of all
12  % localized AMM points over the full rotation of the encoder magnet and
13  % warnings, if the localized AMM points are outside the sensor array plane.
14
15  % SHORTNOTE: This function was used for the performed investigations. In
16  % the evaluation GUI the function track_AMM_point() is used.
17  %%-------------------------------------------------------------------------%%
18
19  %% Allocation of vectors, creation of mesh grids and masks
20  % masks are corresponding spatial areas for gradient filter based algorithms
21  warning_values = zeros(1,361);
22  AMM_x_values = zeros(1,361);
23  AMM_y_values = zeros(1,361);
24  WIDTH  = fix(length(handles.MAGFIELD_BX_operated(:,:,1)) / 2);
25  MASK_SIZE = handles.Spatial_filtermask;
26  HALF_MASK_SIZE = fix(handles.Spatial_filtermask/2); % for warnings determination
27  MESH = [-WIDTH:1:WIDTH];
28  [X,Y] = meshgrid(MESH); % sensor plane coordinate meshgrid
29  %% Possible preprocessing:
30  if (handles.FLAG_TRACKING_BUTTERWORTH_FILT)
31      [handles.MAGFIELD_BX_operated, handles.MAGFIELD_BY_operated] = Butterworth_filt_tracking(handles);
32  end
33  %% Algorithms:
34  if handles.Tracking_method  == 1    % Magnitude Method (for proof)
35      method = 'Prove method: srqt(X^2 + Y ^2)';
36      for Angle = 0:1:360
37          % Method application:
38          COMBINED_FIELD = sqrt(handles.MAGFIELD_BX_operated(:,:,Angle+1).^2 + handles.MAGFIELD_BY_operated(:,:,Angle+1).^2);
39          % Localization of max value:
40          MAX_VALUE = max(max(COMBINED_FIELD));
41          [x, y] = find_value(COMBINED_FIELD,MAX_VALUE);
42          % Calculate AMM points coordinates and set warnings:
43          AMM_x_values(Angle+1) = x -(WIDTH+1);
44          AMM_y_values(Angle+1) = y -(WIDTH+1);
45          if ( AMM_x_values(Angle+1) > ( HALF_MASK_SIZE +1 )  || AMM_y_values(Angle+1) > ( HALF_MASK_SIZE +1 )  ||...
46                  AMM_x_values(Angle+1) < -( HALF_MASK_SIZE +1 )  || AMM_y_values(Angle+1) < -( HALF_MASK_SIZE +1 ) )
```

```
47              warning_values(Angle+1) = 1;
48          end
49      end
50  elseif handles.Tracking_method  == 2   % Absolute Magnitude Method (just for explaination of badness)
51      method = 'Absolute method: |(X)| + |(Y)|';
52      for Angle = 0:1:360
53          % Method application:
54          COMBINED_FIELD = abs(handles.MAGFIELD_BX_operated(:,:,Angle+1)) + abs(handles.MAGFIELD_BY_operated(:,:,Angle+1));
55          % Localization of max value:
56          MAX_VALUE = max(max(COMBINED_FIELD));
57          [x, y] = find_value(COMBINED_FIELD,MAX_VALUE);
58          % Calculate AMM points coordinates and set warnings:
59          AMM_x_values(Angle+1) = x -(WIDTH+1);
60          AMM_y_values(Angle+1) = y -(WIDTH+1);
61          if ( AMM_x_values(Angle+1) > ( HALF_MASK_SIZE +1 )  || AMM_y_values(Angle+1) > ( HALF_MASK_SIZE +1 )  ||...
62                  AMM_x_values(Angle+1) < -( HALF_MASK_SIZE +1 )  || AMM_y_values(Angle+1) < -( HALF_MASK_SIZE +1 ) )
63              warning_values(Angle+1) = 1;
64          end
65      end
66  elseif handles.Tracking_method  == 3    % Approximation Method
67      method = 'Approximation method';
68      for Angle = 0:1:360
69          % Method application:
70          POL_COEFFS = field_approximation_middle_shifted(handles.MAGFIELD_BX_operated(:,:,Angle+1));
71          APPROXIMATED_FIELD_BX = POL_COEFFS(1) + POL_COEFFS(2).*X + POL_COEFFS(3).*Y + POL_COEFFS(4)*X.^2 + POL_COEFFS(5)*X.*Y + POL_COEFFS
        (6).*Y.^2;
72          POL_COEFFS = field_approximation_middle_shifted(handles.MAGFIELD_BY_operated(:,:,Angle+1));
73          APPROXIMATED_FIELD_BY = POL_COEFFS(1) + POL_COEFFS(2).*X + POL_COEFFS(3).*Y + POL_COEFFS(4)*X.^2 + POL_COEFFS(5)*X.*Y + POL_COEFFS
        (6).*Y.^2;
74          COMBINED_FIELD = (APPROXIMATED_FIELD_BY).^2 + (APPROXIMATED_FIELD_BX).^2;
75          % Localization of max value:
76          MAX_VALUE = max(max(COMBINED_FIELD));
77          [x, y] = find_value(COMBINED_FIELD,MAX_VALUE);
78          % Calculate AMM points coordinates and set warnings:
79          AMM_x_values(Angle+1) = x -(WIDTH+1);
80          AMM_y_values(Angle+1) = y -(WIDTH+1);
81          if ( AMM_x_values(Angle+1) > ( HALF_MASK_SIZE +1 )  || AMM_y_values(Angle+1) > ( HALF_MASK_SIZE +1 )  ||...
82                  AMM_x_values(Angle+1) < -( HALF_MASK_SIZE +1 )  || AMM_y_values(Angle+1) < -( HALF_MASK_SIZE +1 ) )
83              warning_values(Angle+1) = 1;
84          end
85      end
86  elseif handles.Tracking_method  == 4   % Approximation + analytical Method
87      method = 'Approximation + analytical Method';
88      for Angle = 0:1:360
89          % Method application:
90          COMBINED_FIELD = (handles.MAGFIELD_BX_operated(:,:,Angle+1)).^2 + (handles.MAGFIELD_BY_operated(:,:,Angle+1)).^2;
91          POL_COEFFS = field_approximation_middle_shifted_without_xy(COMBINED_FIELD);
92          %COMBINED_FIELD = POL_COEFFS(1) + POL_COEFFS(2).*X + POL_COEFFS(3).*Y + POL_COEFFS(4)*X.^2 + POL_COEFFS(5).*Y.^2;
93          % Localize the max value analytically, calculate AMM points coordinates and set warnings:
94          AMM_x_values(Angle+1) = round( - POL_COEFFS(2) / (2 * POL_COEFFS(4)) );
95          AMM_y_values(Angle+1) = round( - POL_COEFFS(3) / (2 * POL_COEFFS(5)) );
96          if ( AMM_x_values(Angle+1) > ( HALF_MASK_SIZE +1 )  || AMM_y_values(Angle+1) > ( HALF_MASK_SIZE +1 )  || ...
97                  AMM_x_values(Angle+1) < -( HALF_MASK_SIZE +1 )  || AMM_y_values(Angle+1) < -( HALF_MASK_SIZE +1 ) )
98              warning_values(Angle+1) = 1;
99          end
100     end
101 elseif handles.Tracking_method  == 5 % Derivative of Gaussians Method
102     method = 'Derivative of Gaussians Method';
103     % Filter creation:
104     DOG_FILTER_X = Derivatite_of_Gaussian_filter(MASK_SIZE,handles.GAUSSIAN_FILTER_BANDWIDTH,0);
105     DOG_FILTER_X = DOG_FILTER_X/max(max(DOG_FILTER_X));
106     DOG_FILTER_Y = Derivatite_of_Gaussian_filter(MASK_SIZE,handles.GAUSSIAN_FILTER_BANDWIDTH,1);
107     DOG_FILTER_Y = DOG_FILTER_Y/max(max(DOG_FILTER_Y));
108     for Angle = 0:1:360
109         % Method application:
110         COMBINED_FIELD = (handles.MAGFIELD_BX_operated(:,:,Angle+1).^2 + handles.MAGFIELD_BY_operated(:,:,Angle+1).^2);
111         GRADIENT_FIELD_X = conv2((COMBINED_FIELD),DOG_FILTER_X,'same');
112         GRADIENT_FIELD_X = GRADIENT_FIELD_X(fix(MASK_SIZE/2)+1:1:end-fix(MASK_SIZE/2),fix( MASK_SIZE/2)+1:1:end-fix(MASK_SIZE/2));
113         GRADIENT_FIELD_Y = conv2((COMBINED_FIELD),DOG_FILTER_Y,'same');
114         GRADIENT_FIELD_Y = GRADIENT_FIELD_Y(fix(MASK_SIZE/2)+1:1:end-fix(MASK_SIZE/2),fix( MASK_SIZE/2)+1:1:end-fix(MASK_SIZE/2));
115         GRADIENT_FIELD = (GRADIENT_FIELD_X.^2 + GRADIENT_FIELD_Y.^2);
116         % Localization of max value:
117         MIN_VALUE = min(min((GRADIENT_FIELD)));
118         [x, y] = find_value((GRADIENT_FIELD),MIN_VALUE);
119         x_values(Angle+1) = x;
120         y_values(Angle+1) = y;
121         % Calculate AMM points coordinates and set warnings:
122         AMM_x_values(Angle+1) = x - round(length(GRADIENT_FIELD)/2)  ;
123         AMM_y_values(Angle+1) = y - round(length(GRADIENT_FIELD)/2)  ;
124         if ( AMM_x_values(Angle+1) > ( HALF_MASK_SIZE +1 )  || AMM_y_values(Angle+1) > ( HALF_MASK_SIZE +1 )  ||...
125                 AMM_x_values(Angle+1) < -( HALF_MASK_SIZE +1 )  || AMM_y_values(Angle+1) < -( HALF_MASK_SIZE +1 ) )
126             warning_values(Angle+1) = 1;
127         end
128     end
129 elseif handles.Tracking_method  == 6 %  Laplacian of Gaussian Method
130     % Filter creation:
131     method = 'Laplacian of Gaussians Method';
132     LOG_FILTER_TRACKING = Laplacian_of_Gaussian_filter('log',MASK_SIZE,handles.LOG_FILTER_BANDWIDTH_TRACKING);
```

```
133     LOG_FILTER_TRACKING = LOG_FILTER_TRACKING/max(max(LOG_FILTER_TRACKING));
134     for Angle = 0:1:360
135         % Method application:
136         COMBINED_FIELD = (handles.MAGFIELD_BX_operated(:,:,Angle+1).^2 + handles.MAGFIELD_BY_operated(:,:,Angle+1).^2);
137         CURVATURE_FIELD = conv2((COMBINED_FIELD),LOG_FILTER_TRACKING,'same');
138         CURVATURE_FIELD = CURVATURE_FIELD(fix(MASK_SIZE/2)+1:1:end-fix(MASK_SIZE/2),fix(MASK_SIZE/2)+1:1:end-fix(MASK_SIZE/2));
139         % Localization of max value:
140         MAX_VALUE = max(max((-CURVATURE_FIELD)));
141         [x, y] = find_value((-CURVATURE_FIELD),MAX_VALUE);
142         x_values(Angle+1) = x;
143         y_values(Angle+1) = y;
144         % Calculate AMM points coordinates and set warnings:
145         AMM_x_values(Angle+1) = x - round(length(CURVATURE_FIELD)/2) ;
146         AMM_y_values(Angle+1) = y - round(length(CURVATURE_FIELD)/2) ;
147         if ( AMM_x_values(Angle+1) > ( HALF_MASK_SIZE +1 ) || AMM_y_values(Angle+1) > ( HALF_MASK_SIZE +1 ) ||...
148                 AMM_x_values(Angle+1) < -( HALF_MASK_SIZE +1 ) || AMM_y_values(Angle+1) < -( HALF_MASK_SIZE +1 ) )
149             warning_values(Angle+1) = 1;
150         end
151     end
152 elseif handles.Tracking_method == 7 %  Butterworth_filt_method and localization of the point
153     method = 'Butterworth filtering and max(X^2 + Y ^2)';
154     [handles.MAGFIELD_BX_operated, handles.MAGFIELD_BY_operated] = Butterworth_filt_tracking(handles);
155     for Angle = 0:1:360
156         % Method application:
157         COMBINED_FIELD = (handles.MAGFIELD_BX_operated(:,:,Angle+1).^2 + handles.MAGFIELD_BY_operated(:,:,Angle+1).^2);
158         COMBINED_FIELD = COMBINED_FIELD(fix(MASK_SIZE/2)+1:1:end-fix(MASK_SIZE/2),fix(MASK_SIZE/2)+1:1:end-fix(MASK_SIZE/2));
159         % Localization of max value:
160         MAX_VALUE = max(max(COMBINED_FIELD));
161         [x, y] = find_value(COMBINED_FIELD,MAX_VALUE);
162         % Calculate AMM points coordinates and set warnings:
163         AMM_x_values(Angle+1) = x - round(length(COMBINED_FIELD)/2) ;
164         AMM_y_values(Angle+1) = y - round(length(COMBINED_FIELD)/2) ;
165         if ( AMM_x_values(Angle+1) > ( HALF_MASK_SIZE +1 ) || AMM_y_values(Angle+1) > ( HALF_MASK_SIZE +1 ) ||...
166                 AMM_x_values(Angle+1) < -( HALF_MASK_SIZE +1 ) || AMM_y_values(Angle+1) < -( HALF_MASK_SIZE +1 ) )
167             warning_values(Angle+1) = 1;
168         end
169     end
170 elseif handles.Tracking_method == 8
171    method = 'Butterworth filtering zero padded and max(X^2 + Y ^2)';
172     [handles.MAGFIELD_BX_operated, handles.MAGFIELD_BY_operated] = Butterworth_filt_tracking_zero_padded(handles,1);
173     for Angle = 0:1:360
174         % Method application:
175         COMBINED_FIELD = (handles.MAGFIELD_BX_operated(:,:,Angle+1).^2 + handles.MAGFIELD_BY_operated(:,:,Angle+1).^2);
176         COMBINED_FIELD = COMBINED_FIELD(fix(MASK_SIZE/2)+1:1:end-fix(MASK_SIZE/2),fix(MASK_SIZE/2)+1:1:end-fix(MASK_SIZE/2));
177         % Localization of max value:
178         MAX_VALUE = max(max(COMBINED_FIELD));
179         [x, y] = find_value(COMBINED_FIELD,MAX_VALUE);
180         % Calculate AMM points coordinates and set warnings:
181         AMM_x_values(Angle+1) = x - round(length(COMBINED_FIELD)/2) ;
182         AMM_y_values(Angle+1) = y - round(length(COMBINED_FIELD)/2) ;
183         if ( AMM_x_values(Angle+1) > ( HALF_MASK_SIZE +1 ) || AMM_y_values(Angle+1) > ( HALF_MASK_SIZE +1 ) ||...
184                 AMM_x_values(Angle+1) < -( HALF_MASK_SIZE +1 ) || AMM_y_values(Angle+1) < -( HALF_MASK_SIZE +1 ) )
185             warning_values(Angle+1) = 1;
186         end
187     end
188 elseif handles.Tracking_method == 9
189     method = 'Prewitt 5x5';
190     MASK_SIZE = 5;
191     GAUSSIAN_FILTER_BANDWIDTH = 1000000000000000; % filter masks results to a gradient plane => Prewitt
192     DOG_FILTER_X = Derivatite_of_Gaussian_filter(MASK_SIZE,GAUSSIAN_FILTER_BANDWIDTH,0);
193     DOG_FILTER_X = DOG_FILTER_X/max(max(DOG_FILTER_X));
194     DOG_FILTER_Y = Derivatite_of_Gaussian_filter(MASK_SIZE,GAUSSIAN_FILTER_BANDWIDTH,1);
195     DOG_FILTER_Y = DOG_FILTER_Y/max(max(DOG_FILTER_Y));
196     for Angle = 0:1:360
197         % Method application:
198         COMBINED_FIELD = (handles.MAGFIELD_BX_operated(:,:,Angle+1).^2 + handles.MAGFIELD_BY_operated(:,:,Angle+1).^2);
199         GRADIENT_FIELD_X = conv2((COMBINED_FIELD),DOG_FILTER_X,'same');
200         GRADIENT_FIELD_X = GRADIENT_FIELD_X(fix(MASK_SIZE/2)+1:1:end-fix(MASK_SIZE/2),fix(MASK_SIZE/2)+1:1:end-fix(MASK_SIZE/2));
201         GRADIENT_FIELD_Y = conv2((COMBINED_FIELD),DOG_FILTER_Y,'same');
202         GRADIENT_FIELD_Y = GRADIENT_FIELD_Y(fix(MASK_SIZE/2)+1:1:end-fix(MASK_SIZE/2),fix(MASK_SIZE/2)+1:1:end-fix(MASK_SIZE/2));
203         GRADIENT_FIELD = (GRADIENT_FIELD_X.^2 + GRADIENT_FIELD_Y.^2);
204         % Localization of max value:
205         MIN_VALUE = min(min((GRADIENT_FIELD)));
206         [x, y] = find_value((GRADIENT_FIELD),MIN_VALUE);
207         x_values(Angle+1) = x;
208         y_values(Angle+1) = y;
209         % Calculate AMM points coordinates and set warnings:
210         AMM_x_values(Angle+1) = x - round(length(GRADIENT_FIELD)/2) ;
211         AMM_y_values(Angle+1) = y - round(length(GRADIENT_FIELD)/2) ;
212         if ( AMM_x_values(Angle+1) > ( HALF_MASK_SIZE +1 ) || AMM_y_values(Angle+1) > ( HALF_MASK_SIZE +1 ) ||...
213                 AMM_x_values(Angle+1) < -( HALF_MASK_SIZE +1 ) || AMM_y_values(Angle+1) < -( HALF_MASK_SIZE +1 ) )
214             warning_values(Angle+1) = 1;
215         end
216     end
217 end
218 %% Calculation of the coordinates of the mean AMM point:
219 %make sure that all AMM points are within the array, and then perform
220 %the calculation of the mean AMM:
```

```
for Angle2 = 0:1:360
  if ( AMM_x_values(Angle2+1) > (WIDTH))
    AMM_x_values(Angle2+1) =  (WIDTH);
  end
  if ( AMM_x_values(Angle2+1) < -(WIDTH))
    AMM_x_values(Angle2+1) = - (WIDTH);
  end
  if ( AMM_y_values(Angle2+1) > (WIDTH))
    AMM_y_values(Angle2+1) =  (WIDTH);
  end
  if ( AMM_y_values(Angle2+1) < -(WIDTH))
    AMM_y_values(Angle2+1) = - (WIDTH);
  end
end
%Output of the mean AMM coordinate x and y:
AMM_x_values_mean = round(mean(AMM_x_values(1:end-1)));
AMM_y_values_mean = round(mean(AMM_y_values(1:end-1)));
end
```

Listing D.10: Sngle LoG algorithm

D.2.4 Angular error correction

Linear angular error correction algorithm based on homography

```
function [x_values_hom_corrected, y_values_hom_corrected, H] = homography_correction(INPUT_VALUES_BX,INPUT_VALUES_BY,X_VECTOR_FAILUREFREE,
    Y_VECTOR_FAILUREFREE,EXTRACTION_VECTOR,amount_of_h_coeefs)
%%-------------------------------------------------------------------------%%
% Function for application of linear angular error correction algorithm
% based on homography (projective transformation).
%
% Author:          Oleg Petrak
% Last update:     25.12.2018
%
% INPUT: Amount of the H matrix coefficients to estimate, ideal x- and y- vectors
% (perfect sine and cosine waves), distorted x- and y-vectors (input signals to correct)
% and the extraction vector for control of the amount of data points for
% the estimation.
% OUTPUT: Corrected x- and y-signals.
%%-------------------------------------------------------------------------%%

%% Homography estimation (projective transformation):
H = homography(INPUT_VALUES_BX(EXTRACTION_VECTOR),INPUT_VALUES_BY(EXTRACTION_VECTOR),X_VECTOR_FAILUREFREE(EXTRACTION_VECTOR),
    Y_VECTOR_FAILUREFREE(EXTRACTION_VECTOR),amount_of_h_coeefs);
%% Correction of input values x and y:
for Angle = 1:1:361
    PROJECTED_POINT = H * [INPUT_VALUES_BX(Angle) INPUT_VALUES_BY(Angle) 1]';
    handles.x_values_hom_corrected(Angle) = PROJECTED_POINT(1)/PROJECTED_POINT(3);
    handles.y_values_hom_corrected(Angle) = PROJECTED_POINT(2)/PROJECTED_POINT(3);
end
%% Output of corrected x- and y-signals:
    x_values_hom_corrected = handles.x_values_hom_corrected;
    y_values_hom_corrected = handles.y_values_hom_corrected;
end
```

Listing D.11: Application function for linear angular error correction algorithm based on homography

```
function [H] = homography(X_VECTOR, Y_VECTOR, X_VECTOR_PROJECTIVE, Y_VECTOR_PROJECTIVE, AMOUNT_h_COEFFS)
%%-------------------------------------------------------------------------%%
% Function for estimation of the homography matrix needed for the linear
% angular error correction algorithm.
%
% Author:          Oleg Petrak
% Last update:     25.12.2018
%
% INPUT: Amount of the H matrix coefficients to estimate, ideal x- and y- vectors
% (perfect sine and cosine waves) and distorted x- and y-vectors (input signals to correct).
% OUTPUT: Homography matrix H.
%%-------------------------------------------------------------------------%%

%%
% A * h = b
% h = A\b (STANDARD: SDV solving of Least Squares Problem)

% h includes the coefficient of the matrix H
% A must include the formulated expression with the original and projected
% points such as:

%                    ----FOR DOF 6----
```

```
%A = [-x1  -y1  1  0    0    0  ;
%      0    0    0 -x1   -y1  1  ;
%     -x2  -y2  1   0    0    0  ;
%      0    0    0 -x2   -y2  1 ];

%                    ----FOR DOF 8 (8 coefficients)----
%A = [   P1X P1Y 1 0 0 0    -P1X*PP1X  -P1Y*PP1X;
%        0 0 0 P1X P1Y 1    -P1X*PP1Y  -P1Y*PP1Y;
%        P2X P2Y 1 0 0 0    -P2X*PP2X  -P2Y*PP2X;
%        0 0 0 P2X P2Y 1    -P2X*PP2Y  -P2Y*PP2Y];

%                    ----FOR DOF 8 (9 coefficients)----
%A = [-P1X  -P1Y  -1  0    0    0   P1X*PP1X   P1Y*PP1X   PP1X;
%      0    0    0 -P1X   -P1Y  -1   P1X*PP1Y   P1Y*PP1Y   PP1Y;
%     -P2X  -P2Y  -1  0    0    0   P2X*PP2X   P2Y*PP2X   PP2X;
%      0    0    0 -P2X   -P2Y  -1   P2X*PP2Y   P2Y*PP2Y   PP2Y];

% P1X - original point
% PP1X - projected point
% b must include the pairs of the projected points

%% INITIALISAZIONS ARE HERE
h = zeros(1,9);
h(7) = 0;
h(8) = 0;
h(9) = 1;
LENGTH_VECTOR = length(X_VECTOR); % Y_VECTOR equivalent

%if (AMOUNT_h_COEFFS == 4)
 %   A_row_even = zeros(LENGTH_VECTOR, AMOUNT_h_COEFFS+2);
 %   A_row_uneven = zeros(LENGTH_VECTOR, AMOUNT_h_COEFFS+2);
 %   A = zeros(LENGTH_VECTOR*2, AMOUNT_h_COEFFS+2);
%else
    A_row_even = zeros(LENGTH_VECTOR, AMOUNT_h_COEFFS);
    A_row_uneven = zeros(LENGTH_VECTOR, AMOUNT_h_COEFFS);
    A = zeros(LENGTH_VECTOR*2, AMOUNT_h_COEFFS);
%end

%% CALCULATIONS OF THE b VECTOR IS HERE:
%b = [PP1X PP1Y PP2X PP2Y PP3X PP3Y PP4X PP4Y PP5X PP5Y PP6X PP6Y...]';
b = zeros(1,LENGTH_VECTOR*2);
m = 1;
for n=1:2:LENGTH_VECTOR*2
     b(n) = X_VECTOR_PROJECTIVE(m);
     b(n+1) = Y_VECTOR_PROJECTIVE(m);
     m = m+1;
end
b = b';
m = 1;
%% CALCULATIONS OF THE MATRIXES A AND HOMOGRAPHY COEFFICIENTS:
 if(AMOUNT_h_COEFFS == 6)    % HERE IS ONLY AN AFFIN PROJECTION INCLUDED: DOF = 6
    % P1X P1Y 1 0 0 0    ;    = uneven structure
    % 0 0 0 P1X P1Y 1    ;    = even structure
    for n = 1:1:LENGTH_VECTOR
        A_row_uneven(n,:) = [X_VECTOR(n) Y_VECTOR(n)  1 0 0 0 ];
        A_row_even(n,:) =   [0 0 0 X_VECTOR(n) Y_VECTOR(n) 1  ];
    end
    for n=1:2:LENGTH_VECTOR*2
        A(n,:) = A_row_uneven(m,:);
        A(n+1,:) = A_row_even(m,:);
        m = m+1;
    end
    % HOMOGRAPHY COEFFICIENTS CALCULATION:
    h(1:6) = A\b;
    % alternatively: QR decomposition (works for fixed-point arithmetic)
%     [Q,R,V] = householder(A'*A);
%     %[Q,R,V] = qr(A'*A);
%     b_ = A'*b;
%     z  = Q'*b_;
%     %Rx = z with back subsitution: R\z;
%     x = backward_substitution(R,z);
%     h(1:6) = x;
 else
     if(AMOUNT_h_COEFFS == 8) % HERE ARE 8 COEFFICIENT INCLUDED: DOF = 8
        % P1X P1Y 1 0 0 0    -P1X*PP1X  -P1Y*PP1X;    = uneven structure
        % 0 0 0 P1X P1Y 1    -P1X*PP1Y  -P1Y*PP1Y;    = even structure
         for n = 1:1:LENGTH_VECTOR
             A_row_uneven(n,:) = [X_VECTOR(n) Y_VECTOR(n)  1 0 0 0 -X_VECTOR(n)*X_VECTOR_PROJECTIVE(n) -Y_VECTOR(n)*X_VECTOR_PROJECTIVE(n)
         ];
             A_row_even(n,:) = [0 0 0 X_VECTOR(n) Y_VECTOR(n) 1  -X_VECTOR(n)*Y_VECTOR_PROJECTIVE(n) -Y_VECTOR(n)*Y_VECTOR_PROJECTIVE(n)];
         end
         for n=1:2:LENGTH_VECTOR*2
             A(n,:) = A_row_uneven(m,:);
             A(n+1,:) = A_row_even(m,:);
             m = m+1;
         end
         % HOMOGRAPHY COEFFICIENTS CALCULATION:
         h(1:8) = A\b;
```

```
%          [Q,R,V] = householder(A'*A);
%          %[Q,R,V] = qr(A'*A);
%          b_ = A'*b;
%          z  = Q'*b_;
%          x = backward_substitution(R,z);
%          h(1:8) = x;
      else    % HERE ARE 9 COEFFICIENT INCLUDED: DOF = 9: ONLY SVD METHOD POSSIBLE
         % MATRIX NEEDED FOR SVD IS HERE:
         %     -P1X  -P1Y  -1   0     0     0    P1X*PP1X    P1Y*PP1X    PP1X = uneven structure
         %      0     0     0  -P1X   -P1Y  -1    P1X*PP1Y    P1Y*PP1Y    PP1Y = even structure
          for n = 1:1:LENGTH_VECTOR
              A_row_uneven(n,:) = [-X_VECTOR(n) -Y_VECTOR(n)  -1 0 0 0 X_VECTOR(n)*X_VECTOR_PROJECTIVE(n) Y_VECTOR(n)*X_VECTOR_PROJECTIVE(n)
         X_VECTOR_PROJECTIVE(n)];
              A_row_even(n,:) =   [0 0 0 -X_VECTOR(n) -Y_VECTOR(n) -1  X_VECTOR(n)*Y_VECTOR_PROJECTIVE(n) Y_VECTOR(n)*Y_VECTOR_PROJECTIVE(n)
         Y_VECTOR_PROJECTIVE(n)];
          end
          for n=1:2:LENGTH_VECTOR*2
              A(n,:) = A_row_uneven(m,:);
              A(n+1,:) = A_row_even(m,:);
              m = m+1;
          end
          % HOMOGRAPHY COEFFICIENTS CALCULATION:
          [U,S,V] = svd(A);
          h = V(:,end);
      end
  end
%% H matrix output:
  H = [h(1) h(2) h(3);
       h(4) h(5) h(6);
       h(7) h(8) h(9)];
end
```

Listing D.12: Function for homography matrix estimation

Nonlinear angular error correction algorithm based on geometric distortion compensation

```
function [NON_LIN_CORRECTED_X_VALUES,NON_LIN_CORRECTED_Y_VALUES] = nonlinear_correction_geometric_distortion(INPUT_X_VALUES,INPUT_Y_VALUES
    ,X_VECTOR_FAILUREFREE,Y_VECTOR_FAILUREFREE,EXTRACTION_POINTS_AMOUNT_FOR_EACH_POLYNOM,DEGREE)
%%-------------------------------------------------------------------------%%
% Function for application of nonlinear angular error correction algorithm
% based on geometric distortion cancelation.
%
% Author:          Oleg Petrak
% Last update:     25.12.2018
%
% INPUT: Ideal vectors  (sine and cosine waves), distorted x- and y-vectors
% (input signals to correct) and the amount of data points for the estimation.
% OUTPUT: Corrected x- and y-signals.
%%-------------------------------------------------------------------------%%

%% SPLITTING THE  IDEAL COMPONENTS INTO 4 SECTIONS:
%%
X_VECTOR_FAILUREFREE_POSITIVE  = [X_VECTOR_FAILUREFREE(fix(length(X_VECTOR_FAILUREFREE)/4*3)+1:end-1) X_VECTOR_FAILUREFREE(1:fix(length(
    X_VECTOR_FAILUREFREE)/4)+1)];
% figure();
% plot(0:1:180,X_VECTOR_FAILUREFREE_POSITIVE);
% grid minor
%X POS:  271 to 359 + 1 to 91
%%
X_VECTOR_FAILUREFREE_NEGATIVE = [X_VECTOR_FAILUREFREE(fix(length(X_VECTOR_FAILUREFREE)/4)+1:(fix(length(X_VECTOR_FAILUREFREE)/4*3)+1))];
% figure();
% plot(0:1:180,X_VECTOR_FAILUREFREE_NEGATIVE);
% grid minor
%X NEG:  91 to 271
%%
Y_VECTOR_FAILUREFREE_POSITIVE  = [Y_VECTOR_FAILUREFREE(1:fix(length(X_VECTOR_FAILUREFREE)/2)+1)];
% figure();
% plot(0:1:180,Y_VECTOR_FAILUREFREE_POSITIVE);
% grid minor
%Y POS:  1 to 181
%%
Y_VECTOR_FAILUREFREE_NEGATIVE = [Y_VECTOR_FAILUREFREE(fix(length(Y_VECTOR_FAILUREFREE)/2)+1:(length(Y_VECTOR_FAILUREFREE)))];
% figure();
% plot(0:1:180,Y_VECTOR_FAILUREFREE_NEGATIVE);
% grid minor
%Y NEG:  181 to 361
%% SPLITTING THE INPUT COMPONENTS INTO 4 SECTIONS:
%
INPUT_X_VALUES_POSITIVE = [INPUT_X_VALUES(fix(length(INPUT_X_VALUES)/4*3)+1:end-1) INPUT_X_VALUES(1:fix(length(INPUT_X_VALUES)/4+1))];
INPUT_X_VALUES_POSITIVE_Y_COMP = [INPUT_Y_VALUES(fix(length(INPUT_Y_VALUES)/4*3)+1:end-1) INPUT_Y_VALUES(1:fix(length(INPUT_Y_VALUES)/4+1)
    )];
```

```
% figure();
% plot(INPUT_X_VALUES_POSITIVE,INPUT_X_VALUES_POSITIVE_Y_COMP);
% grid minor
%X POS:  271 to 359 + 1 to 91   °(270 to (360 + 90 )°
INPUT_X_VALUES_NEGATIVE = [INPUT_X_VALUES(fix(length(INPUT_X_VALUES)/4)+1:(fix(length(INPUT_X_VALUES)/4*3)+1))];
INPUT_X_VALUES_NEGATIVE_Y_COMP = [INPUT_Y_VALUES(fix(length(INPUT_Y_VALUES)/4)+1:(fix(length(INPUT_Y_VALUES)/4*3)+1))];
% figure();
% plot(INPUT_X_VALUES_NEGATIVE,INPUT_X_VALUES_NEGATIVE_Y_COMP);
% grid minor
%X NEG:  91 to 271 °(90 to °270)
INPUT_Y_VALUES_POSITIVE = [INPUT_Y_VALUES(1:fix(length(INPUT_Y_VALUES)/2)+1)];
INPUT_Y_VALUES_POSITIVE_X_COMP = [INPUT_X_VALUES(1:fix(length(INPUT_X_VALUES)/2)+1)];
% figure();
% plot(INPUT_Y_VALUES_POSITIVE_X_COMP,INPUT_Y_VALUES_POSITIVE);
% grid minor
%Y POS:  1 to 181 °(0 to °180)
INPUT_Y_VALUES_NEGATIVE = [INPUT_Y_VALUES(fix(length(INPUT_Y_VALUES)/2)+1:(length(INPUT_Y_VALUES)))];
INPUT_Y_VALUES_NEGATIVE_X_COMP =  [INPUT_X_VALUES(fix(length(INPUT_X_VALUES)/2)+1:(length(INPUT_X_VALUES)))];
% figure();
% plot(INPUT_Y_VALUES_NEGATIVE_X_COMP,INPUT_Y_VALUES_NEGATIVE);
% grid minor
%Y POS:  181 to 361 °(180 to °360)
%% EXTRACTION OF THE VALUES FROM THE INPUT AND IDEAL VECTORS:
EXTRACT_VECTOR = round(linspace(1,181,EXTRACTION_POINTS_AMOUNT_FOR_EACH_POLYNOM));

X_VECTOR_FAILUREFREE_POSITIVE_ESTIMATION = X_VECTOR_FAILUREFREE_POSITIVE(EXTRACT_VECTOR);
X_VECTOR_FAILUREFREE_NEGATIVE_ESTIMATION = X_VECTOR_FAILUREFREE_NEGATIVE(EXTRACT_VECTOR);
Y_VECTOR_FAILUREFREE_POSITIVE_ESTIMATION = Y_VECTOR_FAILUREFREE_POSITIVE(EXTRACT_VECTOR);
Y_VECTOR_FAILUREFREE_NEGATIVE_ESTIMATION = Y_VECTOR_FAILUREFREE_NEGATIVE(EXTRACT_VECTOR);

INPUT_X_VALUES_POSITIVE_ESTIMATION = INPUT_X_VALUES_POSITIVE(EXTRACT_VECTOR);
INPUT_X_VALUES_POSITIVE_Y_COMP_ESTIMATION = INPUT_X_VALUES_POSITIVE_Y_COMP(EXTRACT_VECTOR);
INPUT_X_VALUES_NEGATIVE_ESTIMATION = INPUT_X_VALUES_NEGATIVE(EXTRACT_VECTOR);
INPUT_X_VALUES_NEGATIVE_Y_COMP_ESTIMATION = INPUT_X_VALUES_NEGATIVE_Y_COMP(EXTRACT_VECTOR);
INPUT_Y_VALUES_POSITIVE_ESTIMATION = INPUT_Y_VALUES_POSITIVE(EXTRACT_VECTOR);
INPUT_Y_VALUES_POSITIVE_X_COMP_ESTIMATION = INPUT_Y_VALUES_POSITIVE_X_COMP(EXTRACT_VECTOR);
INPUT_Y_VALUES_NEGATIVE_ESTIMATION = INPUT_Y_VALUES_NEGATIVE(EXTRACT_VECTOR);
INPUT_Y_VALUES_NEGATIVE_X_COMP_ESTIMATION = INPUT_Y_VALUES_NEGATIVE_X_COMP(EXTRACT_VECTOR);
%% ESTIMATION STEP:

if (DEGREE == 2)
    POL_COEFFS_X_POSITIVE = polynomial_approximation_2D_2nd_ord(INPUT_X_VALUES_POSITIVE_ESTIMATION,
      INPUT_X_VALUES_POSITIVE_Y_COMP_ESTIMATION,X_VECTOR_FAILUREFREE_POSITIVE_ESTIMATION);
    POL_COEFFS_X_NEGATIVE = polynomial_approximation_2D_2nd_ord(INPUT_X_VALUES_NEGATIVE_ESTIMATION,
      INPUT_X_VALUES_NEGATIVE_Y_COMP_ESTIMATION,X_VECTOR_FAILUREFREE_NEGATIVE_ESTIMATION);
    POL_COEFFS_Y_POSITIVE = polynomial_approximation_2D_2nd_ord(INPUT_Y_VALUES_POSITIVE_X_COMP_ESTIMATION,
      INPUT_Y_VALUES_POSITIVE_ESTIMATION,Y_VECTOR_FAILUREFREE_POSITIVE_ESTIMATION);
    POL_COEFFS_Y_NEGATIVE = polynomial_approximation_2D_2nd_ord(INPUT_Y_VALUES_NEGATIVE_X_COMP_ESTIMATION,
      INPUT_Y_VALUES_NEGATIVE_ESTIMATION,Y_VECTOR_FAILUREFREE_NEGATIVE_ESTIMATION);
elseif (DEGREE == 4)
    POL_COEFFS_X_POSITIVE = polynomial_approximation_2D_4th_ord(INPUT_X_VALUES_POSITIVE_ESTIMATION,
      INPUT_X_VALUES_POSITIVE_Y_COMP_ESTIMATION,X_VECTOR_FAILUREFREE_POSITIVE_ESTIMATION);
    POL_COEFFS_X_NEGATIVE = polynomial_approximation_2D_4th_ord(INPUT_X_VALUES_NEGATIVE_ESTIMATION,
      INPUT_X_VALUES_NEGATIVE_Y_COMP_ESTIMATION,X_VECTOR_FAILUREFREE_NEGATIVE_ESTIMATION);
    POL_COEFFS_Y_POSITIVE = polynomial_approximation_2D_4th_ord(INPUT_Y_VALUES_POSITIVE_X_COMP_ESTIMATION,
      INPUT_Y_VALUES_POSITIVE_ESTIMATION,Y_VECTOR_FAILUREFREE_POSITIVE_ESTIMATION);
    POL_COEFFS_Y_NEGATIVE = polynomial_approximation_2D_4th_ord(INPUT_Y_VALUES_NEGATIVE_X_COMP_ESTIMATION,
      INPUT_Y_VALUES_NEGATIVE_ESTIMATION,Y_VECTOR_FAILUREFREE_NEGATIVE_ESTIMATION);
elseif (DEGREE == 6)
    POL_COEFFS_X_POSITIVE = polynomial_approximation_2D_6th_ord(INPUT_X_VALUES_POSITIVE_ESTIMATION,
      INPUT_X_VALUES_POSITIVE_Y_COMP_ESTIMATION,X_VECTOR_FAILUREFREE_POSITIVE_ESTIMATION);
    POL_COEFFS_X_NEGATIVE = polynomial_approximation_2D_6th_ord(INPUT_X_VALUES_NEGATIVE_ESTIMATION,
      INPUT_X_VALUES_NEGATIVE_Y_COMP_ESTIMATION,X_VECTOR_FAILUREFREE_NEGATIVE_ESTIMATION);
    POL_COEFFS_Y_POSITIVE = polynomial_approximation_2D_6th_ord(INPUT_Y_VALUES_POSITIVE_X_COMP_ESTIMATION,
      INPUT_Y_VALUES_POSITIVE_ESTIMATION,Y_VECTOR_FAILUREFREE_POSITIVE_ESTIMATION);
    POL_COEFFS_Y_NEGATIVE = polynomial_approximation_2D_6th_ord(INPUT_Y_VALUES_NEGATIVE_X_COMP_ESTIMATION,
      INPUT_Y_VALUES_NEGATIVE_ESTIMATION,Y_VECTOR_FAILUREFREE_NEGATIVE_ESTIMATION);
end
%% ALLOCATION OF VECTORS FOR CORRECTED SIGNALS:
NON_LIN_CORRECTED_X_POSITIVE = zeros(1,(181));
NON_LIN_CORRECTED_X_NEGATIVE = zeros(1,(181));
NON_LIN_CORRECTED_Y_POSITIVE = zeros(1,(181));
NON_LIN_CORRECTED_Y_NEGATIVE = zeros(1,(181));

%% CALCULATION OF CORRECTED SIGNALS:
if (DEGREE == 2)
    for i = 1:1:length(INPUT_X_VALUES_POSITIVE)
        NON_LIN_CORRECTED_X_POSITIVE(i) = POL_COEFFS_X_POSITIVE(1) + POL_COEFFS_X_POSITIVE(2)*INPUT_X_VALUES_POSITIVE(i) +
      POL_COEFFS_X_POSITIVE(3)*INPUT_X_VALUES_POSITIVE_Y_COMP(i) + POL_COEFFS_X_POSITIVE(4)*INPUT_X_VALUES_POSITIVE(i)*
      INPUT_X_VALUES_POSITIVE_Y_COMP(i) +POL_COEFFS_X_POSITIVE(5)*INPUT_X_VALUES_POSITIVE(i)^2  + POL_COEFFS_X_POSITIVE(6)*
      INPUT_X_VALUES_POSITIVE_Y_COMP(i)^2 ;
    end %OK
    for i = 1:1:length(INPUT_X_VALUES_NEGATIVE)
        NON_LIN_CORRECTED_X_NEGATIVE(i) = POL_COEFFS_X_NEGATIVE(1) + POL_COEFFS_X_NEGATIVE(2)*INPUT_X_VALUES_NEGATIVE(i) +
      POL_COEFFS_X_NEGATIVE(3)*INPUT_X_VALUES_NEGATIVE_Y_COMP(i) + POL_COEFFS_X_NEGATIVE(4)*INPUT_X_VALUES_NEGATIVE(i)*
      INPUT_X_VALUES_NEGATIVE_Y_COMP(i) +POL_COEFFS_X_NEGATIVE(5)*INPUT_X_VALUES_NEGATIVE(i)^2  + POL_COEFFS_X_NEGATIVE(6)*
      INPUT_X_VALUES_NEGATIVE_Y_COMP(i)^2 ;
    end %OK
```

```
113     %%
114     for i = 1:1:length(INPUT_Y_VALUES_POSITIVE)
115         NON_LIN_CORRECTED_Y_POSITIVE(i) = POL_COEFFS_Y_POSITIVE(1) + POL_COEFFS_Y_POSITIVE(2)*INPUT_Y_VALUES_POSITIVE_X_COMP(i) +
          POL_COEFFS_Y_POSITIVE(3)*INPUT_Y_VALUES_POSITIVE(i) + POL_COEFFS_Y_POSITIVE(4)*INPUT_Y_VALUES_POSITIVE_X_COMP(i)*
          INPUT_Y_VALUES_POSITIVE(i) +POL_COEFFS_Y_POSITIVE(5)*INPUT_Y_VALUES_POSITIVE_X_COMP(i)^2  + POL_COEFFS_Y_POSITIVE(6)*
          INPUT_Y_VALUES_POSITIVE(i)^2 ;
116     end %OK
117     for i = 1:1:length(INPUT_Y_VALUES_NEGATIVE)
118         NON_LIN_CORRECTED_Y_NEGATIVE(i) = POL_COEFFS_Y_NEGATIVE(1) + POL_COEFFS_Y_NEGATIVE(2)*INPUT_Y_VALUES_NEGATIVE_X_COMP(i) +
          POL_COEFFS_Y_NEGATIVE(3)*INPUT_Y_VALUES_NEGATIVE(i) + POL_COEFFS_Y_NEGATIVE(4)*INPUT_Y_VALUES_NEGATIVE_X_COMP(i)*
          INPUT_Y_VALUES_NEGATIVE(i) +POL_COEFFS_Y_NEGATIVE(5)*INPUT_Y_VALUES_NEGATIVE_X_COMP(i)^2  + POL_COEFFS_Y_NEGATIVE(6)*
          INPUT_Y_VALUES_NEGATIVE(i)^2 ;
119     end %OK
120 elseif (DEGREE == 4)
121     for i = 1:1:(181)
122         NON_LIN_CORRECTED_X_POSITIVE(i) = POL_COEFFS_X_POSITIVE(1) + POL_COEFFS_X_POSITIVE(2)*INPUT_X_VALUES_POSITIVE(i) +
          POL_COEFFS_X_POSITIVE(3)*INPUT_X_VALUES_POSITIVE_Y_COMP(i) + POL_COEFFS_X_POSITIVE(4)*INPUT_X_VALUES_POSITIVE(i)*
          INPUT_X_VALUES_POSITIVE_Y_COMP(i) +POL_COEFFS_X_POSITIVE(5)*INPUT_X_VALUES_POSITIVE(i)^2  + POL_COEFFS_X_POSITIVE(6)*
          INPUT_X_VALUES_POSITIVE_Y_COMP(i)^2 + +POL_COEFFS_X_POSITIVE(7)*INPUT_X_VALUES_POSITIVE(i)^4  + POL_COEFFS_X_POSITIVE(8)*
          INPUT_X_VALUES_POSITIVE_Y_COMP(i)^4;
123     end %OK
124     for i = 1:1:(181)
125         NON_LIN_CORRECTED_X_NEGATIVE(i) = POL_COEFFS_X_NEGATIVE(1) + POL_COEFFS_X_NEGATIVE(2)*INPUT_X_VALUES_NEGATIVE(i) +
          POL_COEFFS_X_NEGATIVE(3)*INPUT_X_VALUES_NEGATIVE_Y_COMP(i) + POL_COEFFS_X_NEGATIVE(4)*INPUT_X_VALUES_NEGATIVE(i)*
          INPUT_X_VALUES_NEGATIVE_Y_COMP(i) +POL_COEFFS_X_NEGATIVE(5)*INPUT_X_VALUES_NEGATIVE(i)^2  + POL_COEFFS_X_NEGATIVE(6)*
          INPUT_X_VALUES_NEGATIVE_Y_COMP(i)^2 + +POL_COEFFS_X_NEGATIVE(7)*INPUT_X_VALUES_NEGATIVE(i)^4  + POL_COEFFS_X_NEGATIVE(8)*
          INPUT_X_VALUES_NEGATIVE_Y_COMP(i)^4;
126     end %OK
127     %%
128     for i = 1:1:(181)
129         NON_LIN_CORRECTED_Y_POSITIVE(i) = POL_COEFFS_Y_POSITIVE(1) + POL_COEFFS_Y_POSITIVE(2)*INPUT_Y_VALUES_POSITIVE_X_COMP(i) +
          POL_COEFFS_Y_POSITIVE(3)*INPUT_Y_VALUES_POSITIVE(i) + POL_COEFFS_Y_POSITIVE(4)*INPUT_Y_VALUES_POSITIVE_X_COMP(i)*
          INPUT_Y_VALUES_POSITIVE(i) +POL_COEFFS_Y_POSITIVE(5)*INPUT_Y_VALUES_POSITIVE_X_COMP(i)^2  + POL_COEFFS_Y_POSITIVE(6)*
          INPUT_Y_VALUES_POSITIVE(i)^2 +POL_COEFFS_Y_POSITIVE(7)*INPUT_Y_VALUES_POSITIVE_X_COMP(i)^4  + POL_COEFFS_Y_POSITIVE(8)*
          INPUT_Y_VALUES_POSITIVE(i)^4;
130     end %OK
131     for i = 1:1:(181)
132         NON_LIN_CORRECTED_Y_NEGATIVE(i) = POL_COEFFS_Y_NEGATIVE(1) + POL_COEFFS_Y_NEGATIVE(2)*INPUT_Y_VALUES_NEGATIVE_X_COMP(i) +
          POL_COEFFS_Y_NEGATIVE(3)*INPUT_Y_VALUES_NEGATIVE(i) + POL_COEFFS_Y_NEGATIVE(4)*INPUT_Y_VALUES_NEGATIVE_X_COMP(i)*
          INPUT_Y_VALUES_NEGATIVE(i) +POL_COEFFS_Y_NEGATIVE(5)*INPUT_Y_VALUES_NEGATIVE_X_COMP(i)^2  + POL_COEFFS_Y_NEGATIVE(6)*
          INPUT_Y_VALUES_NEGATIVE(i)^2 +POL_COEFFS_Y_NEGATIVE(7)*INPUT_Y_VALUES_NEGATIVE_X_COMP(i)^4  + POL_COEFFS_Y_NEGATIVE(8)*
          INPUT_Y_VALUES_NEGATIVE(i)^4;
133     end %OK
134 elseif (DEGREE == 6)
135     for i = 1:1:(181)
136         NON_LIN_CORRECTED_X_POSITIVE(i) = POL_COEFFS_X_POSITIVE(1) + POL_COEFFS_X_POSITIVE(2)*INPUT_X_VALUES_POSITIVE(i) +
          POL_COEFFS_X_POSITIVE(3)*INPUT_X_VALUES_POSITIVE_Y_COMP(i) + POL_COEFFS_X_POSITIVE(4)*INPUT_X_VALUES_POSITIVE(i)*
          INPUT_X_VALUES_POSITIVE_Y_COMP(i) +POL_COEFFS_X_POSITIVE(5)*INPUT_X_VALUES_POSITIVE(i)^2  + POL_COEFFS_X_POSITIVE(6)*
          INPUT_X_VALUES_POSITIVE_Y_COMP(i)^2 +POL_COEFFS_X_POSITIVE(7)*INPUT_X_VALUES_POSITIVE(i)^4  + POL_COEFFS_X_POSITIVE(8)*
          INPUT_X_VALUES_POSITIVE_Y_COMP(i)^4 +POL_COEFFS_X_POSITIVE(9)*INPUT_X_VALUES_POSITIVE(i)^6  + POL_COEFFS_X_POSITIVE(10)*
          INPUT_X_VALUES_POSITIVE_Y_COMP(i)^6;
137     end %OK
138     for i = 1:1:(181)
139         NON_LIN_CORRECTED_X_NEGATIVE(i) = POL_COEFFS_X_NEGATIVE(1) + POL_COEFFS_X_NEGATIVE(2)*INPUT_X_VALUES_NEGATIVE(i) +
          POL_COEFFS_X_NEGATIVE(3)*INPUT_X_VALUES_NEGATIVE_Y_COMP(i) + POL_COEFFS_X_NEGATIVE(4)*INPUT_X_VALUES_NEGATIVE(i)*
          INPUT_X_VALUES_NEGATIVE_Y_COMP(i) +POL_COEFFS_X_NEGATIVE(5)*INPUT_X_VALUES_NEGATIVE(i)^2  + POL_COEFFS_X_NEGATIVE(6)*
          INPUT_X_VALUES_NEGATIVE_Y_COMP(i)^2 + +POL_COEFFS_X_NEGATIVE(7)*INPUT_X_VALUES_NEGATIVE(i)^4  + POL_COEFFS_X_NEGATIVE(8)*
          INPUT_X_VALUES_NEGATIVE_Y_COMP(i)^4 + POL_COEFFS_X_NEGATIVE(9)*INPUT_X_VALUES_NEGATIVE(i)^6  + POL_COEFFS_X_NEGATIVE(10)*
          INPUT_X_VALUES_NEGATIVE_Y_COMP(i)^6;
140     end %OK
141     %%
142     for i = 1:1:(181)
143         NON_LIN_CORRECTED_Y_POSITIVE(i) = POL_COEFFS_Y_POSITIVE(1) + POL_COEFFS_Y_POSITIVE(2)*INPUT_Y_VALUES_POSITIVE_X_COMP(i) +
          POL_COEFFS_Y_POSITIVE(3)*INPUT_Y_VALUES_POSITIVE(i) + POL_COEFFS_Y_POSITIVE(4)*INPUT_Y_VALUES_POSITIVE_X_COMP(i)*
          INPUT_Y_VALUES_POSITIVE(i) +POL_COEFFS_Y_POSITIVE(5)*INPUT_Y_VALUES_POSITIVE_X_COMP(i)^2  + POL_COEFFS_Y_POSITIVE(6)*
          INPUT_Y_VALUES_POSITIVE(i)^2 +POL_COEFFS_Y_POSITIVE(7)*INPUT_Y_VALUES_POSITIVE_X_COMP(i)^4  + POL_COEFFS_Y_POSITIVE(8)*
          INPUT_Y_VALUES_POSITIVE(i)^4 + +POL_COEFFS_Y_POSITIVE(9)*INPUT_Y_VALUES_POSITIVE_X_COMP(i)^6  + POL_COEFFS_Y_POSITIVE(10)*
          INPUT_Y_VALUES_POSITIVE(i)^6;
144     end %OK
145     for i = 1:1:(181)
146         NON_LIN_CORRECTED_Y_NEGATIVE(i) = POL_COEFFS_Y_NEGATIVE(1) + POL_COEFFS_Y_NEGATIVE(2)*INPUT_Y_VALUES_NEGATIVE_X_COMP(i) +
          POL_COEFFS_Y_NEGATIVE(3)*INPUT_Y_VALUES_NEGATIVE(i) + POL_COEFFS_Y_NEGATIVE(4)*INPUT_Y_VALUES_NEGATIVE_X_COMP(i)*
          INPUT_Y_VALUES_NEGATIVE(i) +POL_COEFFS_Y_NEGATIVE(5)*INPUT_Y_VALUES_NEGATIVE_X_COMP(i)^2  + POL_COEFFS_Y_NEGATIVE(6)*
          INPUT_Y_VALUES_NEGATIVE(i)^2 +POL_COEFFS_Y_NEGATIVE(7)*INPUT_Y_VALUES_NEGATIVE_X_COMP(i)^4  + POL_COEFFS_Y_NEGATIVE(8)*
          INPUT_Y_VALUES_NEGATIVE(i)^4 + +POL_COEFFS_Y_NEGATIVE(9)*INPUT_Y_VALUES_NEGATIVE_X_COMP(i)^6  + POL_COEFFS_Y_NEGATIVE(10)*
          INPUT_Y_VALUES_NEGATIVE(i)^6;
147     end %OK
148 end
149 % CONCATINATION OF VECTORS AND OUTPUT OF CORRECTED SIGNALS X and Y:
150 NON_LIN_CORRECTED_X_VALUES =  [NON_LIN_CORRECTED_X_POSITIVE(91:181) NON_LIN_CORRECTED_X_NEGATIVE(2:end-1) NON_LIN_CORRECTED_X_POSITIVE
        (1:91) ];
151 NON_LIN_CORRECTED_Y_VALUES =  [NON_LIN_CORRECTED_Y_POSITIVE(1:end) NON_LIN_CORRECTED_Y_NEGATIVE(2:end)];
152
153 end
```

Listing D.13: Application function for nonlinear angular error correction algorithm based on geometric distortion compensation

```
function [P_VECTOR] = polynomial_approximation_2D_4th_ord(INPUT_VECTOR_X_COORD, INPUT_VECTOR_Y_COORD, IDEAL_VECTOR)
%%-------------------------------------------------------------------------%%
% Function for estimation of the coefficients of 2D polynomial basis
% functions for angular error correction algorithm based on geometric
% distortion cancelation. (4th degree)
%
% Author:           Oleg Petrak
% Last update:      25.12.2018
%
% INPUT: Ideal vector (half of sine or cosine components) and two input
% vectors with x and y coordinates of the input signal to correct.
% OUTPUT: Coefficients of 2D polynomial.
%%-------------------------------------------------------------------------%%

%% A has the form:
%[ 1 x1 y1 x1*y1 x1^2 y1^2 x3^4 y3^4
%  1 x2 y2 x2*y2 x2^2 y2^2 x3^4 y3^4
%  1 x3 y3 x3*y3 x3^2 y3^2 x3^4 y3^4
%...]
%% Vectors allocation
b = IDEAL_VECTOR';
ONES_VECTOR = ones(1,length(IDEAL_VECTOR))';
%% Filling of the vectors with input values
X_Y_VECTOR = (INPUT_VECTOR_X_COORD.*INPUT_VECTOR_Y_COORD)';
X_SQ2_VECTOR = (INPUT_VECTOR_X_COORD.*INPUT_VECTOR_X_COORD)';
Y_SQ2_VECTOR = (INPUT_VECTOR_Y_COORD.*INPUT_VECTOR_Y_COORD)';
X_SQ4_VECTOR = (INPUT_VECTOR_X_COORD.*INPUT_VECTOR_X_COORD.*INPUT_VECTOR_X_COORD.*INPUT_VECTOR_X_COORD)';
Y_SQ4_VECTOR = (INPUT_VECTOR_Y_COORD.*INPUT_VECTOR_Y_COORD.*INPUT_VECTOR_Y_COORD.*INPUT_VECTOR_Y_COORD)';
%% Creation of the matrix A
A = [ONES_VECTOR,INPUT_VECTOR_X_COORD',INPUT_VECTOR_Y_COORD',X_Y_VECTOR,X_SQ2_VECTOR,Y_SQ2_VECTOR,X_SQ4_VECTOR,Y_SQ4_VECTOR];
%% Solve by using SVD:
P_VECTOR = A\b;
end
```

Listing D.14: Function for estimation of coefficients of rectification polynomials (4th degree example)

D.2.5 Data and program flow

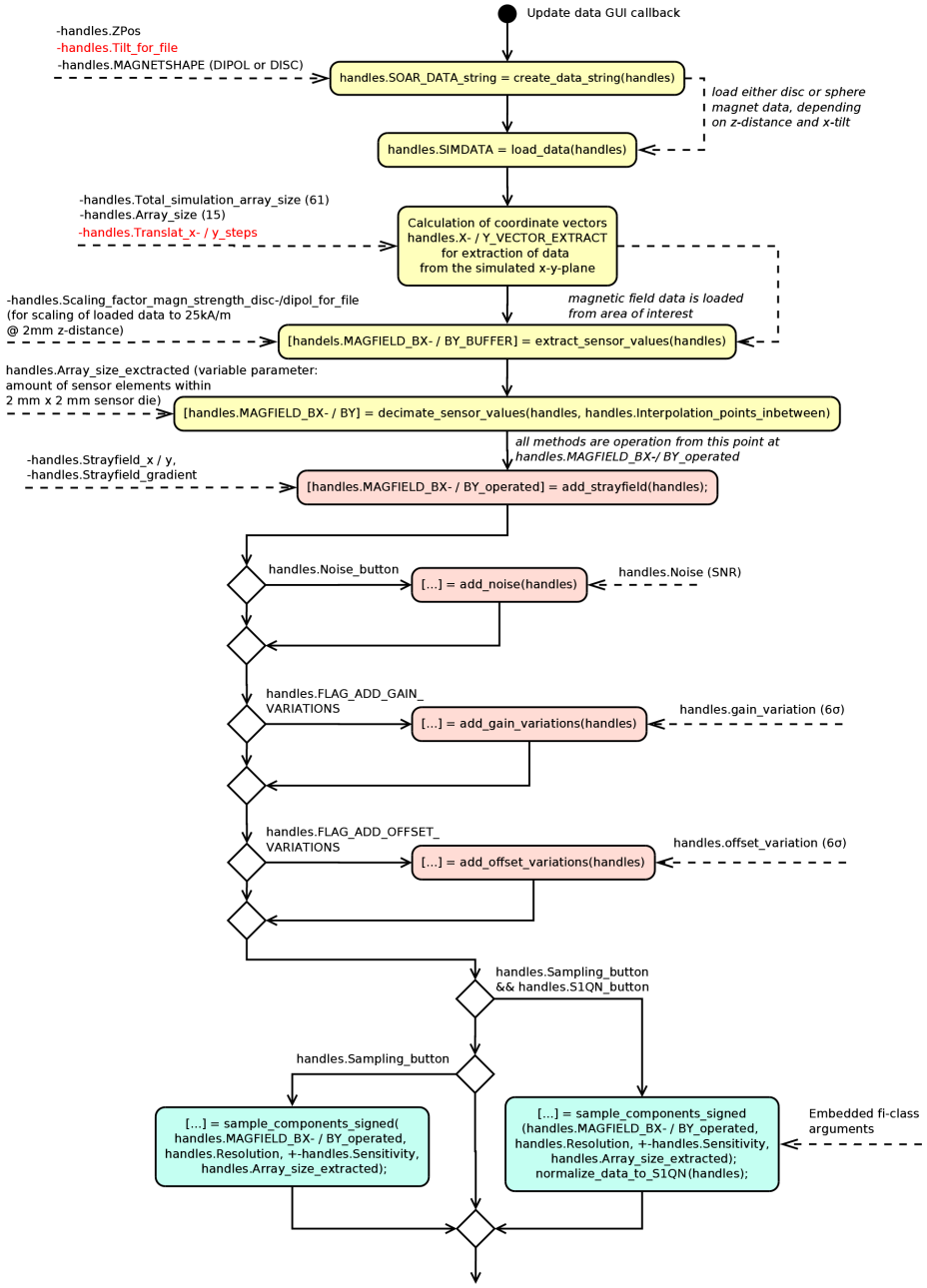


Figure D.3: UML activity diagram of main update function of created evaluation software. Part 1 of 2: Data import operations, interference and „digitalization“ of values.

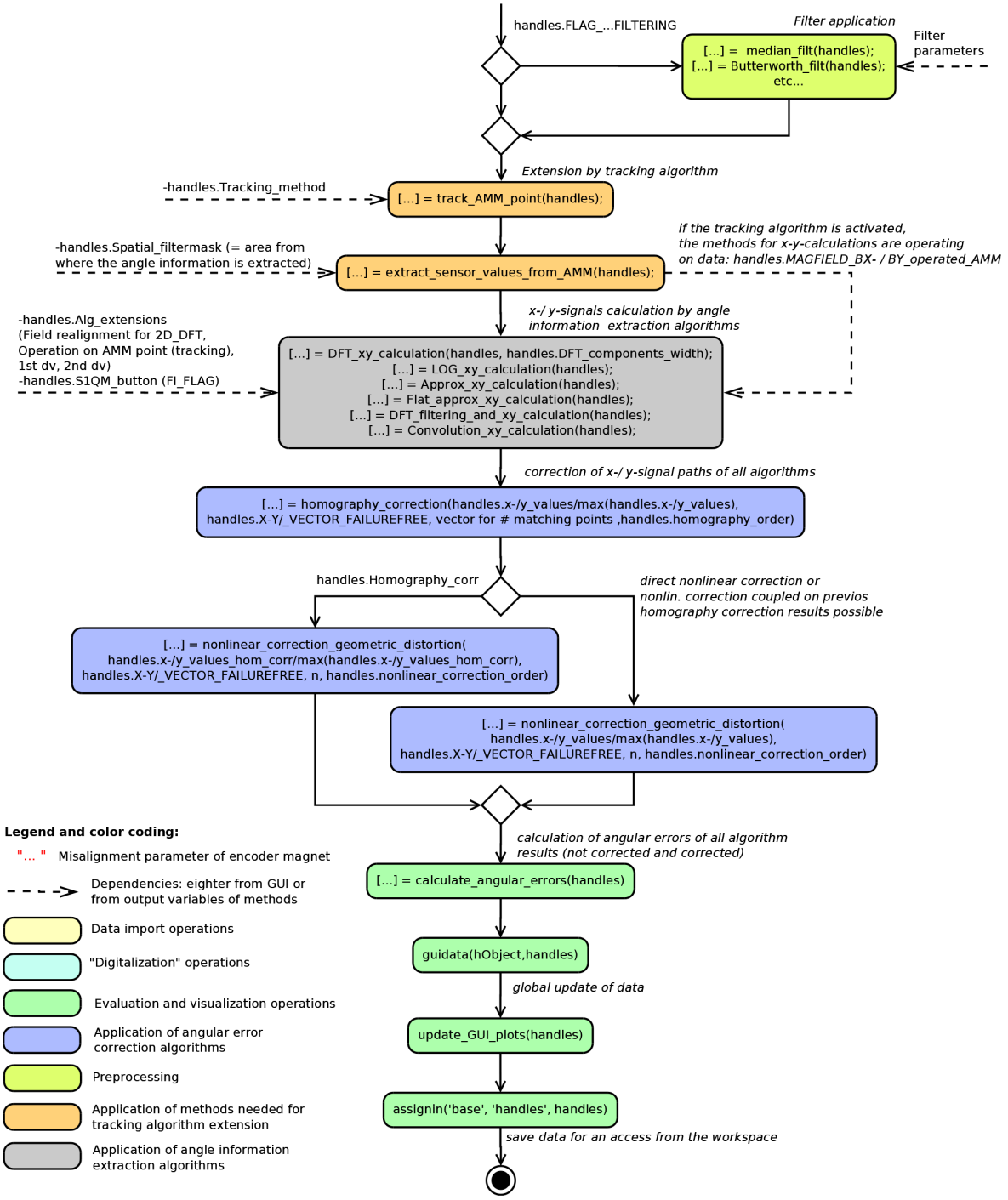


Figure D.4: UML activity diagram of main update function of created evaluation software. Part 2 of 2: Filtering, algorithmic processing and evaluation.

D.3 Relevant functions for investigations

D.3.1 Analog-to-Digital Converter

The absolute resolution of an ADC within a considered range is the smallest analog value (usually a voltage) which cause an increase of digital value of one digit referred to as one least significant bit (LSB, also called as the quantization digit / quantum Q) and is calculated by the formula [58]:

$$LSB = Q = \frac{Full\ scale\ range}{2^n} \hat{=} U_{LSB} = \frac{U_{FS}}{2^n} \tag{D.1}$$

Full scale range is the dynamic range of ADC and is represented electronically by a difference between two defined potentials which are referred to a certain capture range of a analog physical signal. The potential difference ranges from 0 to U_{FS} for an ADC with unipolar polarity, or associated with an offset within the range $\pm U_{FS}/2$ for a bipolar polarity ADC [58]. U_{FS} is in most applications equal to supply voltage level U_{vdd}.
A real ADC exhibits usually besides its quantization error, which is associated with resolution limitation of the ADC device, also called as quantization noise, further following errors regarding its accuracy [15]:

- **Offset error** - Initial ADC level $\neq$ 0
- **Gain error** - Slope of ADC $\neq$ 1
- **Integral nonlinearity INL** - Deviations between the ideal and real transfer function for transitions steps of digital output code. Can also be represented as variation of the slope within the full operational range of the ADC.
- **Differential nonlinearity DNL** - Differences between the voltage threshold levels $\neq$ U_{LSB} exclusively at transitions steps in digital output code.
- **Missing codes** - Represents a possible skipping of successive digital codes.

The quality of an ADC is often measured by SINAD (signal-to-noise and distortion ratio) where other distortion e.g. due to harmonics [55] and noise effects are considered and ENOB (effective number of bits) which takes into account incl. quantization signal-to-noise ratio also all the circuitry from input terminals to the data outputs, called also as overall measurement accuracy [15].
An ideal ADC has a shifted conversion curve and starts its data acquisition at LSB/2, as can be seen in Figure D.5. The error during signal digitalization which makes an ideal ADC is an uncorrelated sawtooth waveform having a peak-to-peak value of a quantum Q [56]. As a result out of that, the maximum error of an ideal ADC is $\pm$ 1/2 LSB, illustrated in Figure D.5 as well.

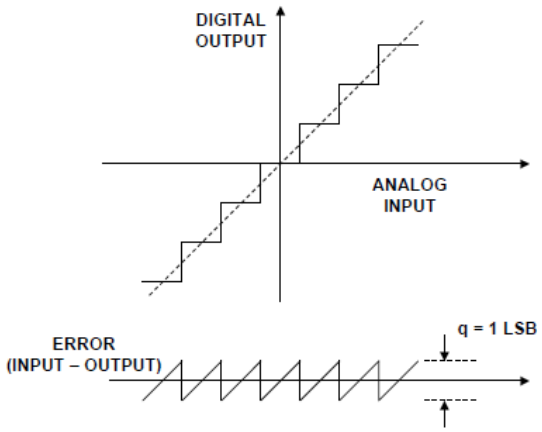


Figure D.5: Ideal n bit ADC transfer function and its quantization noise [56].

The sawtooth shaped waveform represents the ADC quantization error. The quality of digitalization of a signal by an ideal ADC can be calculated by assuming of a full scale sinusoidal input signal. With consideration of RMS values of sinusoidal input and quantization waveform using Equation 4.22 the signal-to-noise ratio (which is generally calculated by Formula 4.5) of an ideal ADC becomes determinable. In dependence of the resolution the SNR of an ideal ADC is calculated by the formula [56]:

$$SNR = 20log_{10}(6.02 \cdot N) + 1.76dB \tag{D.2}$$

over direct component to $F_S/2$, where N is the ADC resolution in bit.

D.3.2 Signal-to-noise ratio

In general, the following noise types represent the most important sources for an decrease of signal quality of a digital semiconductor device for sensing of magnetic fields:

- $1/f$ noise, also called Pink or Flicker noise. Its spectral power density decreases by 3 dB per octave. This noise kind becomes concern for bandwidth limited devices which are operating at lower bandwidths [5].
- $1/f^2$ noise, also called Brownian noise, with a spectral power density decrease of 6 dB per octave. However, practically less relevant but can still decrease the system accuracy in particular bandwidth limited applications [5].
- Johnson-Nyquist noise, also called as thermal noise. This kind of noise is temperature dependent. This noise is caused by the thermal agitation of the charge carriers inside an electrical conductor. It exist at equilibrium state, so regardless of voltage application. It is present in all electrical circuits, sensitive electronic equipment and represents the limiting factor of sensitivity of an electronic sensor [7]. Thermal noise exhibits in the most cases a white noise behavior with a Gaussian amplitude distribution [2], [50].

- Clock jitters in digital electronics. May be more concern for high speed operation devices in association to the ADCs. Variation in the time at which the S&H switch is opened (aperture uncertainty) results in an error voltage. This error is proportional to the magnitude of the jitter and the slew rate of operational amplifiers [32].
- Coupling effects e.g. capacitive coupling or distortions from the supply lines, couplings from chopping devices etc. The immunity of electronic units against the coupling effects is often measured in PSRR[1]).

The bandwidth issues in the time domain are not covered within this work. One can say that the encoder-sensor system is operating in this simulation setup at a nominal rotational frequency of 1 Hz. Therefore, $1/f$ and $1/f^2$ are not represented in the software. Rather, in this work the noise which occurs in the spatial domain of the sensor array plane is of interest. The digital noise sources as well as coupling effects are manufacturer and sensor system complexity depend. Thus, it plays a subordinate relevance.

D.3.3 Gain and offset deviations

In the automotive applications, the upper and lower limit (max. and min. values specified in a data sheet) for deviations of hardware parameters from a typical or nominal value corresponds to the 6 σ limit, what represents a process requirement [68]. The σ represents the population standard deviation, as already explained in Section 4.2.4. In the industry the indication of σ is used as a measure of the variation in a data set which is collected about a process. If a defect in a product or some hardware parameters are defined by specification limits separating it from outcomes of a process classified as „bad“, a 6 sigma process has in this case a mean / average value, that is 6 standard deviations away from the specification limit [12]. Consequently, the $\pm$ limit for the deviation of the expected value represent the absolute upper and lower specification limits of a process, which are required to met. Based on the 6 sigma limits a strategy was derived which is nowadays used by many manufacturers as a quality management method for continuous process improvement, e.g. application of DMAIC-model [46]. Figure D.6 depicts the 6 sigma process limits in the population standard deviation.

[1] Power Supply Rejection Ratio

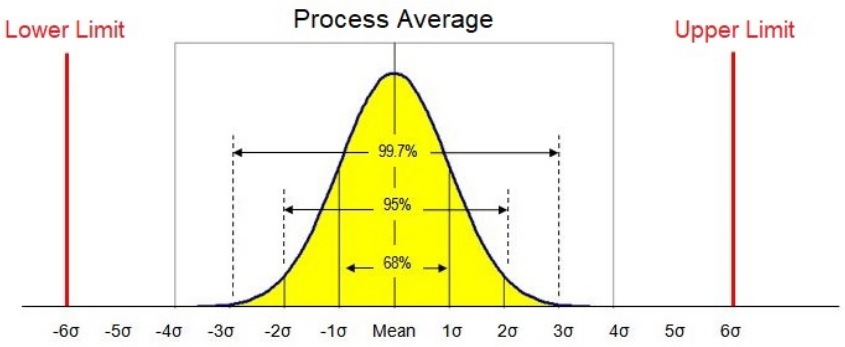


Figure D.6: Six sigma process limits in the population standard deviation [12, modified].

In the automotive applications, where all the electronic components play a significant role for the safety of passengers. By following the 6 σ requirement, the amount defect parts per million is 3.4. This corresponds to a process yield of 99.99966% [12].

D.3.4 Fixed-point arithmetic

For the purpose of representation of negative numbers in digital signal processing two's complement format is used in computer arithmetics [80]. The advantage of two's complement data representation is the fact that the subtraction operations are done by addition of the subtrahend encoded in two's complement and the minuend. Hence, no subtraction units are required in the hardware. Characteristically for this format is that the zero value is unambiguously encoded. The coverage of negative numbers range is 1 binary value more than the positive number range [80]. A decimal number encoded in binary two's complement format is represented by the equation [80]:

$$z = -b_{n-1} \cdot 2^{n-1} + \sum_{i=n-2}^{0} b_i \cdot 2^i \tag{D.3}$$

where b_{n-1},...,b_0 is a n-digits binary number and z its representation as a decimal integer number. In many application it is also necessary to represent the rational numbers. For this purpose the binary encoding of numbers in *sQm.n*-format is used [10], denoted in German literature as SnQm [80]. The *s* represents the signedness of the number, *m* denotes the amount of bits for the integer representation and *n* the amount of bits for the numbers behind the radix point. The decimal interpretation of the binary number encoded in fixed-point format is calculated by the equation [80]:

$$z = -b_m \cdot 2^m + \sum_{i=m-1}^{0} b_i \cdot 2^i + \sum_{i=1}^{n} b_i \cdot 2^{-i} \tag{D.4}$$

where $b_m ... b_0 ... b_n$ is a n-digits binary number and z its representation as a decimal number. Figure D.7 illustrates the signed decimal numbers in binary in two's complement format.

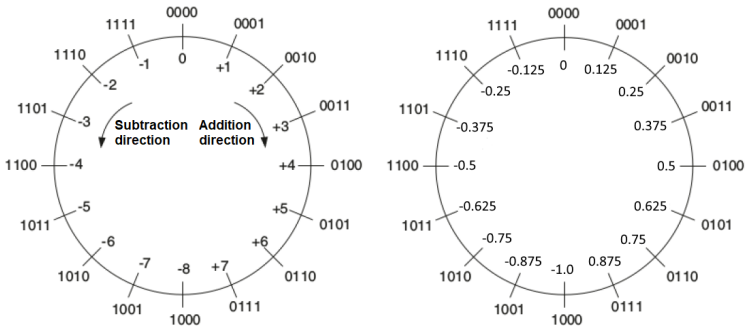


Figure D.7: (a) Number circle illustrating the representation of signed decimal number in binary 4 bit two's complement format [96, modified]. (b) sQ3 binary two's complement format [96, modified] and [80, redrawn].

In Figure D.8 a schematically structure of a storage register for data encoded in s1Q10-format and the significance of its bits is shown.

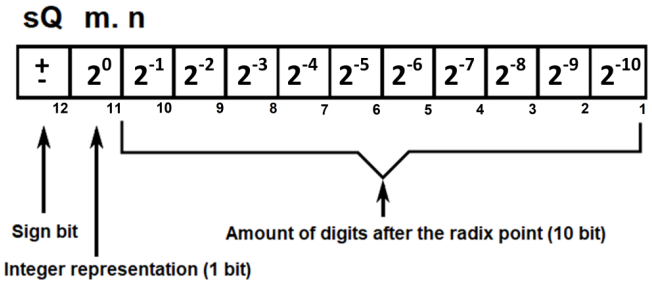


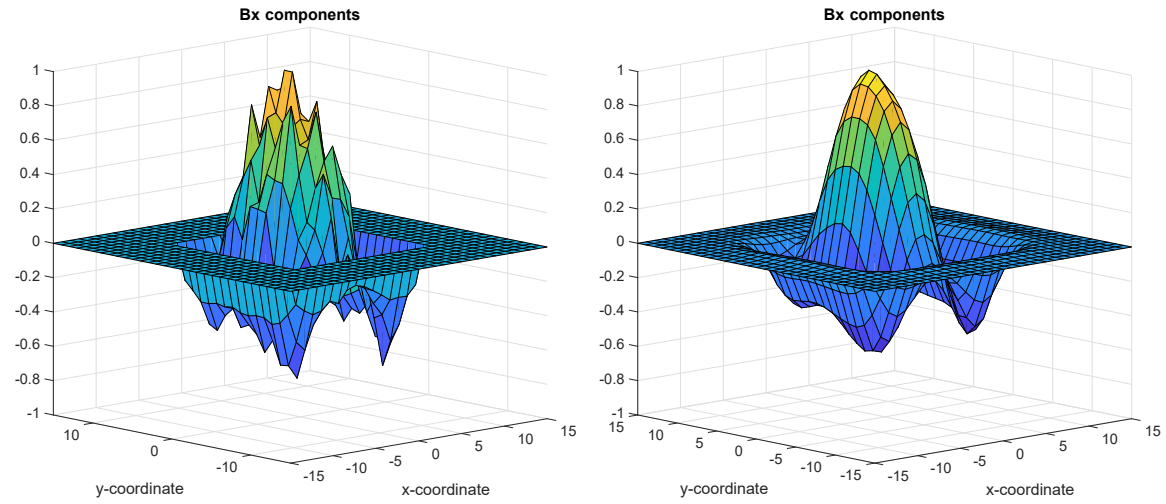
Figure D.8: Schematically structure of a data storage register encoded in s1Q10-format [60, modified]
Remark: the purpose if the illustration is the significance explanation. The value representation corresponds here to the normal binary format.

Appendix E: Preprocessing

E.1 Application of Butterworth low pass filter

E.1.1 Zero padded data arrays

The Figure E.1 illustrates the zero padding procedure and the filtering result of the investigation carried out in Section 5.2.1. The input data array were increased from 15 x 15 to 31 x 31 elements. After the application of Tukey window the data was processed in spatial frequency domain with a Butterworth low pass filter of order = 2 and cut-off frequency = 5.



(a) (b)

Figure E.1: Comparison of filtering effects gained from filtering of a not padded and zero padded signals in spatial frequency domain (31x31 2D-DFT). (a) Distorted zero padded $\mathbf{B_x}$ field components after window application. (b) Zero padded filtered $\mathbf{B_x}$ field components. No remained noise components are observable. *Simulation setup: 15 x 15 array (data zero padded to 31 x 31 elements), 2 mm z-distance, sphere encoder magnet, angular magnet position = 0°, 0.26 mm y-translation, 38 dB SNR.*

An extremely well noise suppression was obtained by of windowing and zero matching procedure in comparison to a filtering procedure applied on direct sensor array data without zero padding, see results and discussion in 5.2.1.

E.2 Median filter application

The content in this section represents an addendum to the investigation carried out in Section 5.2.2. In Figure E.2 a comparison between ideal, noise interfered and filtered field components caused by filtering with median filter for disc magnet data is shown.

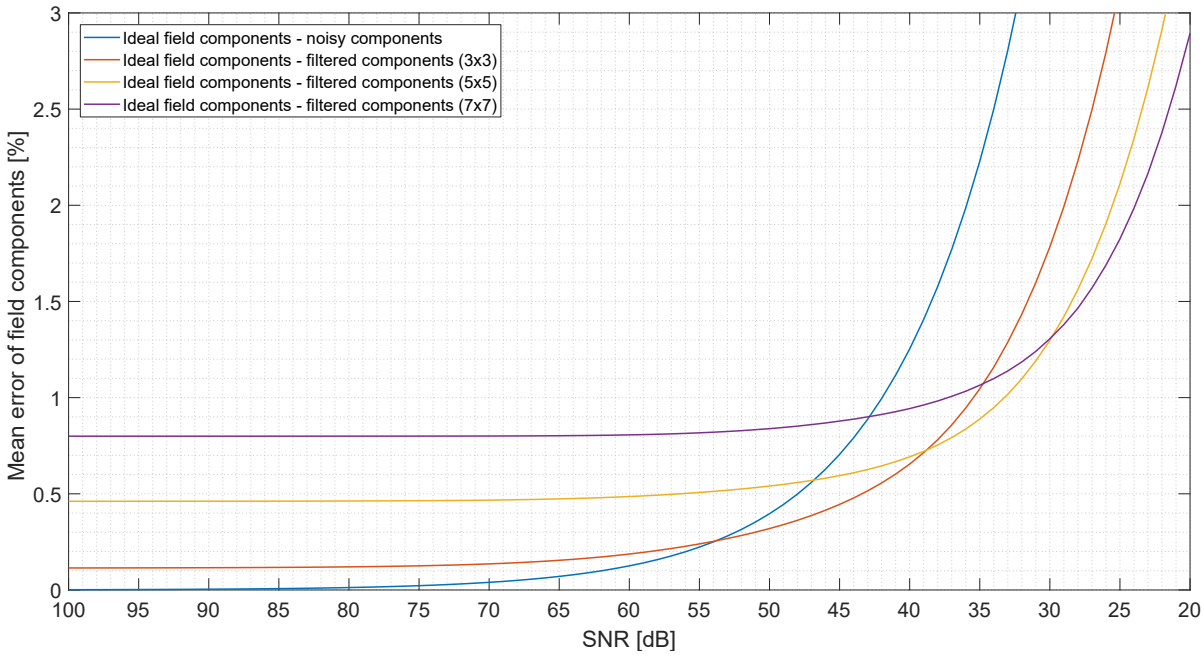


Figure E.2: Mean error of disc magnet field components in % interfered by white Gaussian noise and components filtered by median filter in dependence of SNR. Break-even points are observable at around 54 dB, 47 dB and 43 dB SNR for median filter kernels of 3 x 3, 5 x 5 and 7 x 7 elements.

In case of using the disc encoder magnet all three masks cause an initial deformation of field components which is under 1%. First break-even point is reached at around 54 dB SNR for the 3 x 3 median filter kernel, 47 dB for the filter size of 5 x 5 and 43 dB for 9 x 9 kernel. Break-even points between considered filter mask sizes are shifted to the higher SNR levels as well.

Appendix F: Optimizations

F.1 Exponent n optimization

This Appendix part covers the prove of optimization results of the field denominator matrix exponent n required by the modified approximation algorithm. Chosen cases for the z-distance, array size and extraction mask size are shown below for both magnet shapes. The optimized parameters for the disc magnet can be find on the attached DVD-ROM in the file *Optimization_n_disc.mat*. The structure of data contained in this file be derived from the optimization file *Exponentoptimization*.

F.1.1 Sphere magnet

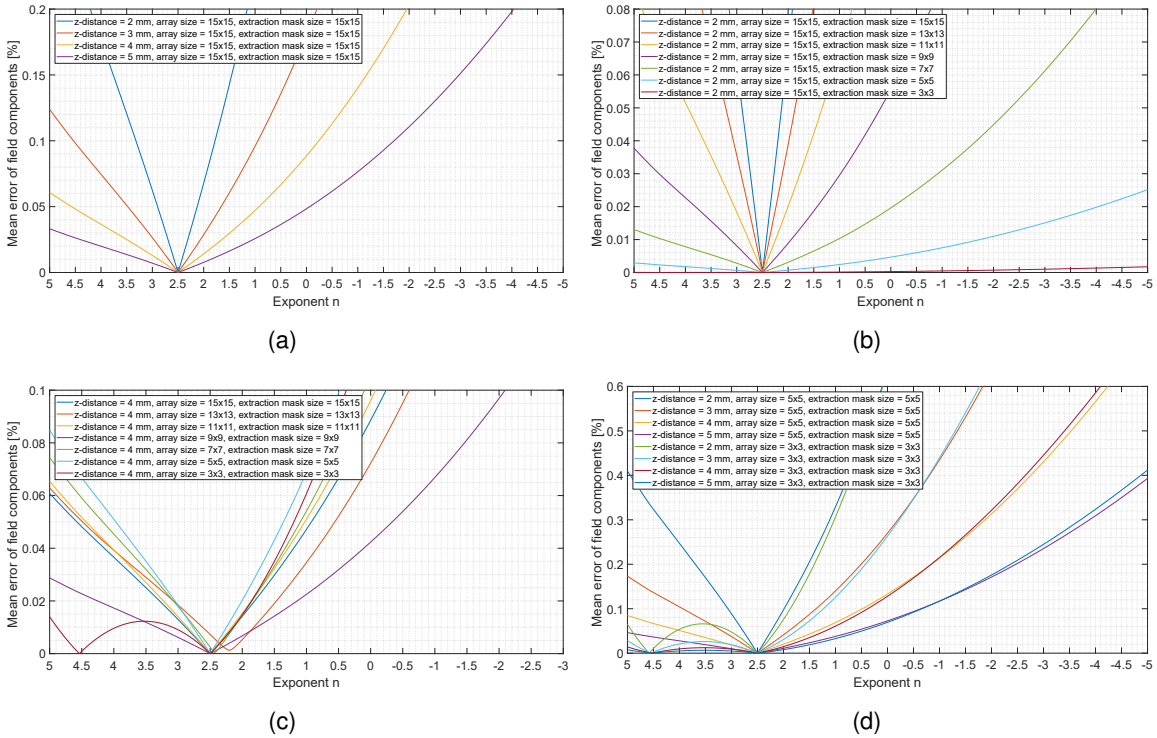


Figure F.1: Optimization of exponent n for the sphere magnet

At an array size of 3 x 3 element two minima are existent. This is a systematic issue. At a amount of sensor points of 3 x 3 there are two states of the field shape possible which fit to the components perfectly. A small deviation from $n = 2.5$ is visible in F.1(c) for an array size of 11

x 11 and 7 x 7 elements. This is most probably associated with interpolation procedure needed for creation of array sizes between 3 x 3 and 11 x 11 elements.

F.1.2 Disc magnet

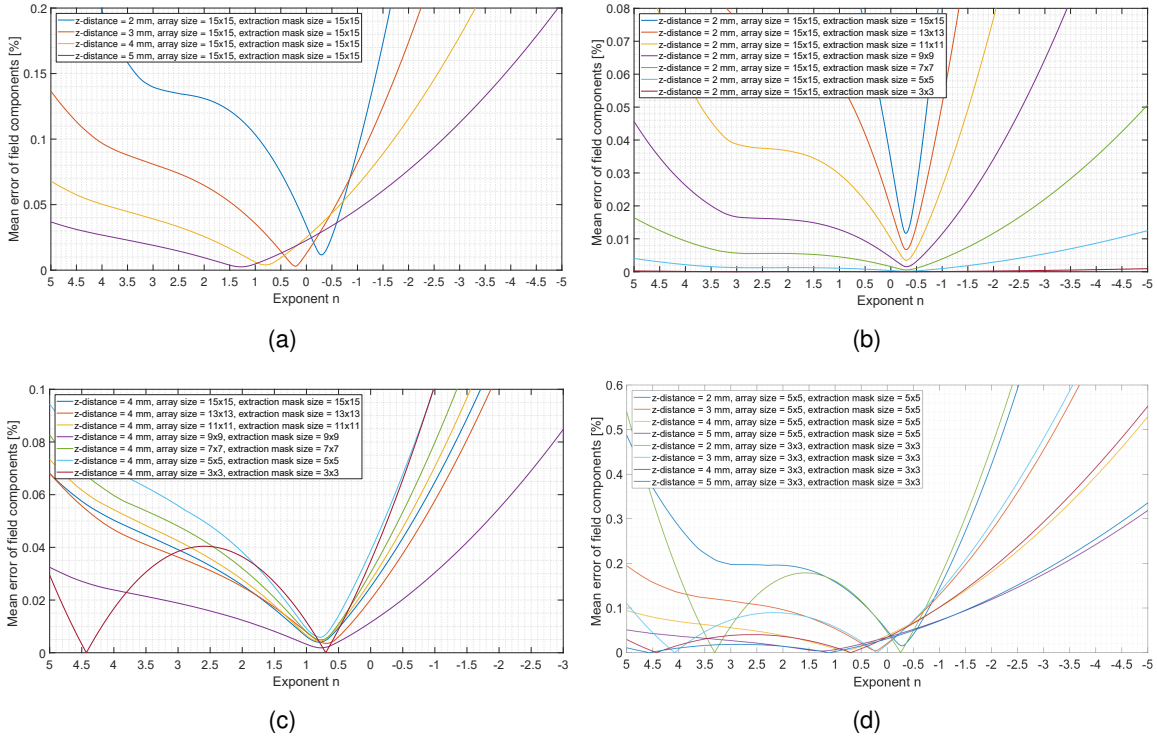


Figure F.2: Optimization of exponent n for the disc magnet

Also here two minima for an array size of 3 x 3 elements are observable. Interesting fact is that fir the closer distance to the encoder magnet the exponent n is negative. This is due to the near field behavior of the disc magnet field, see discussion in Section 5.3.2.

F.2 LoG filter bandwidth optimization

The determined kernel masks of the LoG filter with exceeded σ parameter are depicted in this section. Notable is the fact that at filter kernel sizes of 5 x 5, 7 x 7, 11 x 11 elements values in the masks are existing that are equal to zero.

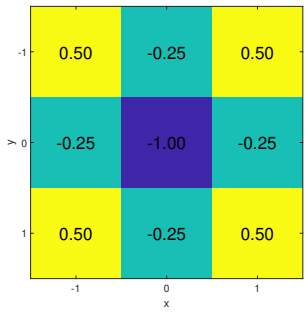


Figure F.3: Optimized LoG filter coefficients for 3x3 kernel size

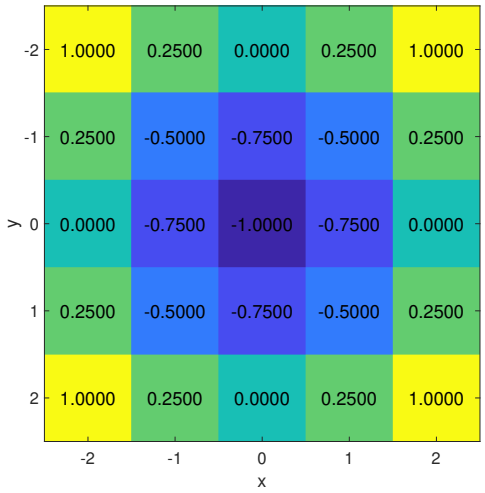


Figure F.4: Optimized LoG filter coefficients for 5x5 kernel size

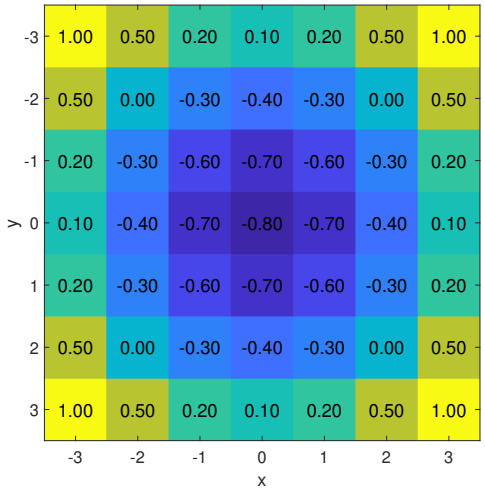


Figure F.5: Optimized LoG filter coefficients for 7x7 kernel size

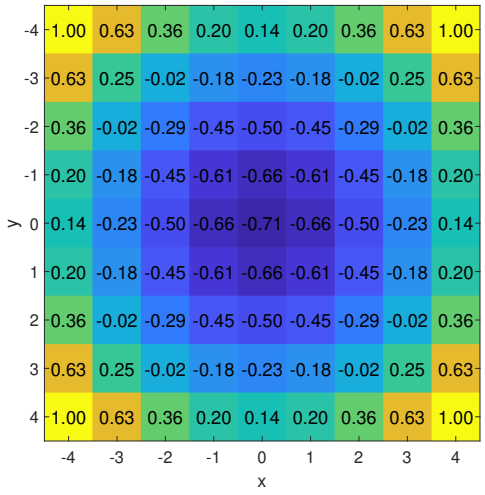


Figure F.6: Optimized LoG filter coefficients for 9x9 kernel size

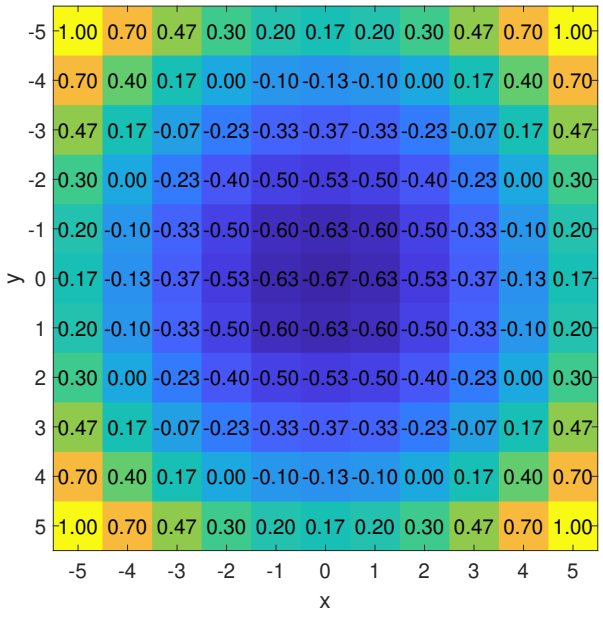


Figure F.7: Optimized LoG filter coefficients for 11x11 kernel size

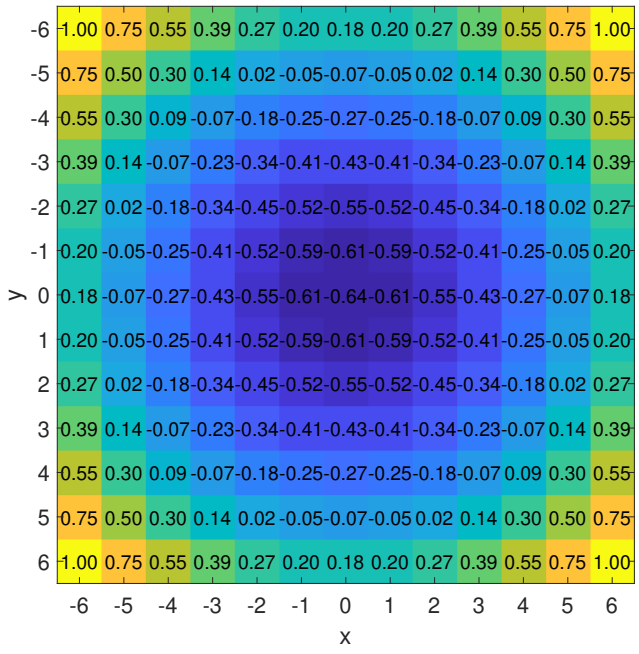


Figure F.8: Optimized LoG filter coefficients for 13x13 kernel size

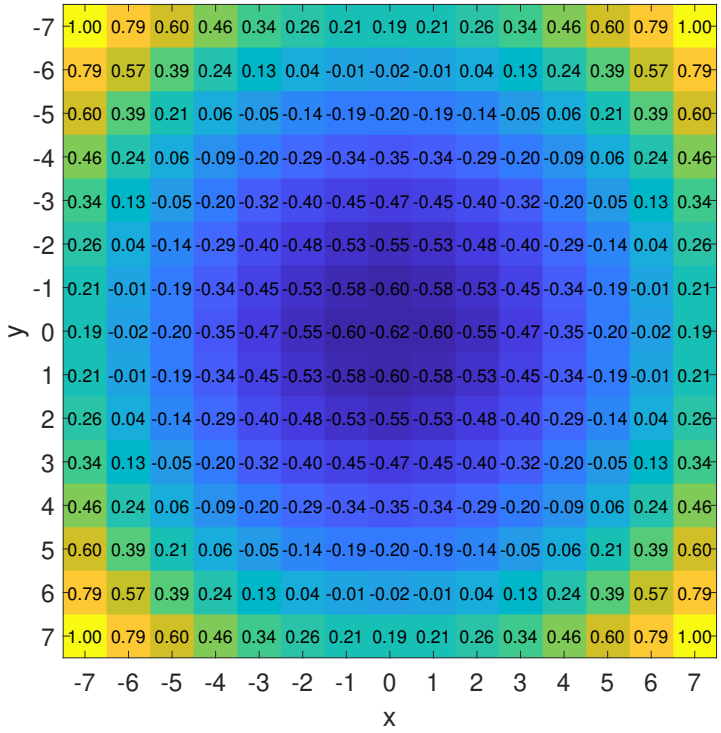


Figure F.9: Optimized LoG filter coefficients for 15x15 kernel size

Appendix G: Investigation results

The investigation results as well as the corresponding test script and evaluation files except the result that were performed directly from the GUI can be found on the attached DVD-ROM.

It is reccomended to read the Appendix D to get an understanding about evaluation panels of GUI evaluation software. A lot of results discussed in this section are based on direct evaluation of results in the GUI.

G.1 Stray field suppression

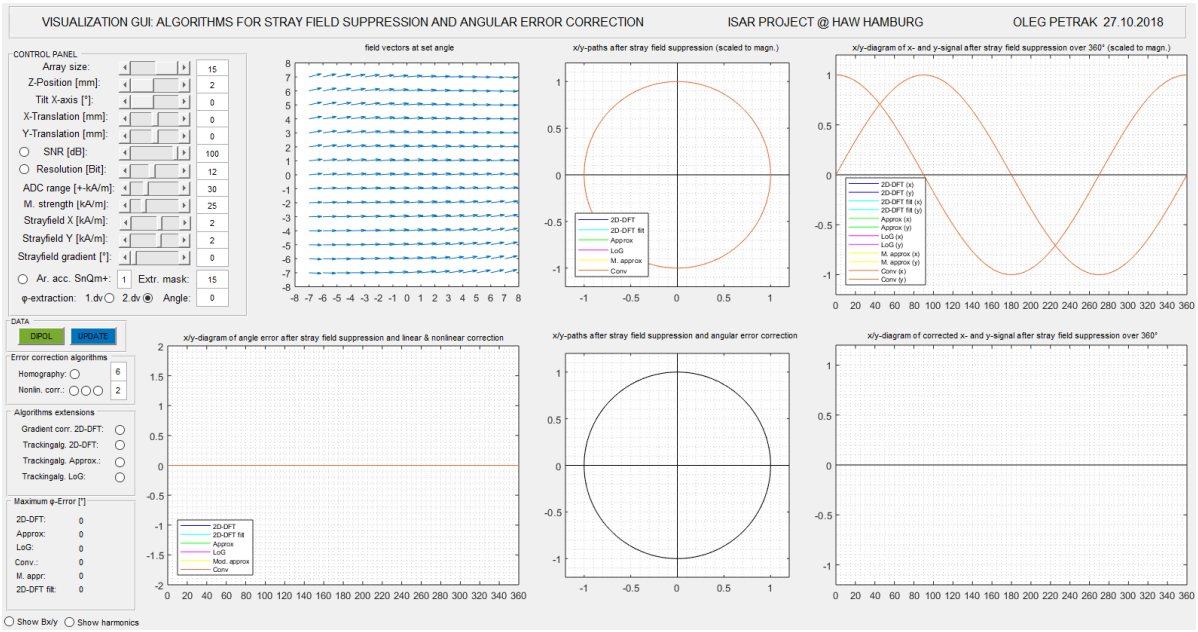
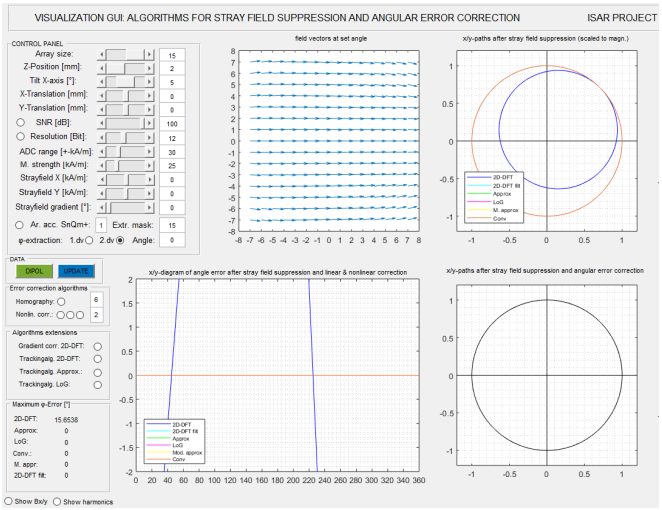
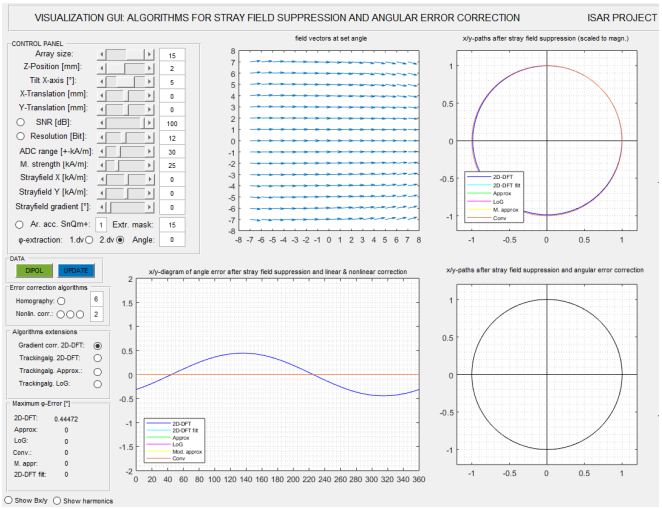


Figure G.1: Application of algorithms and evaluation of their results in created GUI: stray field robustness test. Stray field suppression is guaranteed by all algorithms. Configuration of applied stray field: 2 kA/m in x- an y- direction. Angular error is 0°. Consider the panel *Maximum φ-error [°]:* 2D-DFT, Approx., LoG, Conv. and M. appr. are the algorithms described in Section 3.3. 2D-DFT filt. is the algorithm proposed in [53].

G.2 Reasons for 2D-DFT extension by dynamic realignment of field components



(a) Not extended 2D-DFT algorithm.



(b) Extended 2D-DFT algorithm.

Figure G.2: Application of algorithms and evaluation of their results in created GUI: 2D-DFT extension. Application of algorithms and evaluation of their results in created GUI: 2D-DFT extension by dynamic realignment of field components. a) Maximum angular error at a x-tilt configuration of 5° without the realignment of field components extension is *extreme* and amounts to $\approx 15.6°$ what represents practically an unacceptable value. b) Maximum angular error at an x-tilt configuration of 5° in case of activated dynamic realignment of field components extension amounts to $\approx 0.44°$. By activation of dynamic realignment of field components extension the angular error of the 2D-DFT algorithms reduces extremely, However, it remains a maximum error, what is still a relative high value with respect to the set angular requirement of $\varphi_{err} \leq 1°$.

G.3 Misalignment sensitivity of algorithms

Figure G.3 shows sensitivity of all algorithms for angle information extraction including stray field suppression coupled into the evaluation software against a relative tiny randomly chosen misalignment case. Extended 2D-DFT, Approximation, single LoG and convolution algorithms show a maximum angular error of approximately 1.2°. The algorithm proposed in [53] exhibit an error of ≈ 3.2°. Modified approximation approach shows an extreme error of almost 40°.

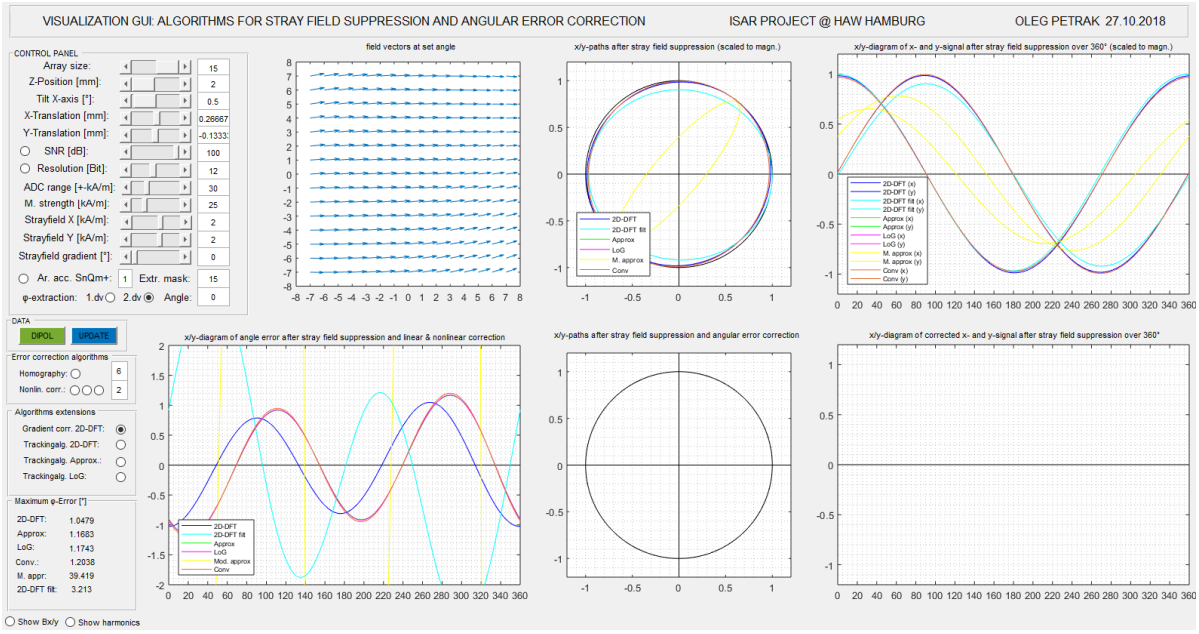
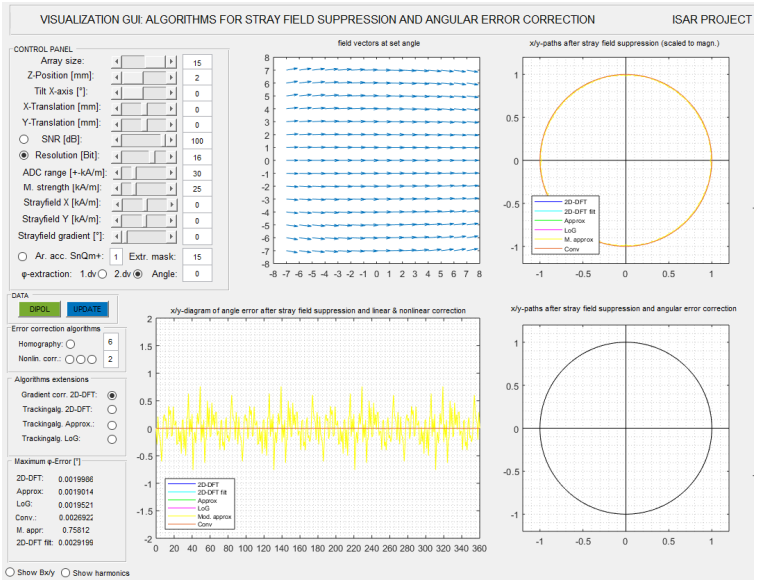


Figure G.3: Application of algorithms and evaluation of their results in created GUI. A small misalignment of the encoder magnet is applied in the control panel: tilt related to the x-axis of 0.5° and a small translation of 0.266 mm in x- and -0.133 in y-direction. The angular error of the extended 2D-DFT algorithm counts in this case to approximately 1.05° degree. Approximation, LoG and convolution algorithms exhibit a maximum angular error of ≈ 1.1 to 1.2°. The modified approximation algorithm shows an **extreme maximum error** of almost 40°. Algorithm in [53] exhibits an angular error of 3.2°.

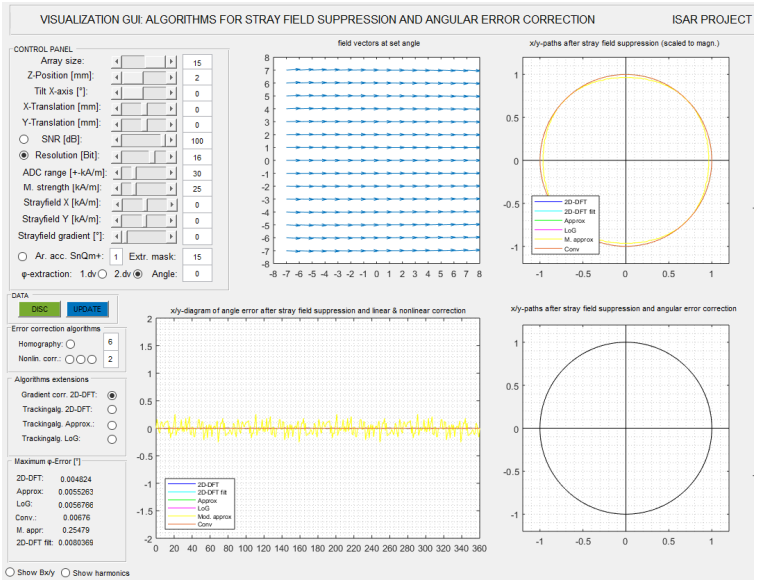
G.4 Reasons for exclusion of the modified approximation algorithm

From the modified approximation algorithm a much higher stability and angular accuracy was expected due to a significant well matching of shape of approximated function with measured field data. Unfortunately, according to the obtained results from preliminary investigation the modified approximation algorithm exhibits the worst sensitivity against quantization. It represents an extreme outlier and hence was not considered within framework of main investigations. The observed effects of this issue are shown and discussed in this section.
Figure G.4 depicts the angular error of the algorithms resulting from the digital acquisition of

measured field components of sphere and disc magnet by use of an 16 ADC. In comparison to it in Figure G.5 the quantization effects resulting from application of an 12 bit ADC are shown.



(a) Sphere encoder magnet.

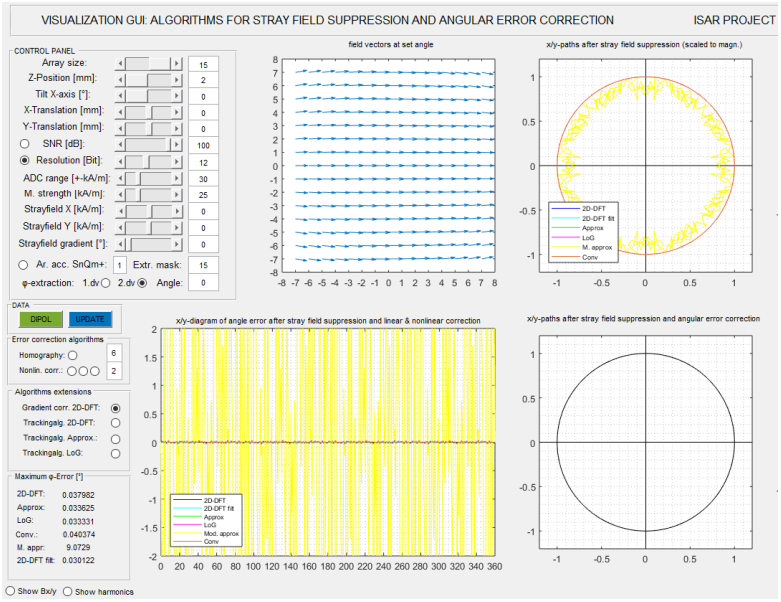


(b) Disc encoder magnet.

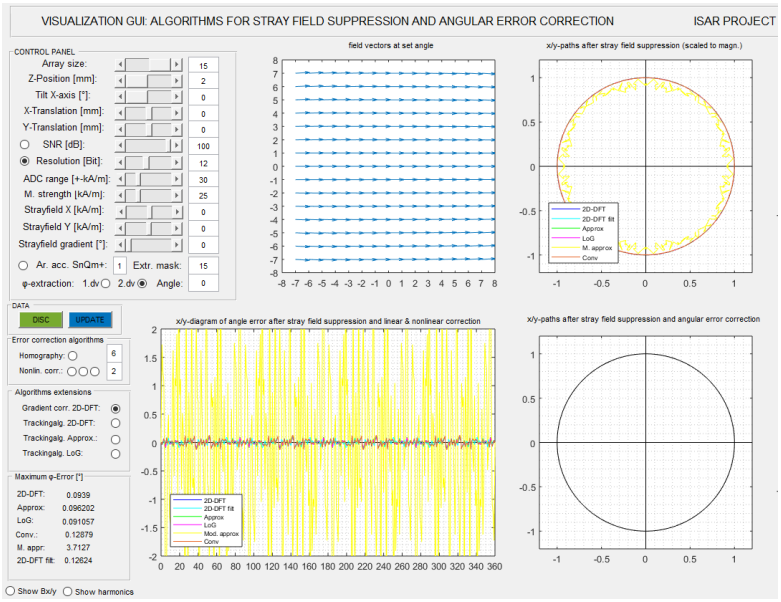
Figure G.4: Application of algorithms and evaluation of their results in created GUI: Quantization test 1. 16 bit ADC resolution. a) Algorithm results applied on sphere encoder magnet data. b) Algorithm results applied on disc encoder magnet data. Modified approximation algorithm exhibit an extreme outlier regarding the quantization noise sensitivity. Interesting fact is that the quantization noise for the disc magnet data is less than the quantization angular error if sphere magnet is used.

At an relative high ADC resolution of 16 bit there is still an quantization effect in the progression

of the angular error of the modified approximation algorithm visible. In both cases the maximum angular error is much higher than from its competitors: for the sphere encoder magnet 0.7° instead of 0.002° and for disc encoder magnet 0.25° in comparison to 0.004° from other candidates in the group.



(a) Sphere encoder magnet.



(b) Disc encoder magnet.

Figure G.5: Application of algorithms and evaluation of their results in created GUI: Quantization test 1. 12 bit ADC resolution. a) Algorithm results applied on sphere magnet data. b) Algorithm results applied on disc magnet data. The noise sensitivity of the modified approximation algorithm is unacceptable.

At an ADC resolution of 12 bit the quantization effect in the progression of the angular error of the modified approximation algorithm is extreme. In both cases the maximum angular error is significant higher than from the others: for the sphere encoder magnet 9° instead of 0.04° and for disc encoder magnet 3.7° in comparison to 0.01°. In Figure G.6 a small misalignment of the encoder magnet is applied. The ADC resolutions is set to 16 bit.

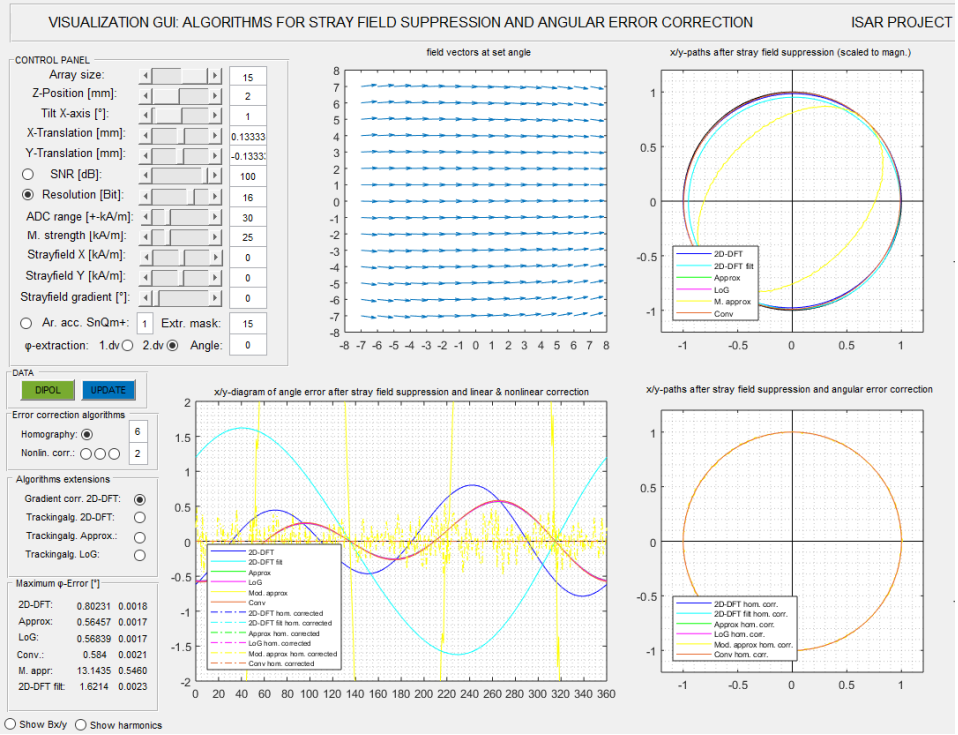


Figure G.6: Application of algorithms and evaluation of their results in created GUI: a small misalignment of the encoder magnet is applied in the control panel. Angular error correction is activated. The modified approximation algorithm shows the highest initial maximum error due to the misalignment of almost 13°. All errors are completely corrected by applied algorithm. In comparison to the whole group the remained maximum angular error of modified approach amount to 0.5° while the errors of the rest candidates is around 0.002° only.

It is suspected that the sensitivity problem is associated with the division by additional terms included the denominator during the estimation process. In case of including of the field weakening denominator matrix the values get changed in kind of a „manipulation“. Measured values especially near the boundaries are treated not in the same way as the values in the center of the sensor array and therefore not valued equally. This impact becomes visible not only in case of acquiring of values but also in case of misalignment. In any case, this issue shall be definitely investigated in the context of further work.

G.5 Proof of concept

G.5.1 Subblock 1.0

ADC resolution [bit]	**Array size:** / **Error [°]:**	3x3		5x5		7x7		9x9		11x11		13x13		15x15	
		max	mean	max	mean	max	mean	max	mean	max	mean	max	mean	max	mean
8	*Ext. 2D-DFT*	1.232	0.457	1.223	0.355	1.090	0.338	1.144	0.304	0.695	0.200	0.739	0.184	0.588	0.157
	Approx.	1.232	0.457	1.231	0.354	1.097	0.346	1.101	0.301	0.695	0.196	0.741	0.174	0.586	0.152
	LoG	1.232	0.456	1.235	0.353	1.103	0.346	1.108	0.302	0.695	0.198	0.740	0.172	0.584	0.152
	Conv.	1.232	0.456	1.558	0.500	1.327	0.433	1.411	0.403	1.039	0.285	0.768	0.221	0.639	0.187
10	*Ext. 2D-DFT*	0.447	0.173	0.335	0.119	0.310	0.077	0.258	0.065	0.196	0.064	0.120	0.053	0.114	0.039
	Approx.	0.447	0.173	0.325	0.122	0.265	0.076	0.244	0.066	0.189	0.064	0.105	0.048	0.112	0.037
	LoG	0.446	0.173	0.325	0.122	0.268	0.076	0.243	0.065	0.186	0.064	0.113	0.049	0.111	0.037
	Conv.	0.446	0.173	0.436	0.165	0.371	0.092	0.250	0.077	0.177	0.069	0.165	0.062	0.126	0.046
12	*Ext. 2D-DFT*	0.084	0.027	0.064	0.023	0.079	0.019	0.061	0.020	0.049	0.014	0.045	0.017	0.038	0.010
	Approx.	0.084	0.027	0.069	0.023	0.074	0.020	0.065	0.018	0.044	0.013	0.040	0.016	0.034	0.010
	LoG	0.084	0.027	0.068	0.023	0.073	0.020	0.065	0.018	0.042	0.013	0.039	0.016	0.033	0.010
	Conv.	0.084	0.027	0.112	0.035	0.078	0.025	0.061	0.024	0.045	0.017	0.059	0.021	0.040	0.011
14	*Ext. 2D-DFT*	0.034	0.010	0.016	0.006	0.018	0.005	0.015	0.004	0.012	0.004	0.006	0.002	0.007	0.002
	Approx.	0.034	0.010	0.016	0.006	0.018	0.005	0.014	0.004	0.011	0.004	0.006	0.002	0.006	0.003
	LoG	0.034	0.010	0.016	0.006	0.018	0.005	0.014	0.004	0.011	0.004	0.007	0.002	0.007	0.002
	Conv.	0.034	0.010	0.024	0.008	0.018	0.006	0.019	0.006	0.015	0.005	0.008	0.003	0.009	0.003
16	*Ext. 2D-DFT*	0.007	0.002	0.004	0.002	0.004	0.001	0.004	0.001	0.003	0.001	0.002	0.001	0.002	0.001
	Approx.	0.007	0.002	0.004	0.002	0.004	0.001	0.003	0.001	0.003	0.001	0.002	0.001	0.002	0.001
	LoG	0.007	0.002	0.004	0.002	0.004	0.001	0.003	0.001	0.003	0.001	0.002	0.001	0.002	0.001
	Conv.	0.007	0.002	0.007	0.002	0.004	0.001	0.004	0.001	0.005	0.001	0.002	0.001	0.003	0.001

Table G.1: Proof of concept results. Subblock 1.0

Encoder magnet: sphere. Maximum and mean angular errors at z-distance = 2 mm.

ADC resolution [bit]	**Array size:** / **Error [°]:**	3x3		5x5		7x7		9x9		11x11		13x13		15x15	
		max	mean	max	mean	max	mean	max	mean	max	mean	max	mean	max	mean
8	*Ext. 2D-DFT*	22.000	7.530	18.491	6.122	19.590	6.874	19.735	7.430	24.645	8.196	21.386	8.436	23.553	8.605
	Approx.	22.000	7.530	16.000	5.861	16.408	5.996	16.176	6.224	21.244	7.123	18.213	6.976	19.286	7.052
	LoG	22.000	7.529	16.111	5.862	16.462	6.075	16.464	6.386	21.621	7.356	18.899	7.364	20.274	7.549
	Conv.	22.000	7.529	22.438	6.617	28.695	10.631	28.316	11.458	34.821	11.792	30.013	11.703	32.041	11.264
10	*Ext. 2D-DFT*	5.992	1.793	4.289	1.561	3.596	1.004	3.017	1.138	2.480	0.860	2.152	0.873	2.603	0.744
	Approx.	5.992	1.793	4.188	1.544	3.544	0.941	2.727	1.151	2.394	0.819	1.868	0.806	2.375	0.646
	LoG	5.980	1.793	4.176	1.544	3.553	0.944	2.722	1.162	2.414	0.818	1.980	0.839	2.481	0.674
	Conv.	5.980	1.793	5.528	2.085	4.644	1.376	4.154	1.500	2.983	1.069	3.242	1.208	3.160	0.946
12	*Ext. 2D-DFT*	1.674	0.554	1.281	0.398	0.849	0.261	1.070	0.285	0.597	0.188	0.539	0.152	0.505	0.161
	Approx.	1.674	0.554	1.203	0.384	0.712	0.273	0.913	0.277	0.614	0.182	0.515	0.157	0.467	0.165
	LoG	1.673	0.554	1.201	0.383	0.717	0.273	0.910	0.277	0.606	0.185	0.521	0.151	0.474	0.165
	Conv.	1.673	0.554	1.600	0.474	1.083	0.361	1.243	0.381	0.720	0.295	0.596	0.176	0.678	0.184
14	*Ext. 2D-DFT*	0.386	0.125	0.288	0.098	0.234	0.063	0.220	0.076	0.193	0.057	0.134	0.053	0.162	0.039
	Approx.	0.386	0.125	0.289	0.098	0.249	0.064	0.217	0.073	0.175	0.053	0.132	0.050	0.147	0.040
	LoG	0.386	0.125	0.288	0.098	0.248	0.063	0.213	0.073	0.178	0.053	0.131	0.051	0.143	0.040
	Conv.	0.386	0.125	0.412	0.157	0.249	0.087	0.279	0.089	0.256	0.073	0.156	0.067	0.157	0.041
16	*Ext. 2D-DFT*	0.079	0.026	0.066	0.017	0.045	0.016	0.055	0.015	0.036	0.013	0.043	0.014	0.035	0.011
	Approx.	0.079	0.026	0.062	0.019	0.053	0.016	0.055	0.015	0.036	0.012	0.047	0.012	0.029	0.011
	LoG	0.079	0.026	0.062	0.019	0.053	0.016	0.055	0.015	0.035	0.013	0.048	0.012	0.029	0.011
	Conv.	0.079	0.026	0.083	0.034	0.045	0.019	0.066	0.017	0.039	0.018	0.051	0.016	0.040	0.013

Table G.2: Proof of concept results. Subblock 1.0

Encoder magnet: sphere. Maximum and mean angular errors at z-distance = 5 mm.

ADC resolution [bit]	**Array size:** / **Error [°]:**	3x3		5x5		7x7		9x9		11x11		13x13		15x15	
		max	mean	max	mean	max	mean	max	mean	max	mean	max	mean	max	mean
8	*Ext. 2D-DFT*	4.367	1.324	2.758	0.920	1.906	0.743	2.197	0.647	1.409	0.593	1.654	0.617	2.133	0.569
	Approx.	4.367	1.324	2.614	0.952	1.879	0.749	2.140	0.668	1.457	0.530	1.491	0.548	1.950	0.511
	LoG	4.371	1.323	2.617	0.953	1.912	0.749	2.155	0.669	1.500	0.541	1.505	0.554	2.083	0.529
	Conv.	4.371	1.323	3.260	1.503	3.183	1.032	3.153	1.041	2.157	0.929	2.437	0.823	2.769	0.768
10	*Ext. 2D-DFT*	1.172	0.377	0.966	0.250	0.624	0.209	0.530	0.171	0.347	0.150	0.405	0.145	0.269	0.112
	Approx.	1.172	0.377	0.885	0.258	0.502	0.196	0.544	0.167	0.352	0.147	0.468	0.131	0.286	0.104
	LoG	1.171	0.377	0.884	0.257	0.492	0.196	0.541	0.168	0.340	0.148	0.466	0.133	0.318	0.109
	Conv.	1.171	0.377	1.303	0.368	1.055	0.280	0.708	0.244	0.515	0.190	0.562	0.193	0.388	0.153
12	*Ext. 2D-DFT*	0.258	0.080	0.162	0.052	0.154	0.060	0.135	0.049	0.120	0.037	0.115	0.030	0.094	0.034
	Approx.	0.258	0.080	0.169	0.052	0.137	0.059	0.122	0.049	0.120	0.035	0.089	0.030	0.096	0.034
	LoG	0.258	0.080	0.167	0.051	0.138	0.059	0.123	0.049	0.120	0.034	0.096	0.030	0.091	0.034
	Conv.	0.258	0.080	0.228	0.095	0.190	0.062	0.179	0.063	0.122	0.045	0.177	0.039	0.129	0.039
14	*Ext. 2D-DFT*	0.064	0.023	0.039	0.015	0.036	0.013	0.029	0.013	0.028	0.009	0.038	0.009	0.018	0.006
	Approx.	0.064	0.023	0.043	0.015	0.035	0.013	0.031	0.012	0.024	0.009	0.039	0.009	0.019	0.007
	LoG	0.064	0.023	0.043	0.015	0.035	0.013	0.032	0.012	0.024	0.009	0.039	0.009	0.018	0.007
	Conv.	0.064	0.023	0.053	0.022	0.053	0.017	0.051	0.016	0.036	0.010	0.038	0.010	0.025	0.008
16	*Ext. 2D-DFT*	0.016	0.006	0.014	0.004	0.011	0.004	0.011	0.003	0.009	0.002	0.006	0.003	0.005	0.002
	Approx.	0.016	0.006	0.012	0.003	0.011	0.004	0.012	0.003	0.008	0.002	0.006	0.002	0.006	0.002
	LoG	0.016	0.006	0.013	0.003	0.011	0.004	0.012	0.003	0.008	0.002	0.006	0.002	0.006	0.002
	Conv.	0.016	0.006	0.018	0.005	0.014	0.005	0.012	0.003	0.009	0.003	0.008	0.003	0.007	0.002

Table G.3: Proof of concept results. Subblock 1.0
Encoder magnet: disc. Maximum and mean angular errors at z-distance = 2 mm.

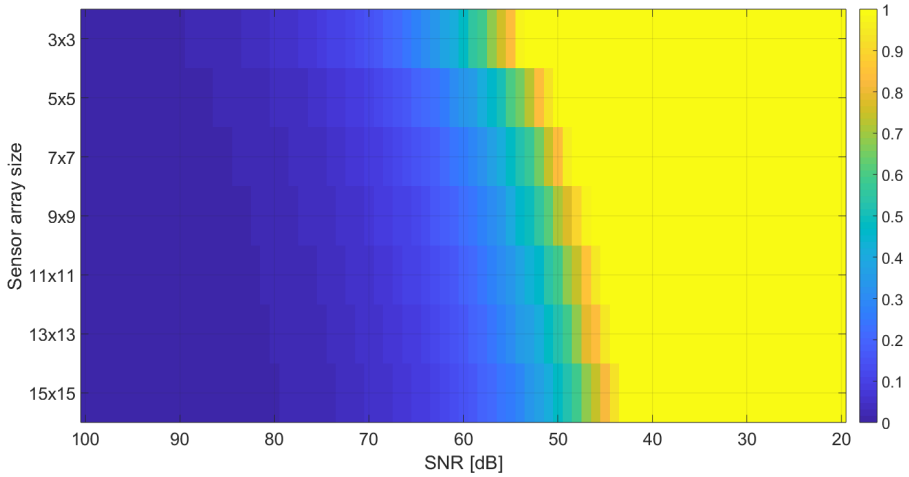
ADC resolution [bit]	**Array size:** / **Error [°]:**	3x3		5x5		7x7		9x9		11x11		13x13		15x15	
		max	mean	max	mean	max	mean	max	mean	max	mean	max	mean	max	mean
8	*Ext. 2D-DFT*	22.435	8.134	16.097	6.595	21.190	7.275	16.504	6.763	19.842	7.242	20.118	7.466	20.558	7.654
	Approx.	22.435	8.134	15.241	6.156	19.945	6.633	15.471	5.912	18.235	6.207	17.902	6.357	18.799	6.354
	LoG	22.435	8.132	15.268	6.170	20.012	6.678	15.529	6.017	18.620	6.436	18.378	6.712	19.434	6.835
	Conv.	22.435	8.132	16.549	5.604	24.623	10.532	23.803	10.497	28.689	10.667	26.477	10.433	27.416	10.187
10	*Ext. 2D-DFT*	5.524	1.705	3.887	1.266	3.454	0.918	3.064	0.878	2.324	0.935	2.161	0.667	1.975	0.662
	Approx.	5.524	1.705	3.912	1.244	3.499	0.827	2.770	0.817	2.318	0.885	2.205	0.597	1.807	0.570
	LoG	5.517	1.702	3.885	1.240	3.537	0.823	2.782	0.821	2.320	0.892	2.188	0.596	1.886	0.612
	Conv.	5.517	1.702	5.169	1.406	3.596	1.192	3.232	1.097	2.711	1.154	2.559	0.912	2.397	0.891
12	*Ext. 2D-DFT*	1.524	0.467	1.065	0.400	0.763	0.272	0.834	0.236	0.506	0.193	0.592	0.195	0.459	0.137
	Approx.	1.524	0.467	1.068	0.397	0.739	0.270	0.764	0.224	0.521	0.185	0.553	0.186	0.453	0.127
	LoG	1.522	0.467	1.071	0.398	0.743	0.269	0.765	0.226	0.521	0.185	0.545	0.186	0.456	0.128
	Conv.	1.522	0.467	1.560	0.567	0.941	0.301	0.929	0.329	0.716	0.214	0.588	0.220	0.452	0.179
14	*Ext. 2D-DFT*	0.292	0.106	0.282	0.075	0.196	0.055	0.154	0.051	0.176	0.047	0.146	0.038	0.093	0.032
	Approx.	0.292	0.106	0.275	0.076	0.191	0.051	0.147	0.051	0.193	0.047	0.125	0.034	0.100	0.033
	LoG	0.291	0.106	0.275	0.076	0.190	0.050	0.150	0.051	0.191	0.047	0.127	0.035	0.096	0.033
	Conv.	0.291	0.106	0.266	0.109	0.223	0.073	0.227	0.067	0.138	0.054	0.162	0.048	0.100	0.040
16	*Ext. 2D-DFT*	0.093	0.032	0.061	0.023	0.038	0.014	0.039	0.014	0.045	0.014	0.036	0.011	0.029	0.009
	Approx.	0.093	0.032	0.055	0.022	0.039	0.014	0.044	0.014	0.042	0.013	0.041	0.011	0.028	0.008
	LoG	0.092	0.032	0.055	0.022	0.040	0.014	0.043	0.013	0.042	0.013	0.040	0.011	0.027	0.008
	Conv.	0.092	0.032	0.082	0.032	0.060	0.019	0.049	0.017	0.051	0.018	0.045	0.013	0.031	0.011

Table G.4: Proof of concept results. Subblock 1.0

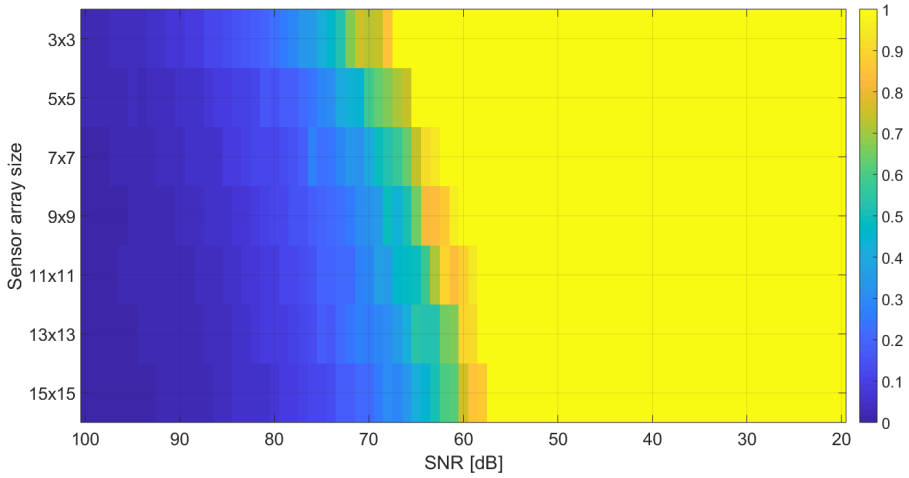
Encoder magnet: disc. Maximum and mean angular errors at z-distance = 5 mm.

G.5.2 Subblock 1.1

Magnet shape: sphere

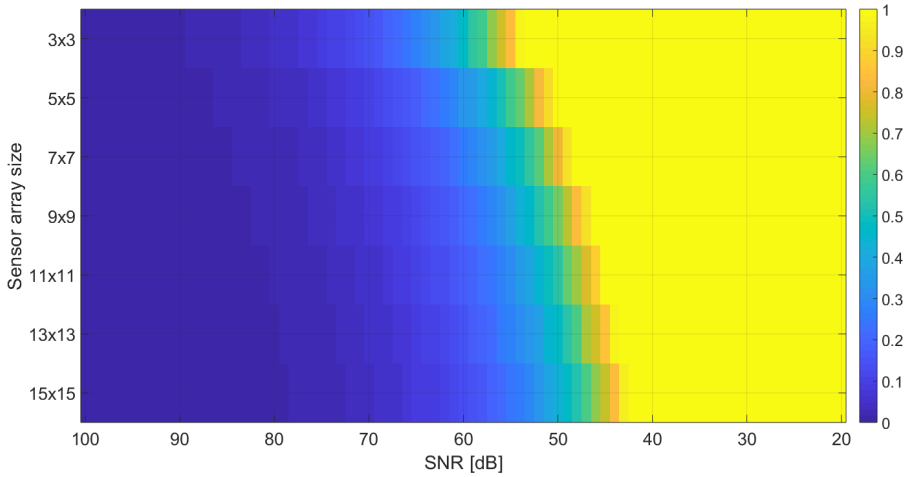


(a) Mean angular error.

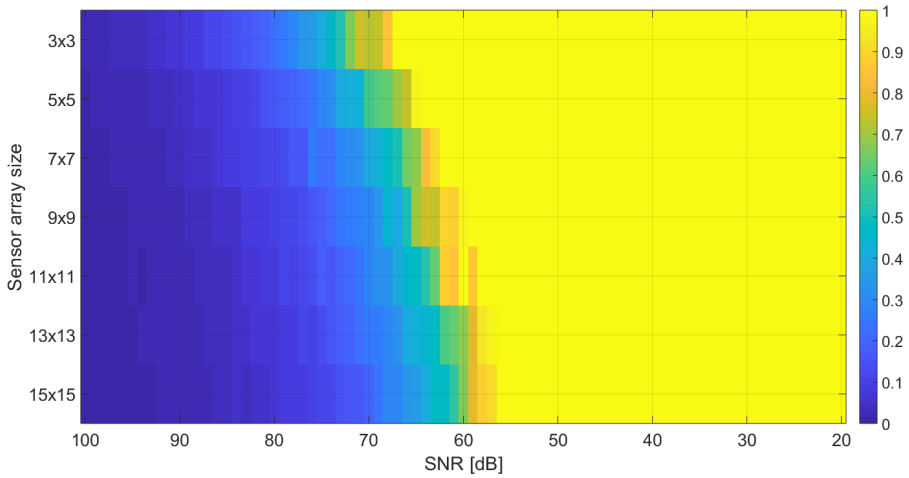


(b) Maximum angular error.

Figure G.7: 1° border for mean and maximum angular errors of extended 2D-DFT algorithm for angle information extraction. Encoder magnet: sphere. 5 mm z-distance.

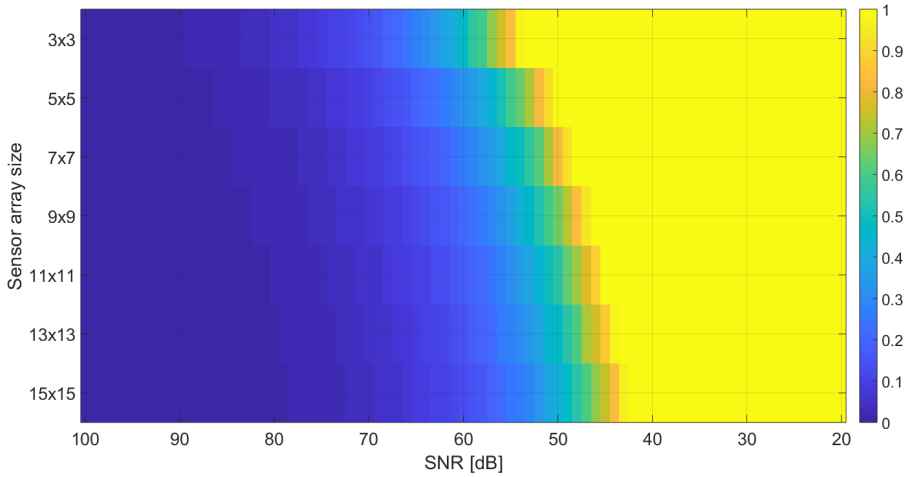


(a) Mean angular error.

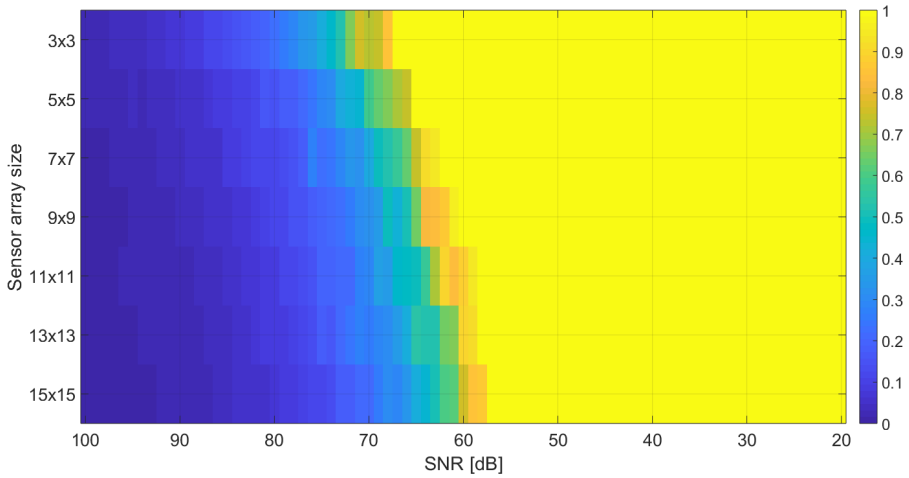


(b) Maximum angular error.

Figure G.8: 1° border for mean and maximum angular errors of approximation based algorithm for angle information extraction. Encoder magnet: sphere. 5 mm z-distance.

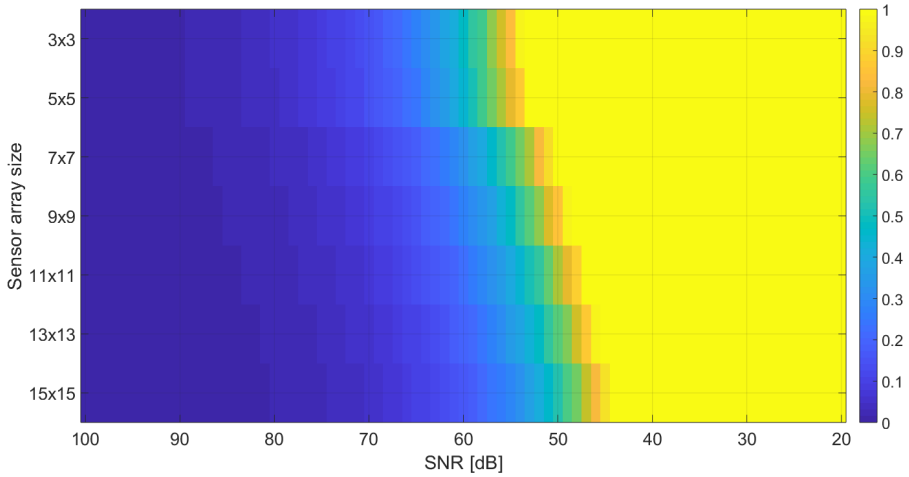


(a) Mean angular error.

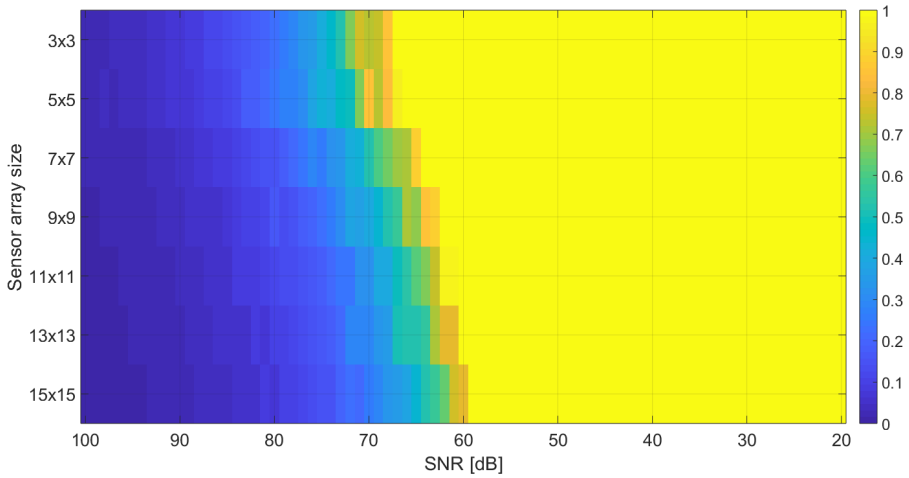


(b) Maximum angular error.

Figure G.9: 1° border for mean and maximum angular errors of single LoG algorithm for angle information extraction. Encoder magnet: sphere. 5 mm z-distance.



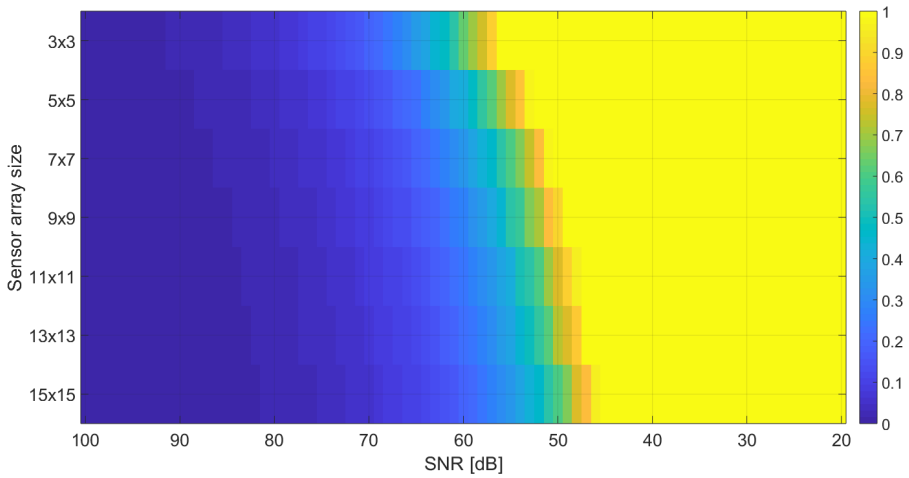
(a) Mean angular error.



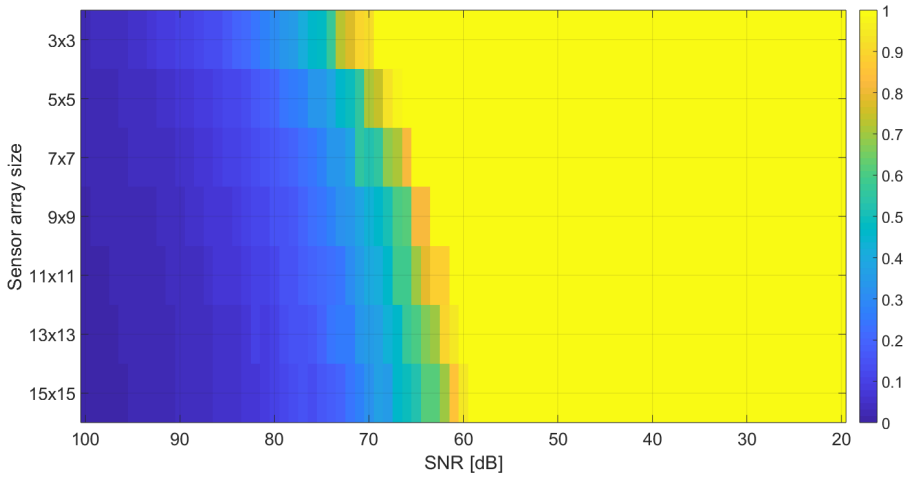
(b) Maximum angular error.

Figure G.10: 1° border for mean and maximum angular errors of LoG convolution for angle information extraction. Encoder magnet: sphere. 5 mm z-distance.

Magnet shape: disc

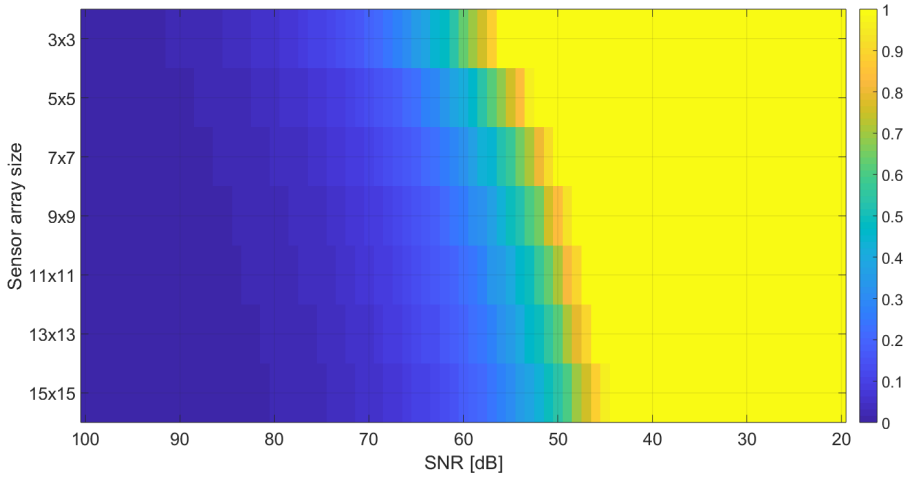


(a) Mean angular error.

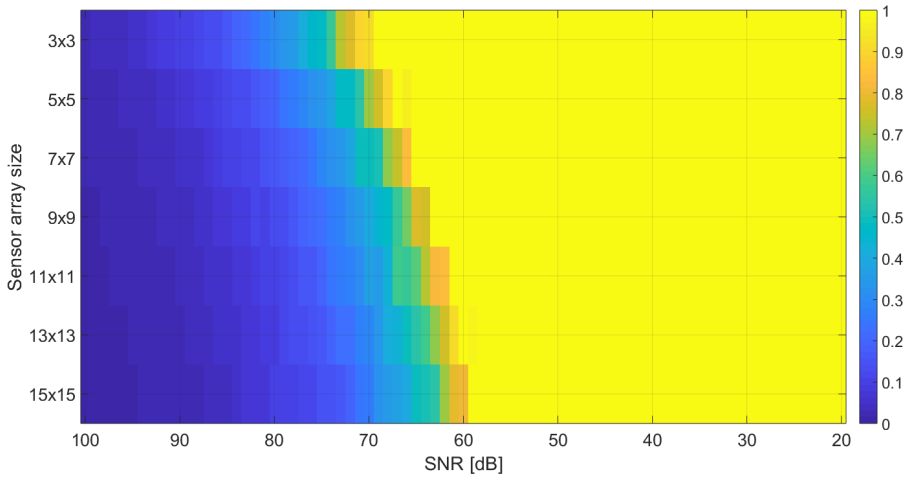


(b) Maximum angular error.

Figure G.11: 1° border for mean and maximum angular errors of extended 2D-DFT algorithm for angle information extraction. Encoder magnet: disc. 5 mm z-distance.

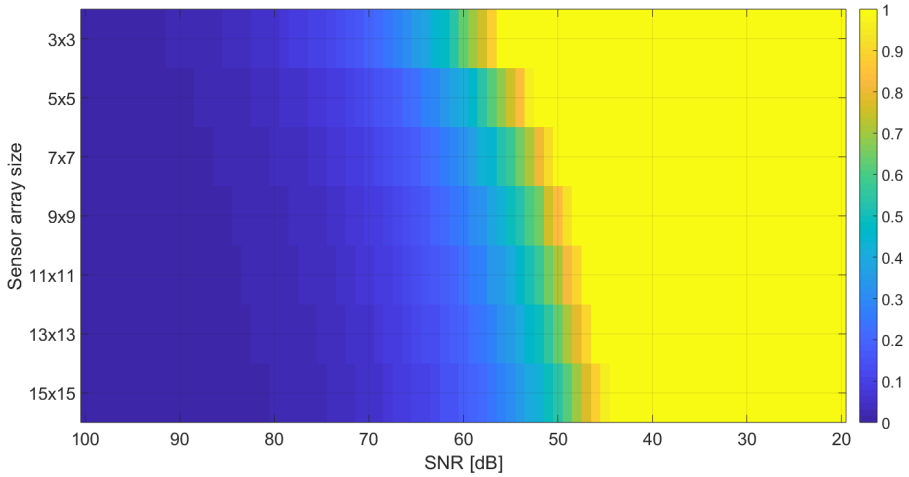


(a) .Mean angular error.

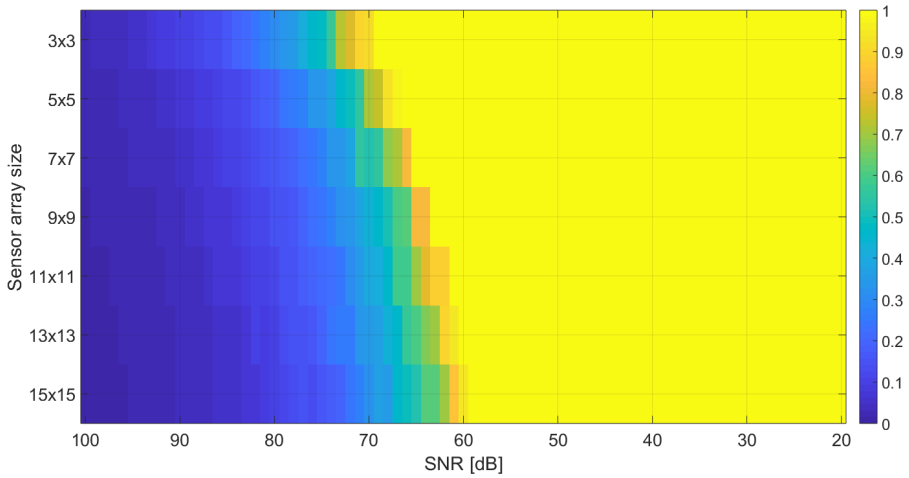


(b) .Maximum angular error.

Figure G.12: 1° border for mean and maximum angular errors of approximation based algorithm for angle information extraction. Encoder magnet: disc. 5 mm z-distance.

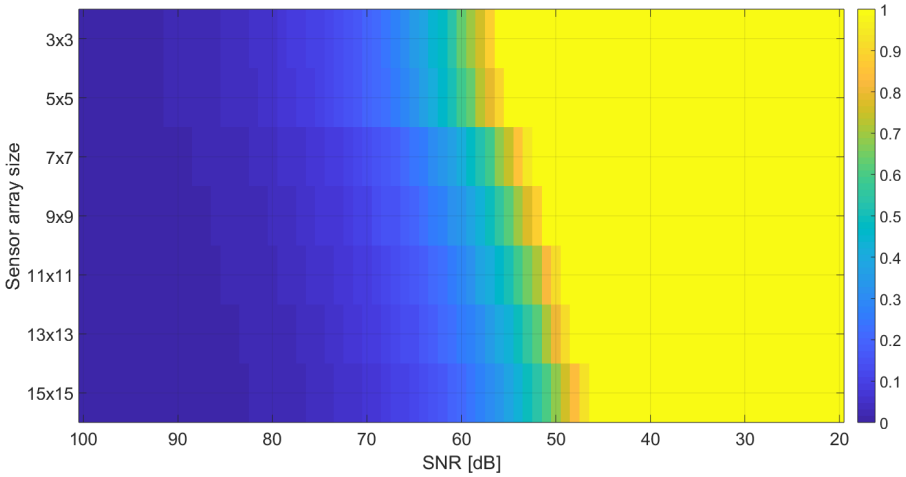


(a) Mean angular error.

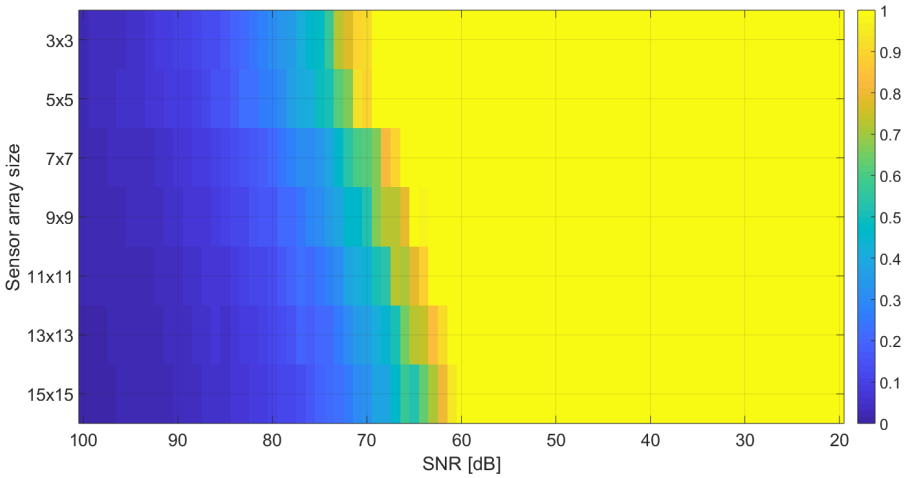


(b) Maximum angular error.

Figure G.13: 1° border for mean and maximum angular errors of single LoG algorithm for angle information extraction. Encoder magnet: disc. 5 mm z-distance.



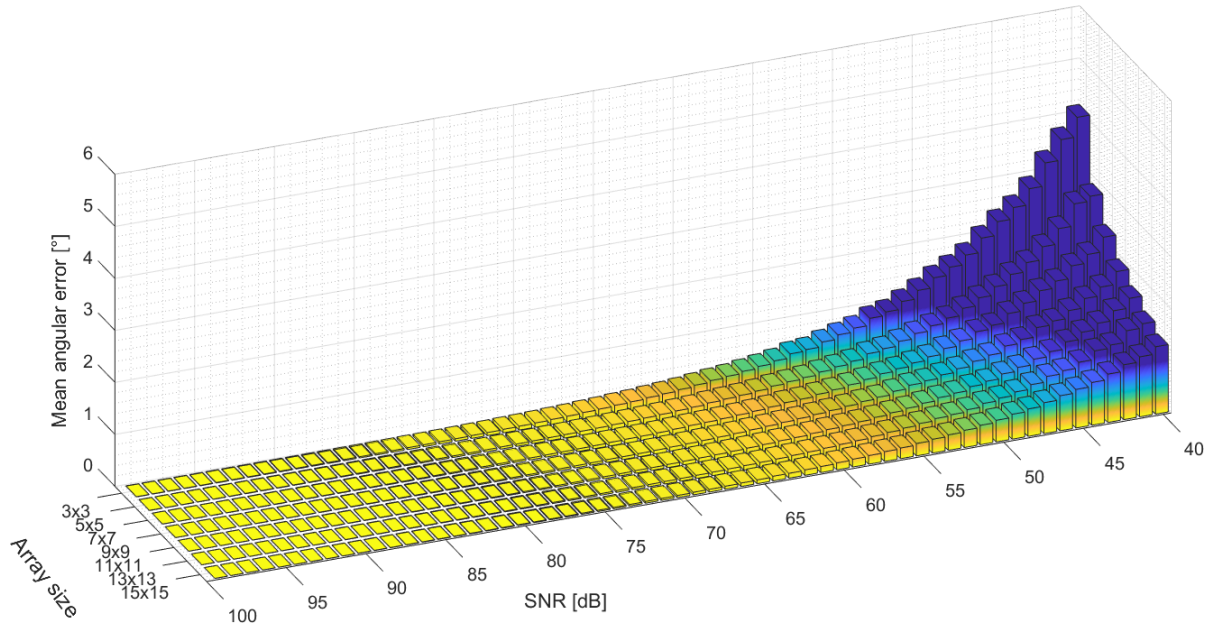
(a) Mean angular error.



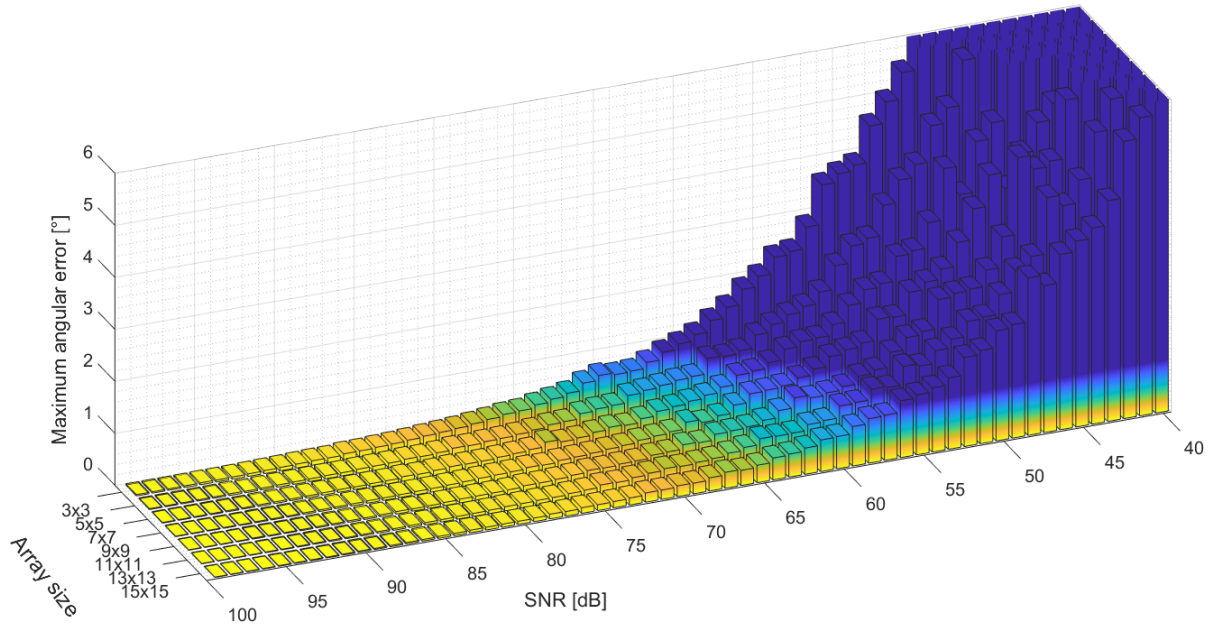
(b) Maximum angular error.

Figure G.14: 1° border for mean and maximum angular errors of LoG convolution for angle information extraction. Encoder magnet: disc. 5 mm z-distance.

Bar plots for complete angular error visualization, comparison for subblock 1.1. Magnet shape: sphere

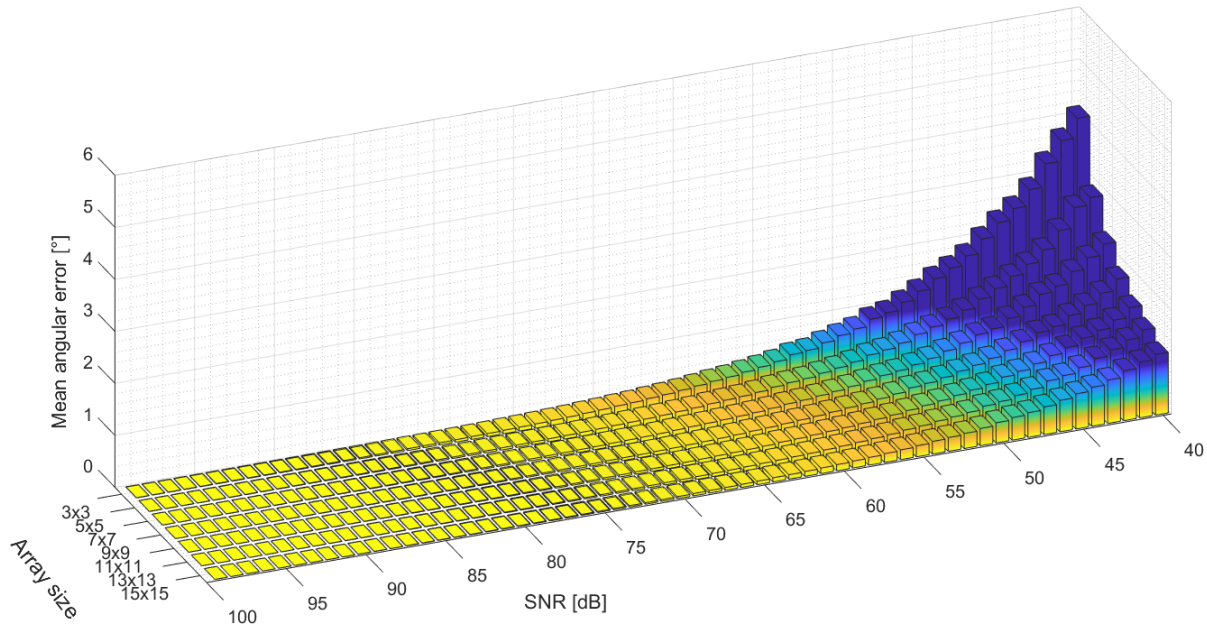


(a) Mean angular error.

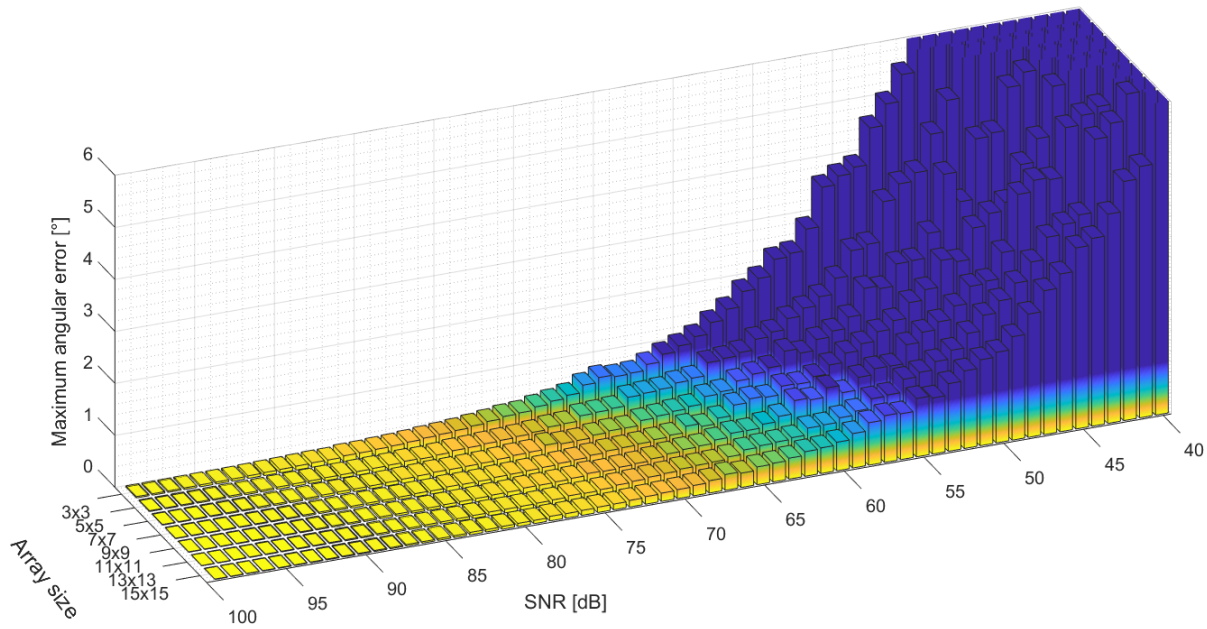


(b) Maximum angular error.

Figure G.15: Bar plot for mean and maximum angular errors of extended 2D-DFT algorithm for angle information extraction. Encoder magnet: disc. 5 mm z-distance.

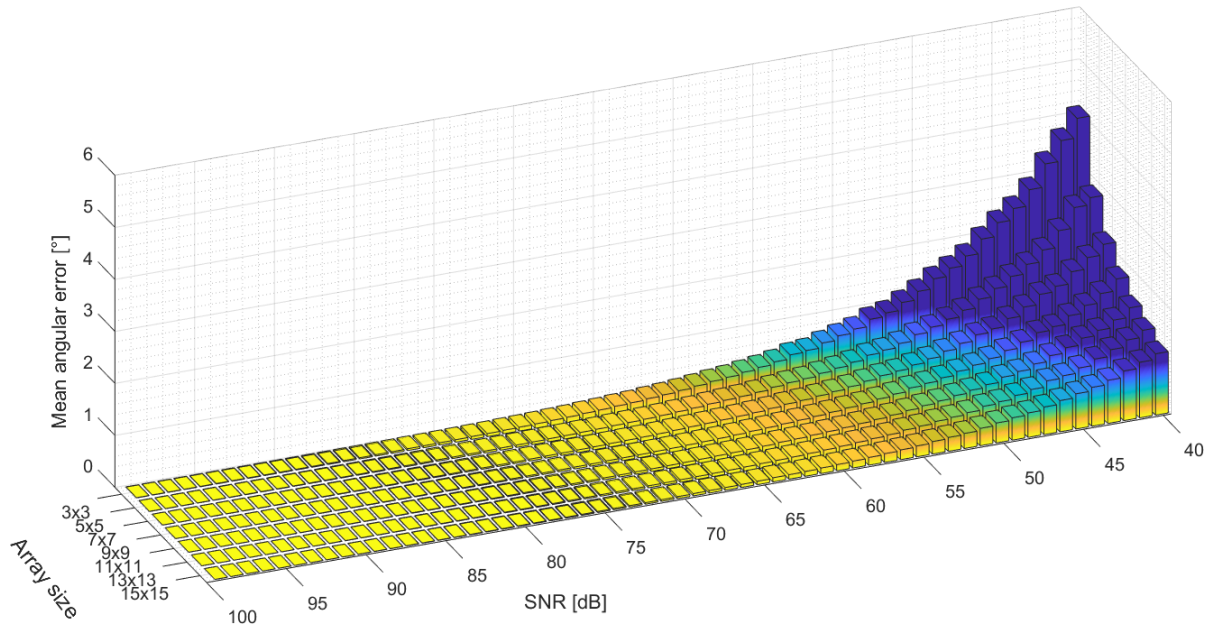


(a) Mean angular error.

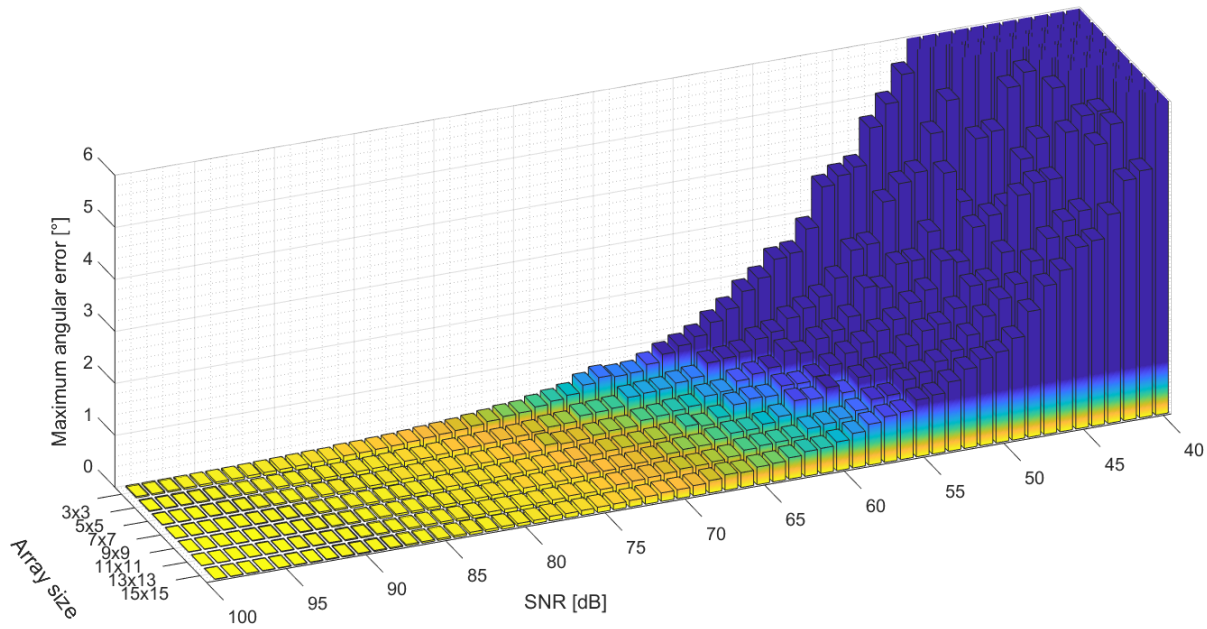


(b) Maximum angular error.

Figure G.16: Bar plot for mean and maximum angular errors of single LoG algorithm for angle information extraction. Encoder magnet: disc. 5 mm z-distance.



(a) Mean angular error.



(b) Maximum angular error.

Figure G.17: Bar plot for mean and maximum angular errors of approximation based algorithm for angle information extraction. Encoder magnet: disc. 5 mm z-distance.

G.5.3 Subblock 1.2

ADC resolution [bit]	**Array size:** / **Error [°]:**	3x3		5x5		7x7		9x9		11x11		13x13		15x15	
		max	mean	max	mean	max	mean	max	mean	max	mean	max	mean	max	mean
8	*Ext. 2D-DFT*	63.435	23.176	44.888	16.250	23.804	8.396	22.223	9.046	20.546	6.945	18.014	7.412	20.516	7.947
	Approx.	63.435	23.176	43.111	16.397	24.118	8.590	20.002	7.990	18.745	6.750	16.079	6.449	16.909	7.145
	LoG	63.435	23.167	43.146	16.382	23.913	8.593	19.937	7.988	18.660	6.748	16.421	6.644	16.732	7.128
	Conv.	63.435	23.167	46.694	13.712	25.560	9.433	22.466	8.428	25.353	8.960	20.649	9.126	17.742	7.340
10	*Ext. 2D-DFT*	36.393	20.135	18.750	9.032	21.386	11.489	18.401	9.879	14.042	7.403	10.967	5.366	9.845	5.324
	Approx.	36.393	20.135	19.720	8.867	16.578	8.747	18.462	10.184	10.687	5.439	9.994	5.350	9.903	5.325
	LoG	36.439	20.159	19.563	8.805	16.819	8.894	18.794	10.385	10.764	5.525	9.681	5.109	9.552	5.177
	Conv.	36.439	20.159	42.543	23.016	30.787	17.259	28.486	16.023	16.795	9.575	14.534	7.393	13.218	6.958
12	*Ext. 2D-DFT*	38.021	22.477	18.778	11.219	23.606	14.077	13.435	7.681	9.235	5.506	10.705	6.590	9.807	5.263
	Approx.	38.021	22.477	21.412	12.810	21.520	13.071	12.764	7.555	10.099	6.229	12.382	7.555	9.713	6.000
	LoG	38.012	22.471	21.413	12.805	21.473	13.036	12.694	7.524	9.912	6.074	12.205	7.495	8.961	5.527
	Conv.	38.012	22.471	36.454	22.098	24.459	13.984	14.115	8.442	13.147	7.566	14.169	8.427	9.332	5.027
14	*Ext. 2D-DFT*	31.877	19.613	25.650	15.745	13.907	8.068	17.375	10.293	13.353	8.144	9.034	5.331	9.825	6.098
	Approx.	31.877	19.613	21.917	13.400	14.167	8.032	18.380	11.085	14.610	9.063	7.938	4.808	9.390	5.878
	LoG	31.924	19.641	22.004	13.463	14.168	8.014	18.017	10.891	14.546	9.035	7.559	4.565	9.222	5.764
	Conv.	31.924	19.641	44.844	27.045	16.147	8.721	14.115	8.597	13.396	8.211	12.174	6.906	12.152	7.240
16	*Ext. 2D-DFT*	22.208	13.696	24.664	15.192	18.171	10.298	8.870	5.381	13.069	8.052	9.482	5.838	8.057	4.565
	Approx.	22.208	13.696	24.148	14.879	18.090	9.809	8.510	4.985	12.209	7.387	10.210	6.299	7.365	4.079
	LoG	22.081	13.615	23.979	14.777	18.001	9.731	8.318	4.871	12.170	7.422	10.010	6.215	7.230	3.985
	Conv.	22.081	13.615	21.533	13.211	26.086	14.431	11.820	6.707	14.919	9.148	10.834	6.642	7.998	4.806

Table G.5: Proof of concept results. Subblock 1.2

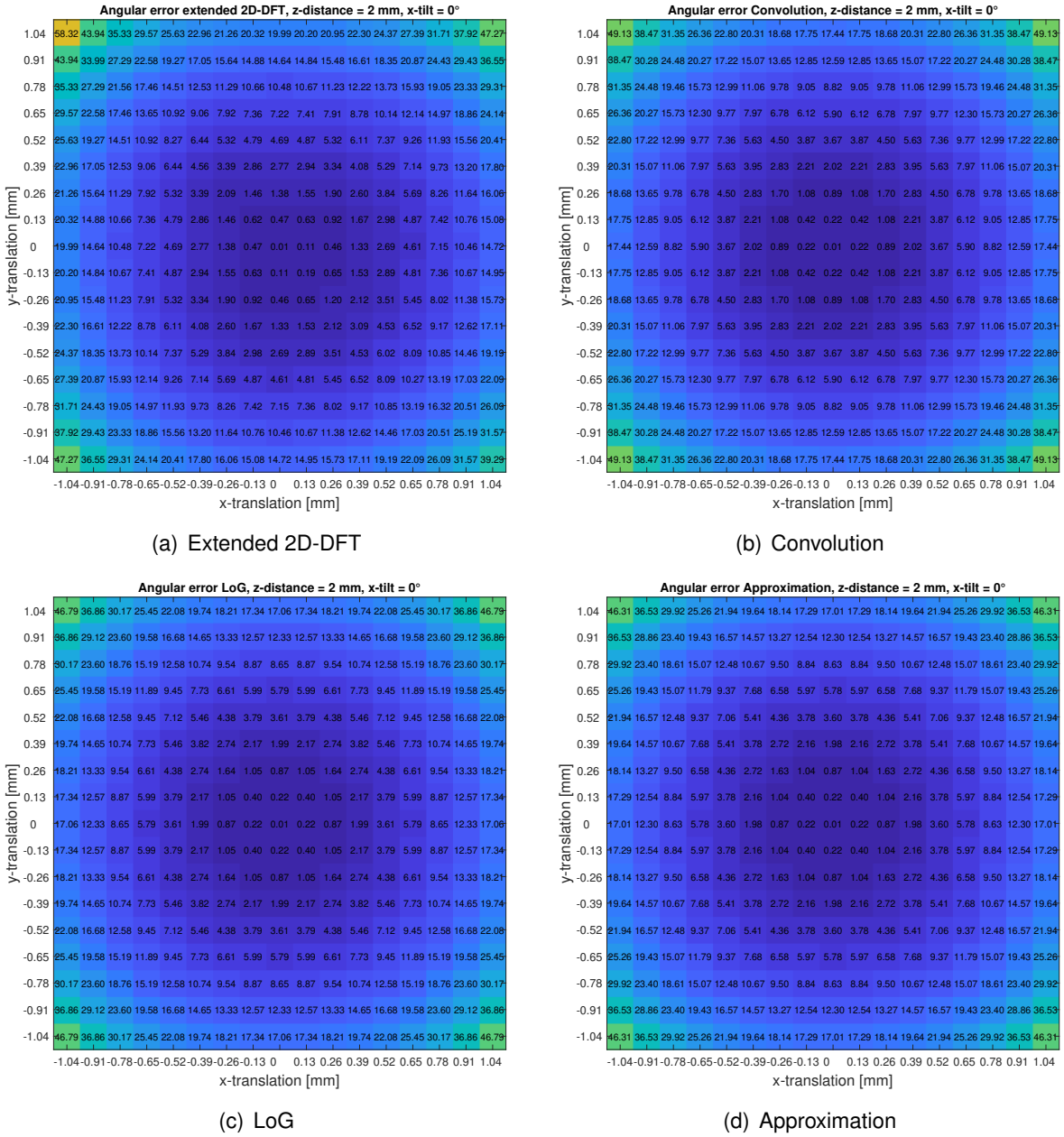
Encoder magnet: sphere. Maximum and mean angular errors at z-distance = 5 mm.

ADC resolution [bit]	**Array size:** / **Error [°]:**	3x3		5x5		7x7		9x9		11x11		13x13		15x15	
		max	mean	max	mean	max	mean	max	mean	max	mean	max	mean	max	mean
8	*Ext. 2D-DFT*	61.462	17.973	28.730	11.694	24.517	9.027	20.688	8.268	19.514	9.009	16.800	7.641	14.388	4.112
	Approx.	61.462	17.973	28.386	11.566	20.618	6.564	19.215	7.795	17.808	8.108	14.187	5.391	11.029	3.645
	LoG	61.416	17.980	28.380	11.562	20.455	6.676	19.157	7.734	17.962	8.110	13.839	5.634	11.331	3.774
	Conv.	61.416	17.980	44.359	13.740	26.312	10.440	24.428	7.466	23.024	9.371	20.117	9.096	18.760	5.597
10	*Ext. 2D-DFT*	39.601	19.887	19.719	10.779	16.688	8.802	18.143	9.199	8.741	4.329	9.060	4.381	11.212	5.863
	Approx.	39.601	19.887	18.962	10.231	13.753	7.067	15.808	8.162	10.351	5.723	9.481	4.881	11.640	5.817
	LoG	39.604	19.886	19.038	10.286	13.893	7.155	15.789	8.114	10.278	5.564	9.317	4.691	11.510	5.784
	Conv.	39.604	19.886	26.933	14.416	22.609	11.579	20.313	10.667	11.521	6.061	13.264	7.136	11.453	6.181
12	*Ext. 2D-DFT*	30.961	17.378	17.307	10.084	13.841	8.371	14.261	8.596	10.095	6.189	8.554	5.236	7.368	4.089
	Approx.	30.961	17.378	18.725	11.022	11.421	6.860	13.344	8.074	10.425	6.362	7.282	4.268	6.402	3.594
	LoG	30.956	17.381	18.705	11.006	11.463	6.897	13.202	7.988	10.374	6.340	6.926	3.878	6.714	3.812
	Conv.	30.956	17.381	27.162	15.404	17.577	9.644	13.228	7.700	11.987	7.323	12.021	7.398	11.113	6.703
14	*Ext. 2D-DFT*	35.258	19.472	12.659	6.869	20.218	12.232	11.107	6.646	8.355	4.628	9.133	5.441	5.799	3.258
	Approx.	35.258	19.472	12.144	7.255	19.049	11.609	11.722	7.083	9.210	5.520	6.960	4.201	6.476	3.842
	LoG	35.235	19.464	12.120	7.228	18.927	11.518	11.546	6.946	9.000	5.419	6.771	4.117	6.662	3.918
	Conv.	35.235	19.464	21.963	12.757	14.726	8.437	9.220	5.221	10.628	6.433	11.847	7.185	7.355	4.401
16	*Ext. 2D-DFT*	36.006	20.753	24.544	14.047	14.643	9.026	18.600	11.069	9.925	5.544	10.476	6.188	8.968	5.494
	Approx.	36.006	20.753	24.354	13.958	14.303	8.559	18.830	11.122	11.286	6.562	8.809	5.150	7.967	4.668
	LoG	35.990	20.733	24.338	13.919	14.292	8.552	19.009	11.216	10.916	6.282	8.770	5.115	7.597	4.503
	Conv.	35.990	20.733	24.322	13.932	16.381	9.954	21.383	12.727	15.695	8.691	13.075	7.906	8.826	5.239

Table G.6: Proof of concept results. Subblock 1.2
Encoder magnet: disc. Maximum and mean angular errors at z-distance = 5 mm.

G.5.4 Misalignment test (block 2)

G.5.5 Misalignment robustness (subblock 2.0)



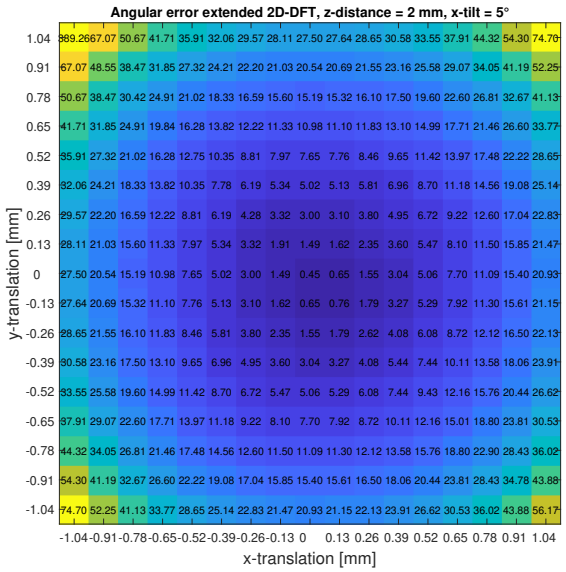
(a) Extended 2D-DFT

(b) Convolution

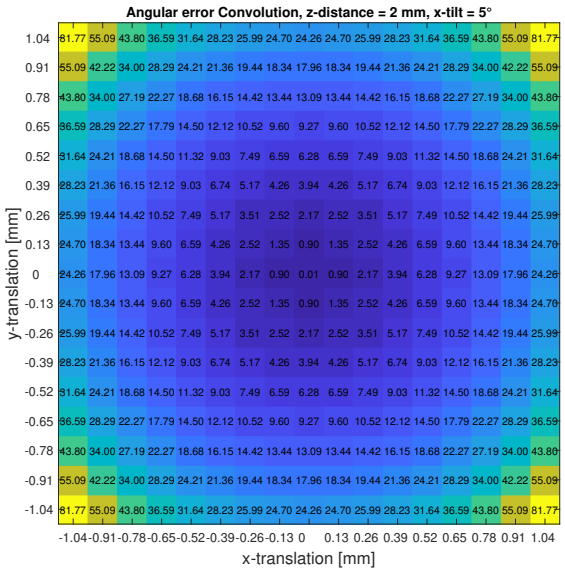
(c) LoG

(d) Approximation

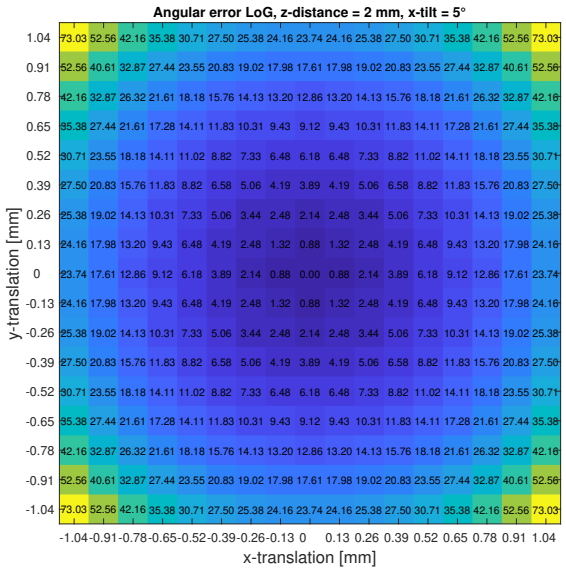
Figure G.18: Maximum angular errors of algorithms at z-distance = 2 mm, x-tilt = 0°.



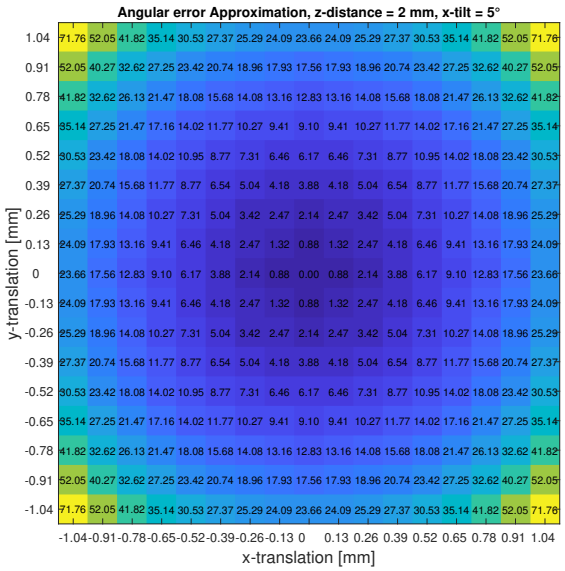
(a) Extended 2D-DFT



(b) Convolution

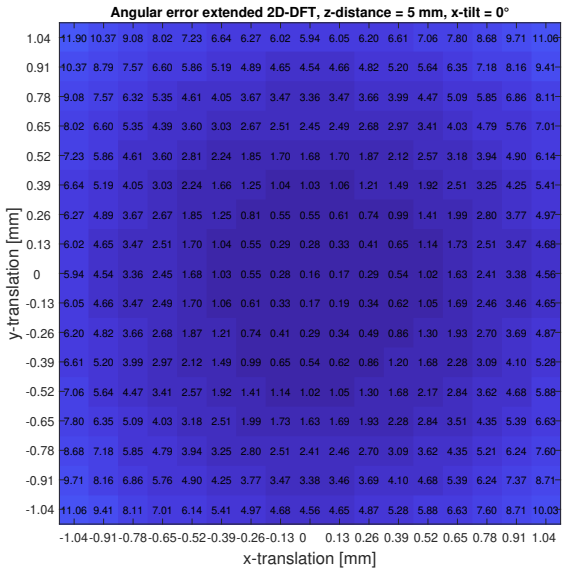


(c) LoG

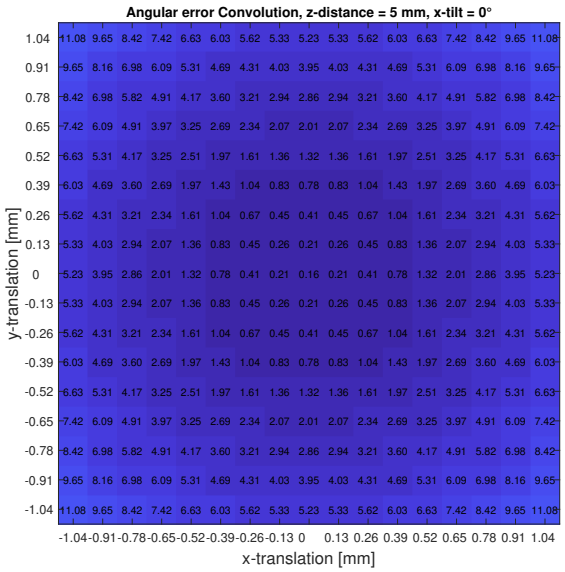


(d) Approximation

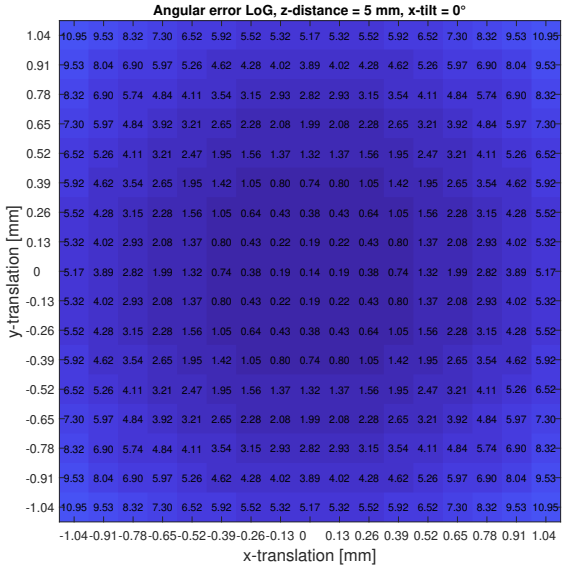
Figure G.19: Maximum angular errors of algorithms at z-distance = 2 mm, x-tilt = 5°.



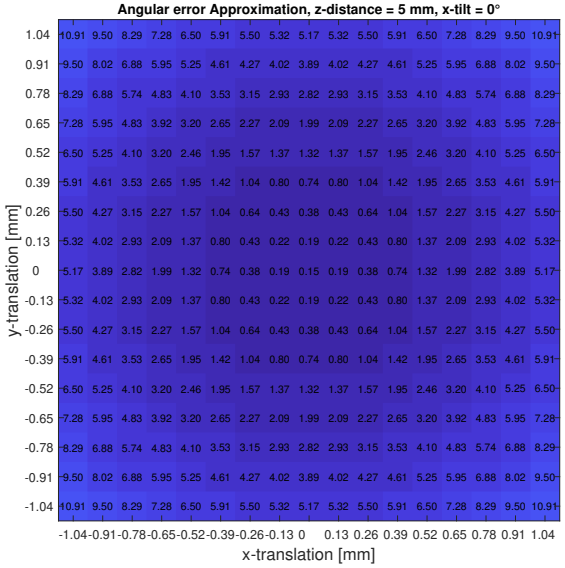
(a) Extended 2D-DFT



(b) Convolution

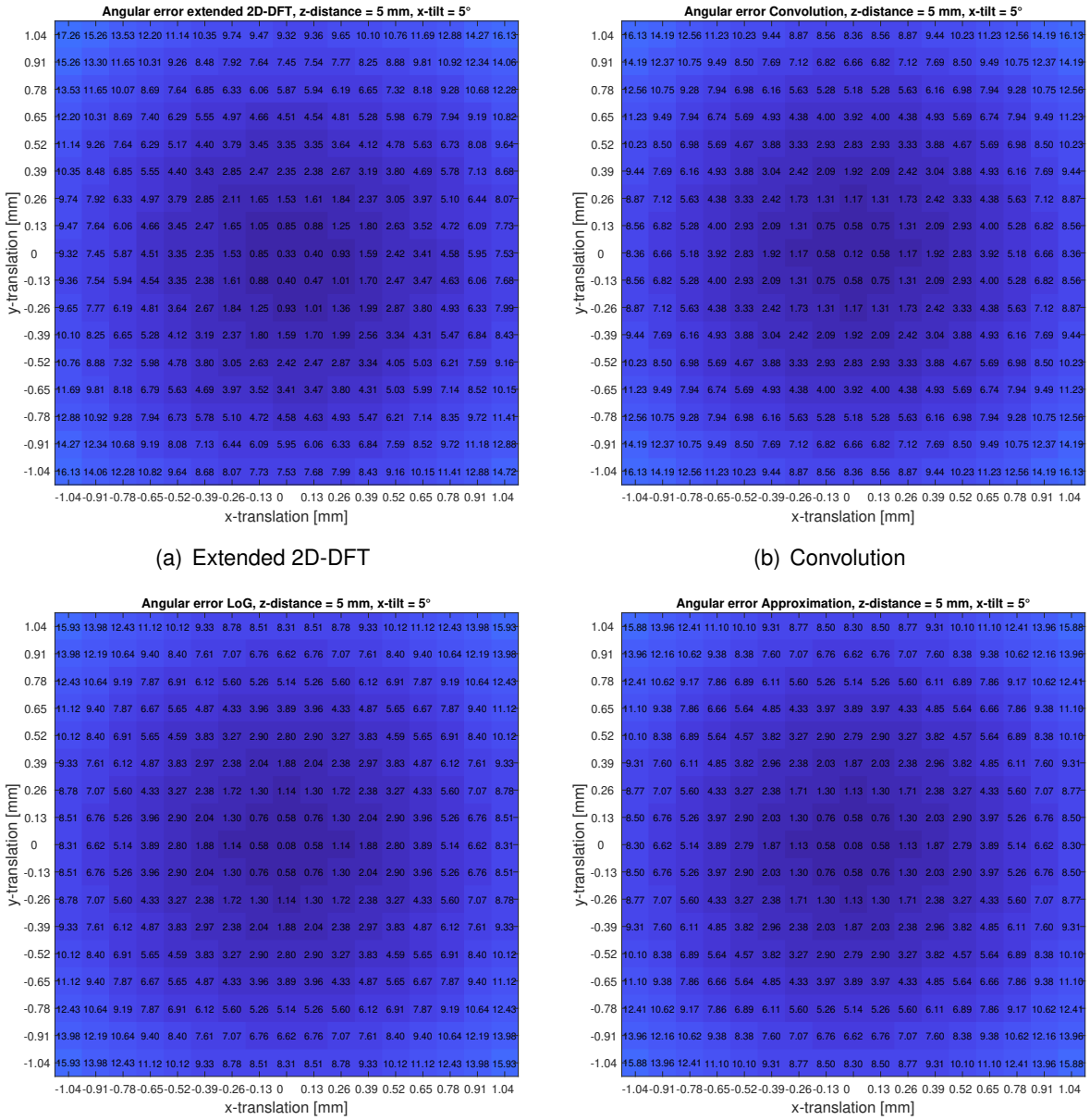


(c) LoG



(d) Approximation

Figure G.20: Maximum angular errors of algorithms at z-distance = 5 mm, x-tilt = 0°.



(a) Extended 2D-DFT

(b) Convolution

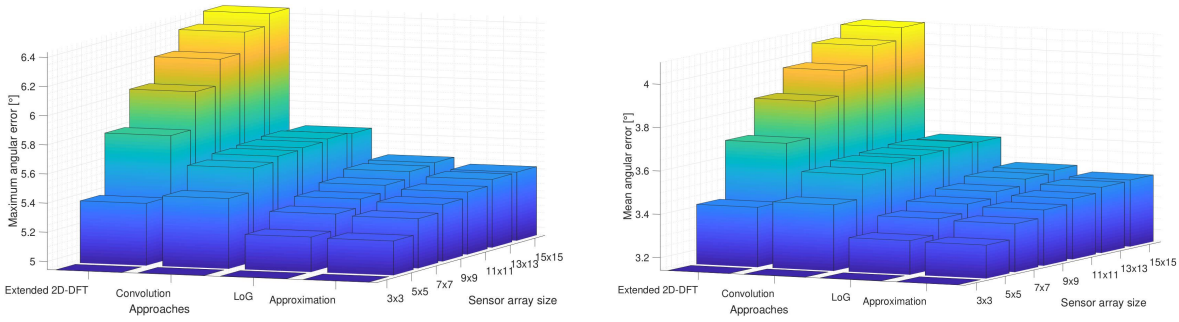
(c) LoG

(d) Approximation

Figure G.21: Maximum angular errors of algorithms at z-distance = 5 mm, x-tilt = 5°.

G.5.6 Array size and misalignment effect dependency (subblock 2.1)

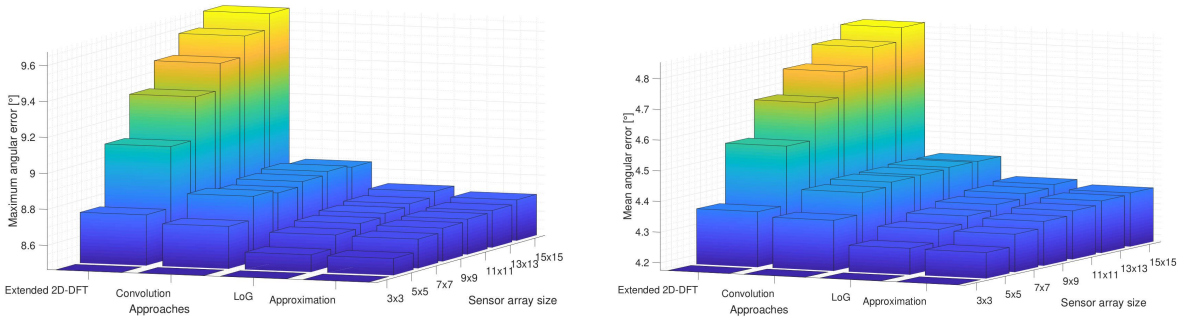
Misalignment case 1



(a) Maximum angular error. (b) Mean angular error.

Figure G.22: Array size and misalignment effect dependency investigation (subblock 2.1): case 1.

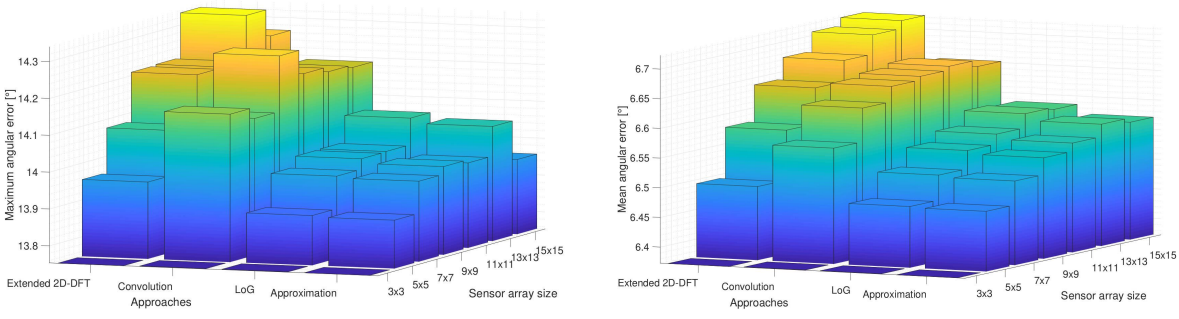
Misalignment case 2



(a) Maximum angular error. (b) Mean angular error.

Figure G.23: Array size and misalignment effect dependency investigation (subblock 2.1): case 2.

Misalignment case 3



(a) Maximum angular error. (b) Mean angular error.

Figure G.24: Array size and misalignment effect dependency investigation (subblock 2.1): case 3.

G.6 Trackings algorithms test (block 3)

G.6.1 Results for disc magnet

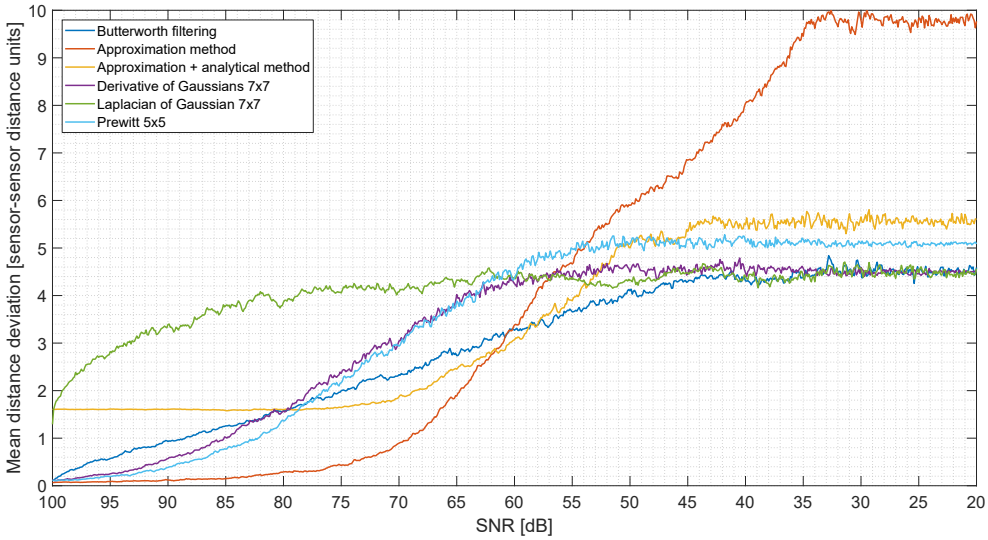


Figure G.25: Mean distance deviation for all localized AMM points for the disc magnet data. LoG accuracy decreases rapidly. LoG and simplified approximation exhibit the highest initial errors. Approximations methods are stable until $\approx$ 75 - 70 dB.

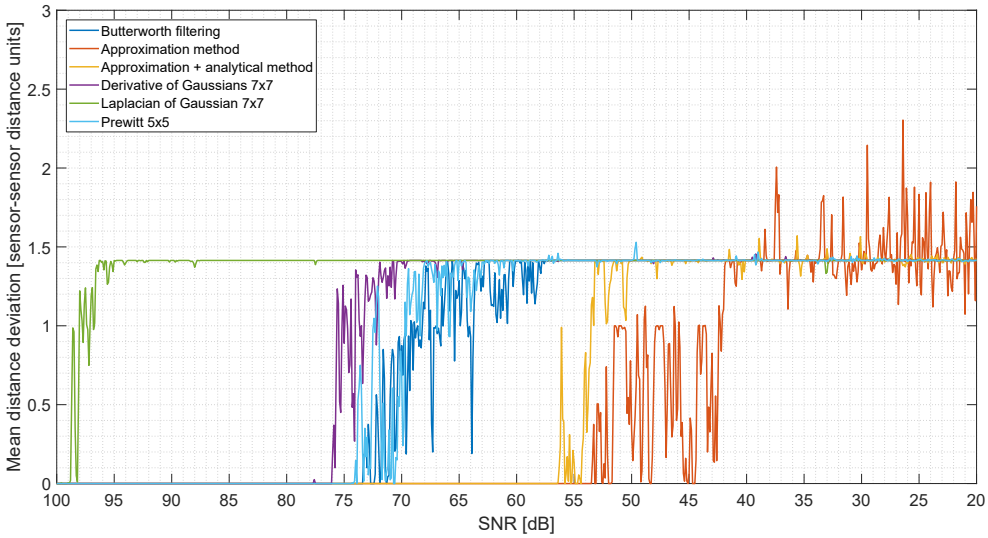
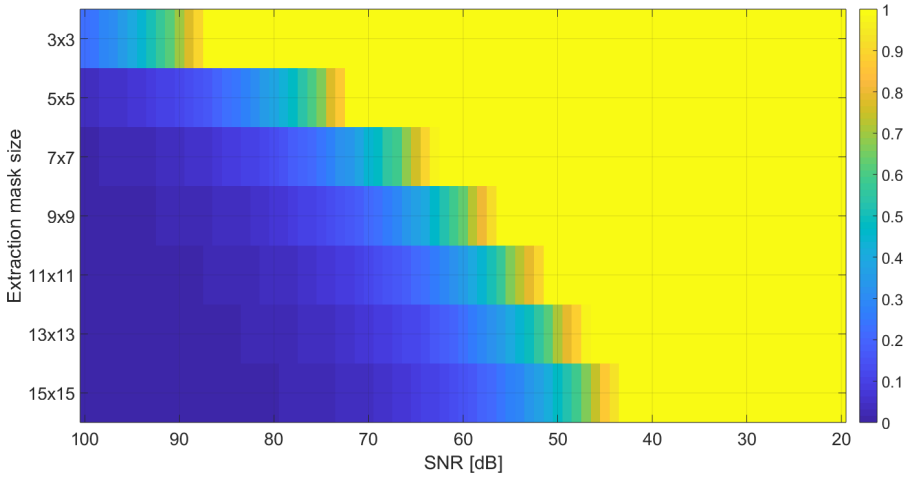


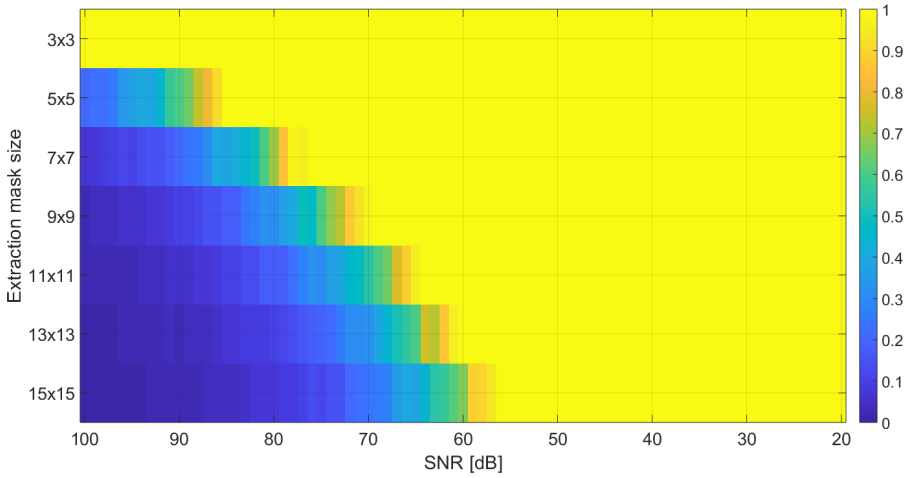
Figure G.26: Mean distance deviation for localized mean AMM point for the disc magnet data. LoG exhibits the worst accuracy, normal approximation the best one. Accuracy limits of Butterworth, LoG, DoG are close to each other. Simplified approximation is the second best.

G.7 Extended algorithms test (block 4)

G.7.1 Extended algorithms test 2 (subblock 4.1)

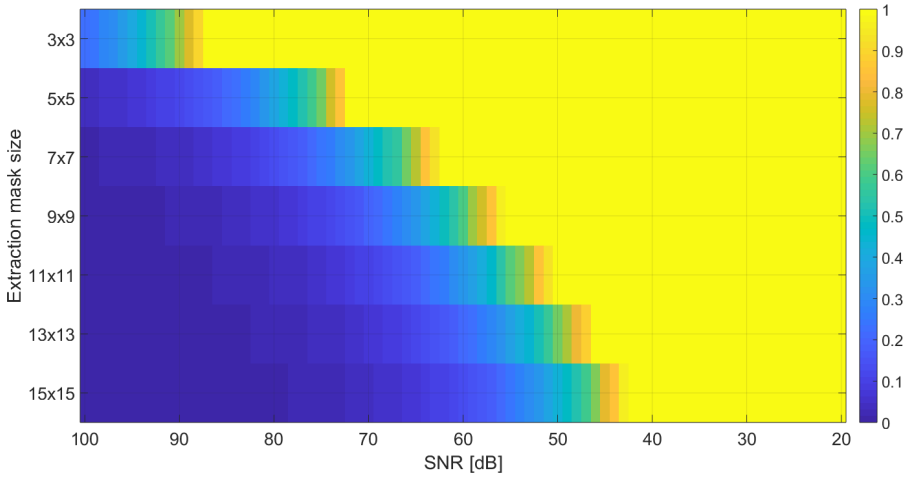


(a) Mean angular error.

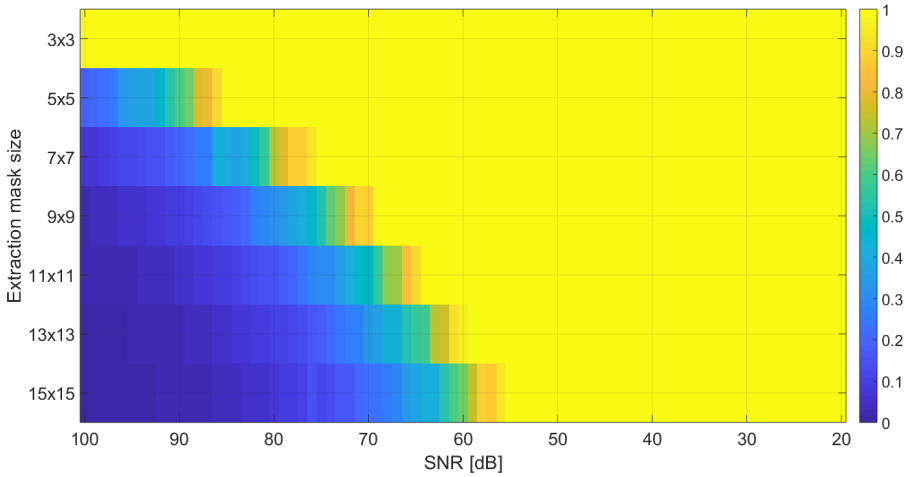


(b) Maximum angular error.

Figure G.27: 1° border for mean and maximum angular errors of extended 2D-DFT algorithm for angle information extraction combined with AMM tracking feature. Encoder magnet: sphere. 5 mm z-distance.

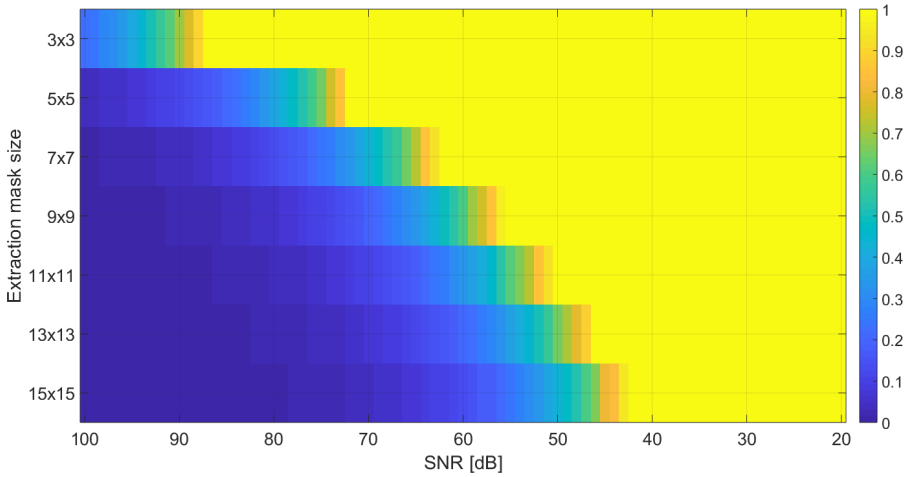


(a) Mean angular error.

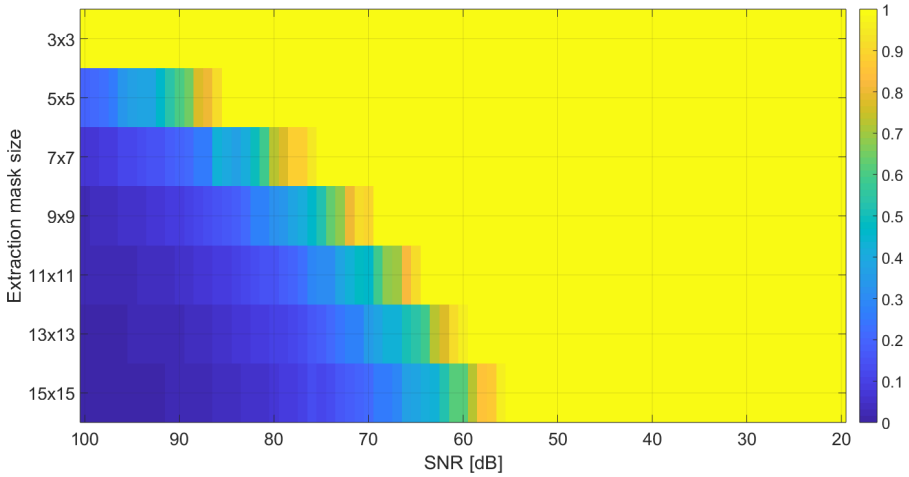


(b) Maximum angular error.

Figure G.28: 1° border for mean and maximum angular errors of single LoG algorithm for angle information extraction combined with AMM tracking feature. Encoder magnet: sphere. 5 mm z-distance.



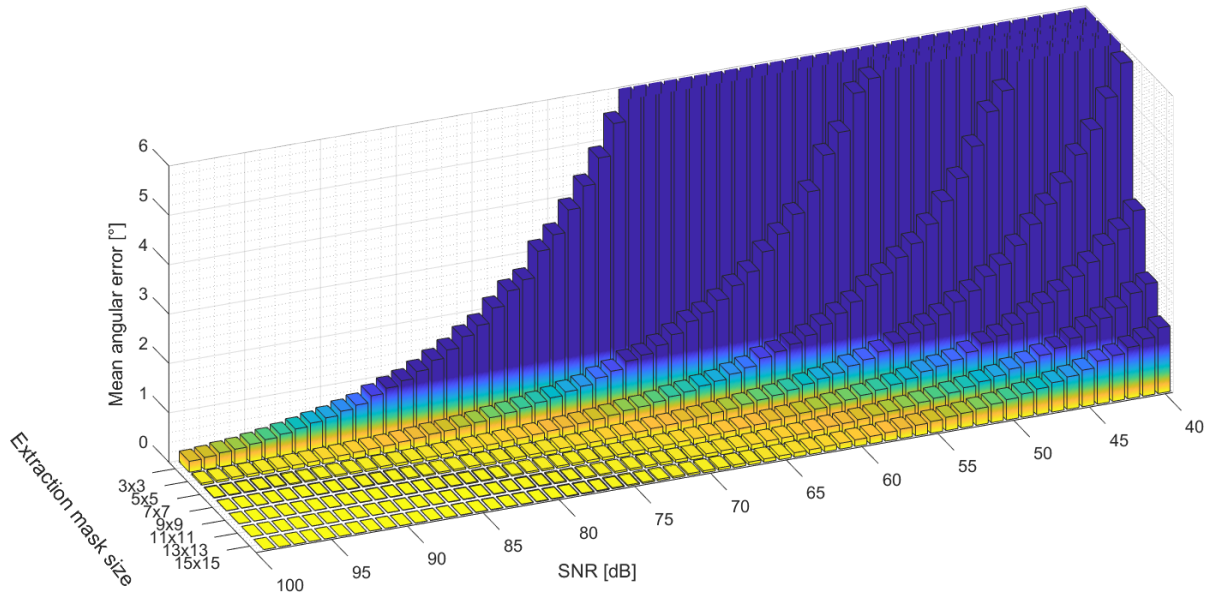
(a) Mean angular error.



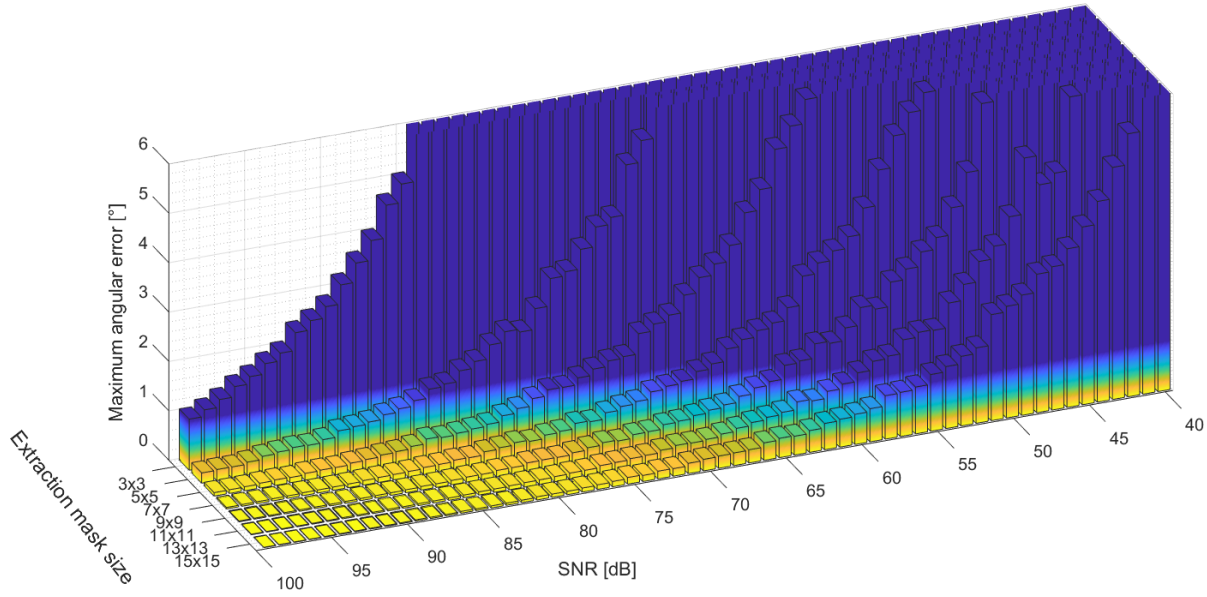
(b) Maximum angular error.

Figure G.29: 1° border for mean and maximum angular errors of approximation based algorithm for angle information extraction combined with AMM tracking feature. Encoder magnet: sphere. 5 mm z-distance.

Bar plots for complete angular error visualization, comparison for subblock 4.1. Magnet shape: sphere

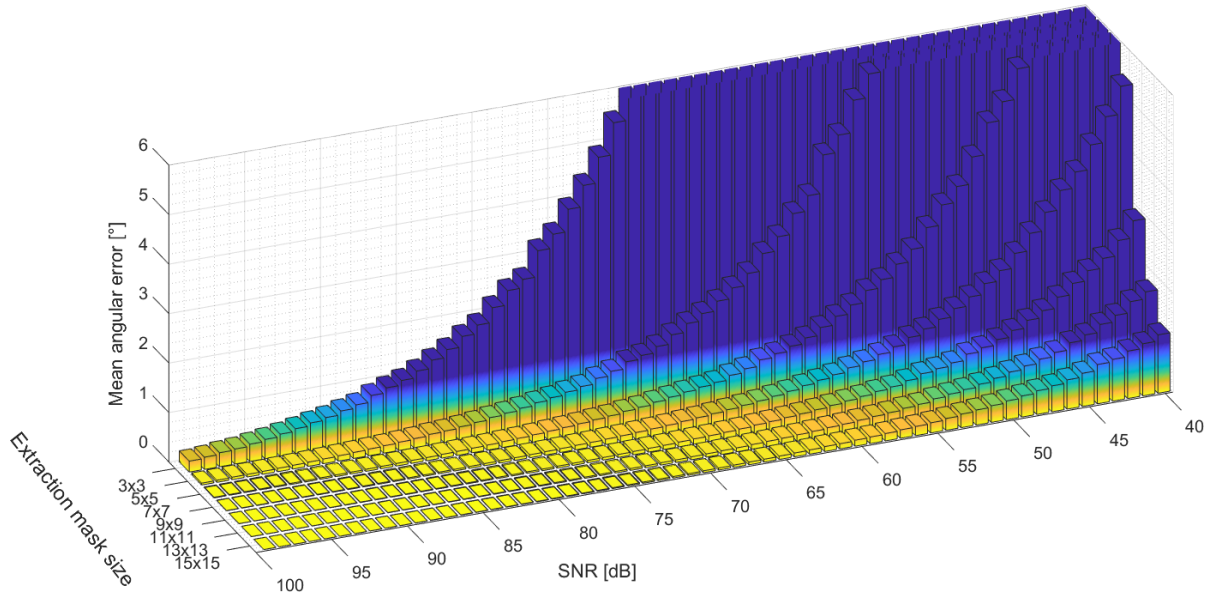


(a) Mean angular error.

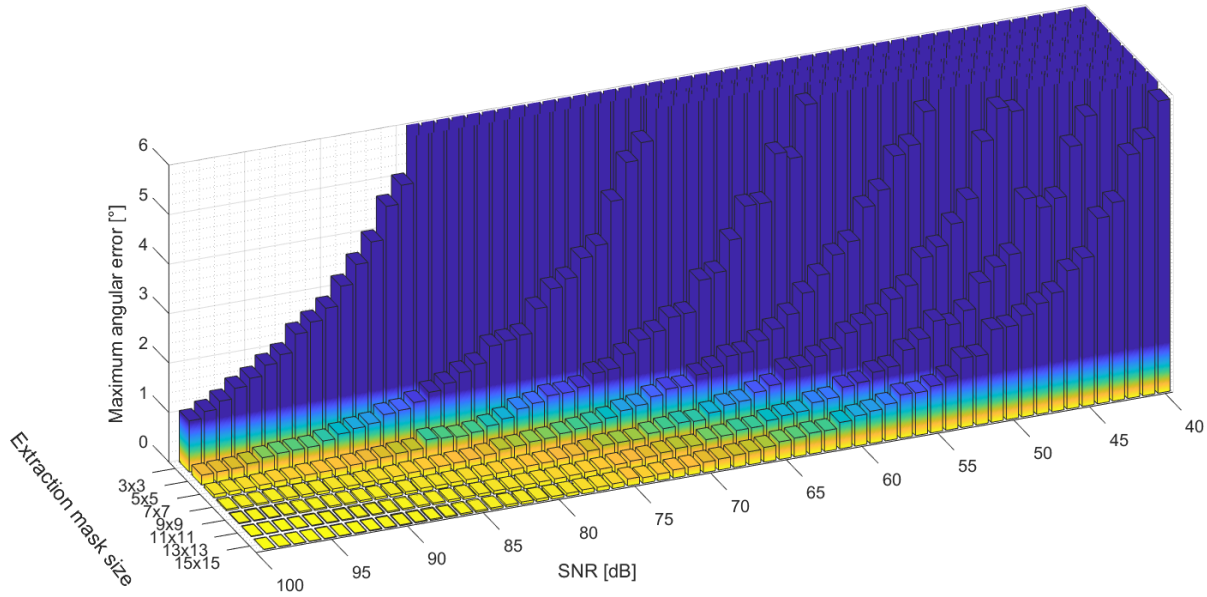


(b) Maximum angular error.

Figure G.30: Bar plots for mean and maximum angular errors of extended 2D-DFT algorithm for angle information extraction combined with AMM tracking feature. Encoder magnet: sphere. 5 mm z-distance.

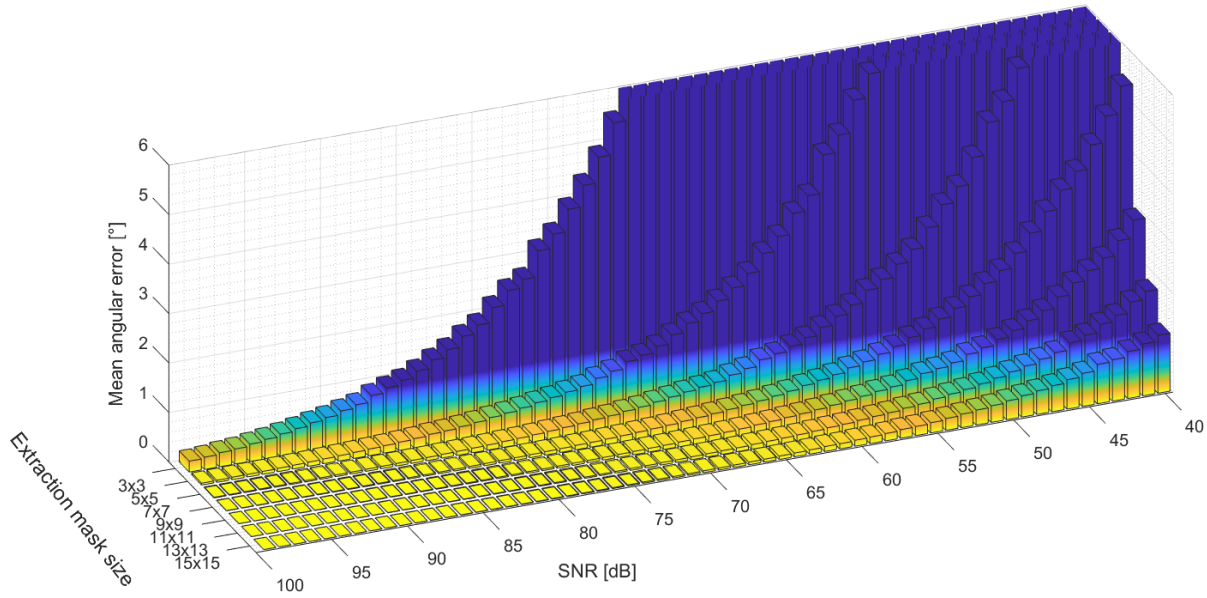


(a) Mean angular error.

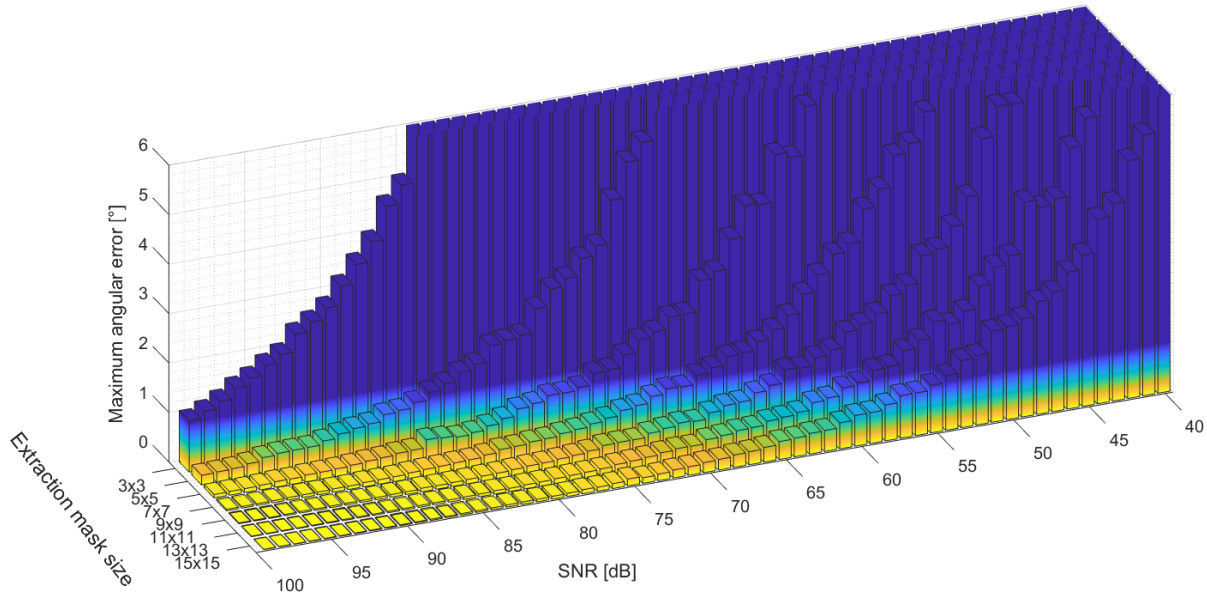


(b) Maximum angular error.

Figure G.31: Bar plots for mean and maximum angular errors of single LoG algorithm for angle information extraction combined with AMM tracking feature. Encoder magnet: sphere. 5 mm z-distance.



(a) Mean angular error.



(b) Maximum angular error.

Figure G.32: Bar plots for mean and maximum angular errors of approximation based algorithm for angle information extraction combined with AMM tracking feature. Encoder magnet: sphere. 5 mm z-distance.

G.8 Angular error correction (block 5)

G.8.1 Harmonics cancelation

Results of linear angular error correction algorithm based on homography

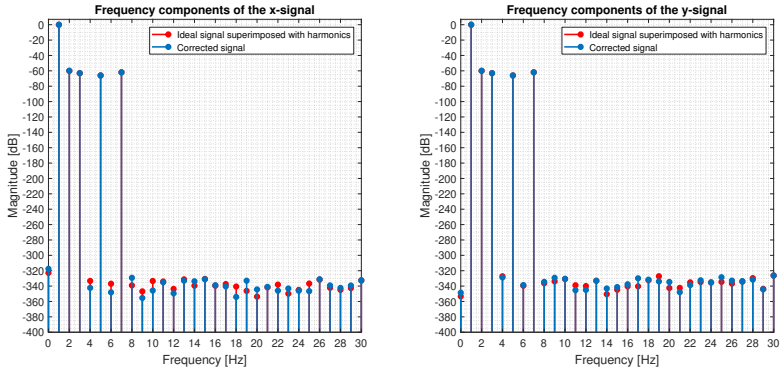


Figure G.33: DOF = 6 (6 coefficients), amount of matching points = 360

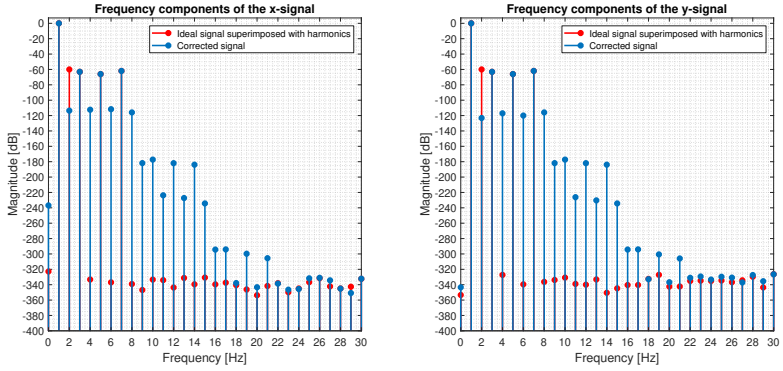


Figure G.34: DOF = 8 (8 coefficients), amount of matching points = 360

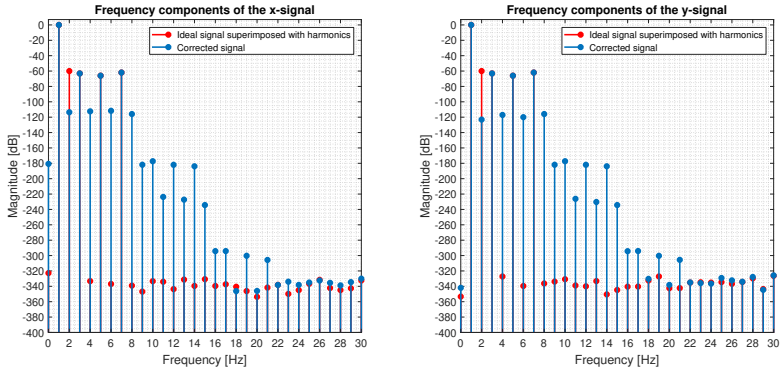


Figure G.35: DOF = 8 (9 coefficients), amount of matching points = 360

Results of nonlinear angular error correction algorithm based on geometric distortion cancelation

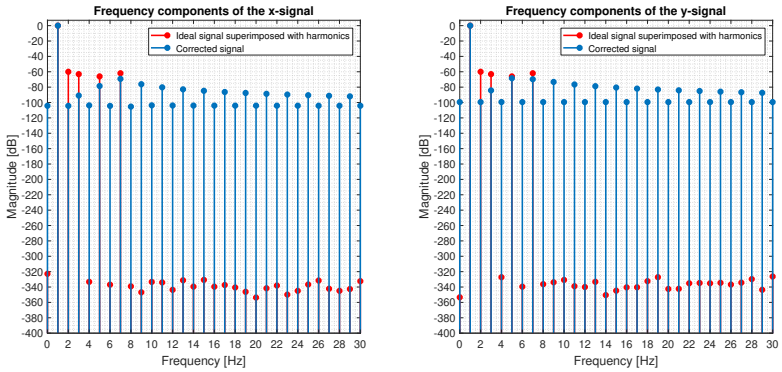


Figure G.36: Degree = 2 (6 coefficients), amount of matching points = 181*4

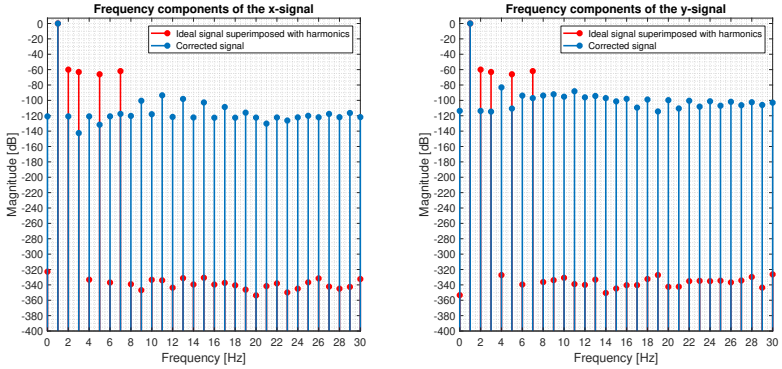


Figure G.37: Degree = 4 (8 coefficients), amount of matching points = 181*4

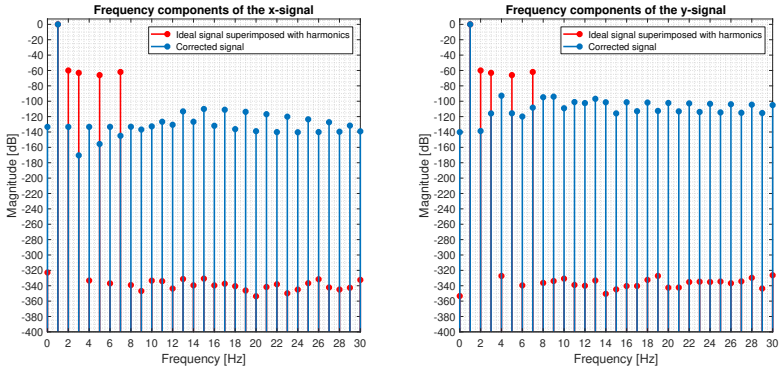


Figure G.38: Degree = 6 (10 coefficients), amount of matching points = 181*4

G.8.2 Misalignment compensation: extreme case

Maximum angular errors

Results of linear angular error correction algorithm based on homography

Sphere magnet

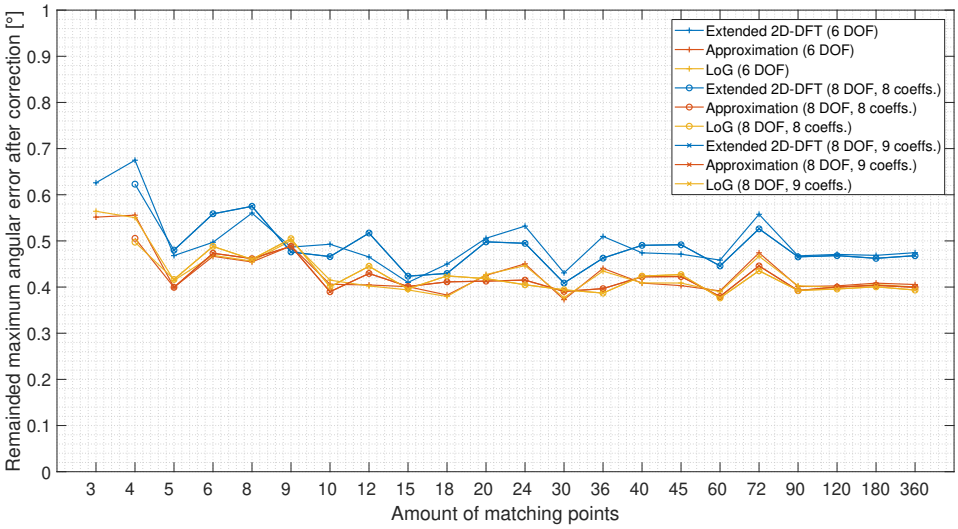


Figure G.39: Maximum angular error. *Simulation setup: 15 x 15 array, 1.596 mm x- and 1.463 mm y-translation, 7° x-tilt, 5 mm z-distance, 14 bit ADC, sphere magnet*

Disc magnet

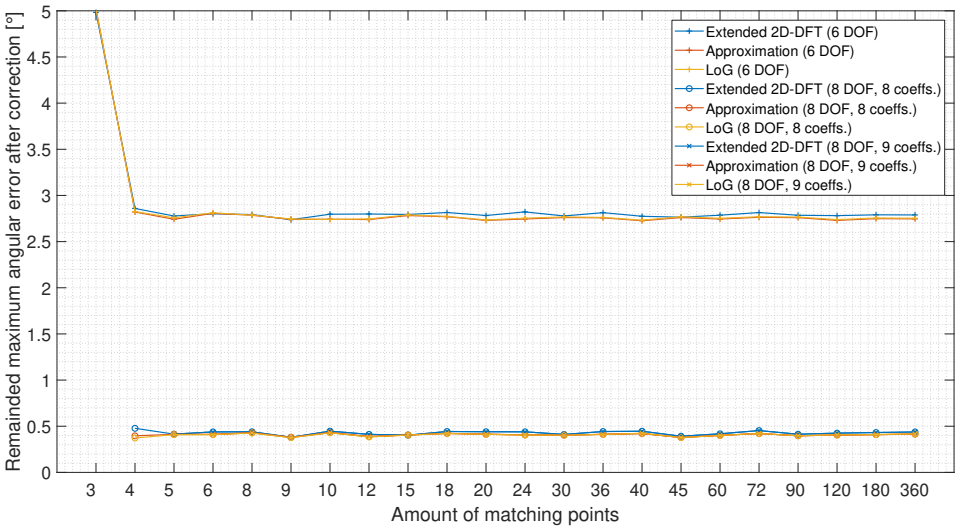


Figure G.40: Maximum angular error. *Simulation setup: 15 x 15 array, 1.596 mm x- and 1.463 mm y-translation, 7° x-tilt, 5 mm z-distance, 14 bit ADC, sphere magnet.*

Results of nonlinear angular error correction algorithm based on geometric distortion compensation

Sphere magnet

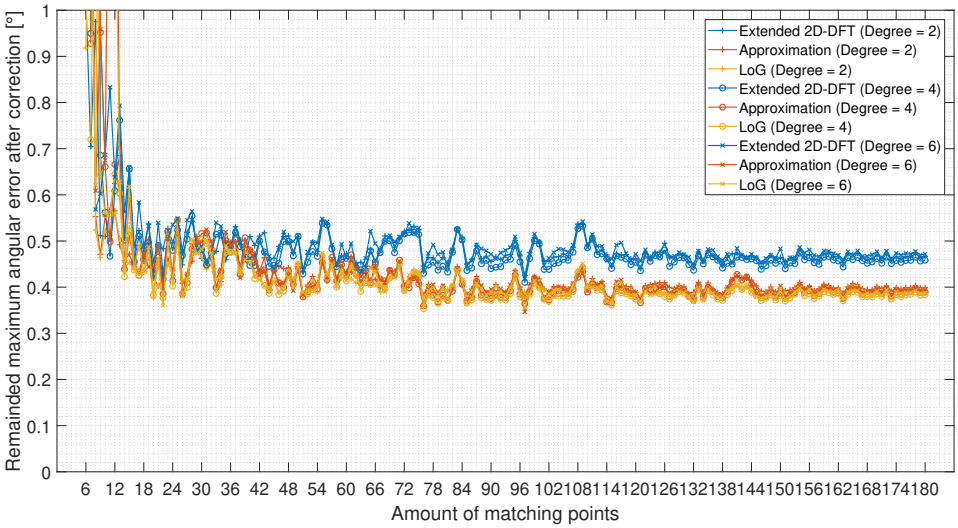


Figure G.41: Maximum angular error. *Simulation setup: 15 x 15 array, 1.596 mm x- and 1.463 mm y-translation, 7° x-tilt, 5 mm z-distance, 14 bit ADC, sphere magnet.*

Disc magnet

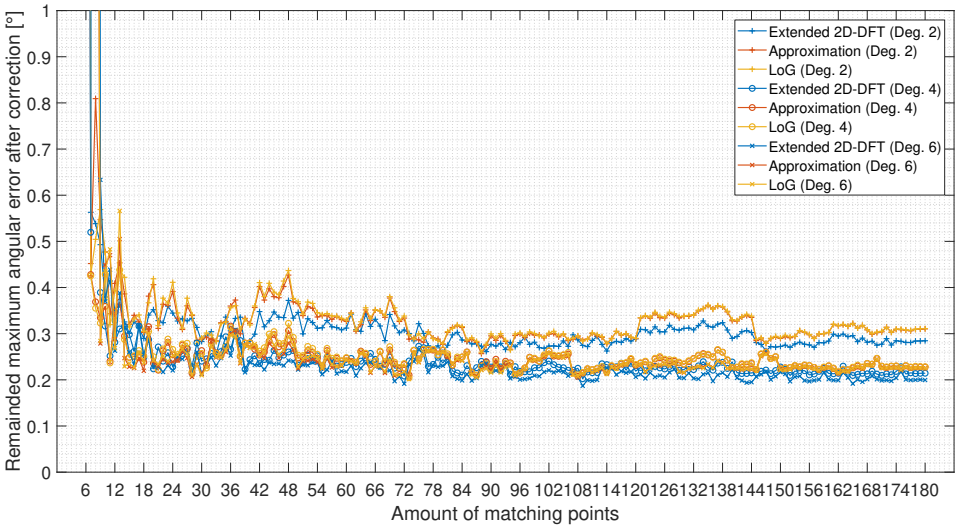
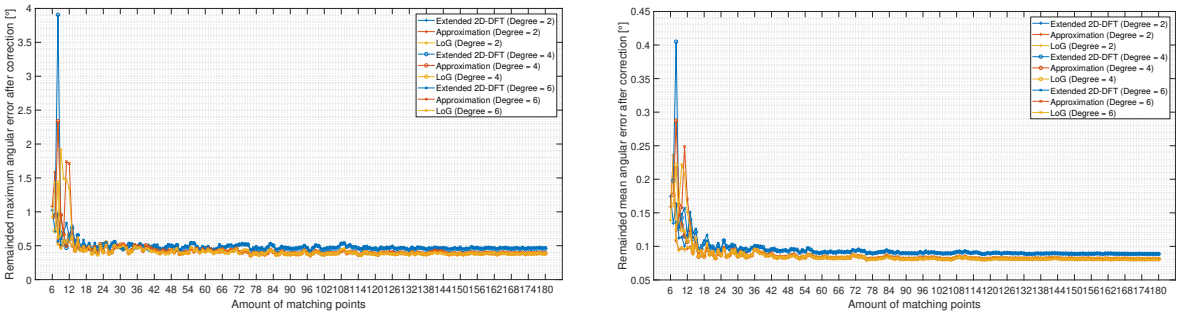


Figure G.42: Maximum angular error. *Simulation setup: 15 x 15 array, 1.596 mm x- and 1.463 mm y-translation, 7° x-tilt, 5 mm z-distance, 14 bit ADC, sphere magnet.*

Results of nonlinear angular error correction algorithm based on geometric distortion cancelation (full scale versions)

Sphere magnet

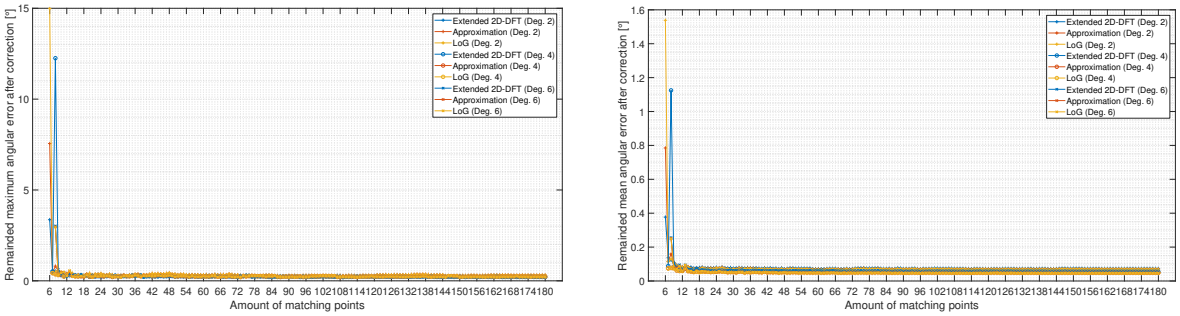


(a) Maximum angular error

(b) mean angular error

Figure G.43: Investigation of nonlinear angular error correction algorithm based on geometric distortion cancelation: misalignment compensation. *Simulation setup: 15 x 15 array, 1.596 mm x- and 1.463 mm y-translation, 7° x-tilt, 5 mm z-distance, 14 bit ADC, sphere magnet.*

Disc magnet



(a) Maximum angular error

(b) Mean angular error

Figure G.44: Investigation of nonlinear angular error correction algorithm based on geometric distortion cancelation: misalignment compensation. *Simulation setup: 15 x 15 array, 1.596 mm x- and 1.463 mm y-translation, 7° x-tilt, 5 mm z-distance, 14 bit ADC, disc magnet.*

Harmonics

The results are shown only for extended 2D-DFT algorithms because its larger error in comparison to LoG and approximation algorithms.

Results of linear angular error correction algorithm based on homography

Sphere magnet

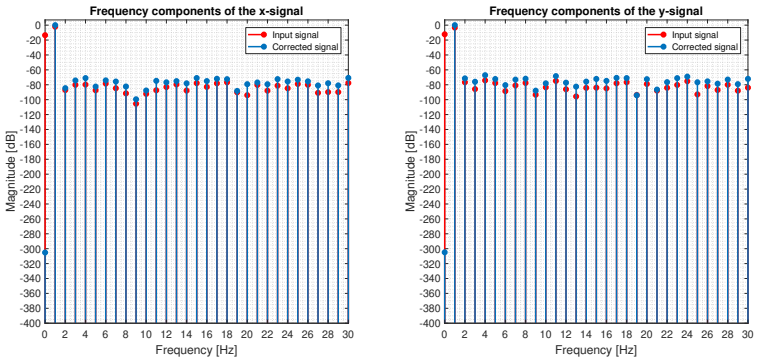


Figure G.45: DOF = 6 (6 coefficients), amount of matching points = 360

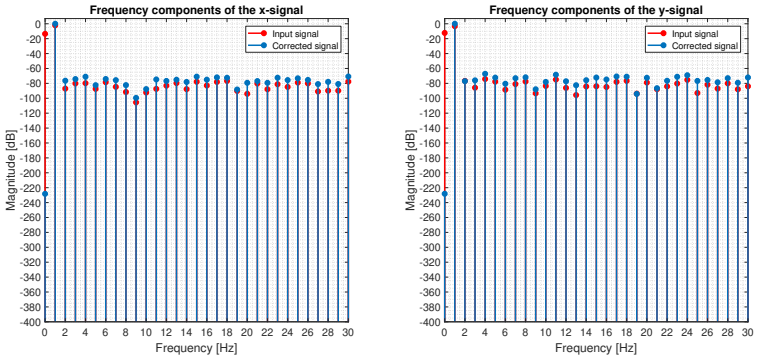


Figure G.46: DOF = 8 (8 coefficients), amount of matching points = 360

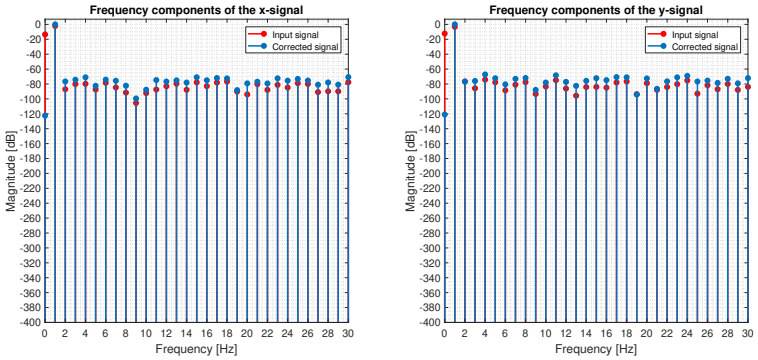


Figure G.47: DOF = 8 (9 coefficients), amount of matching points = 360

Disc magnet

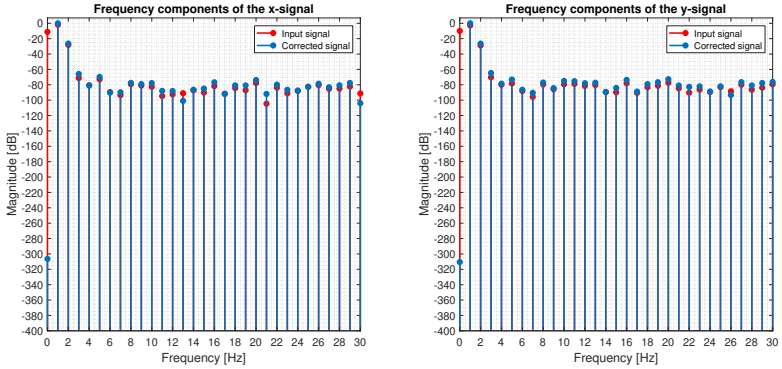


Figure G.48: DOF = 6 (6 coefficients), amount of matching points = 360

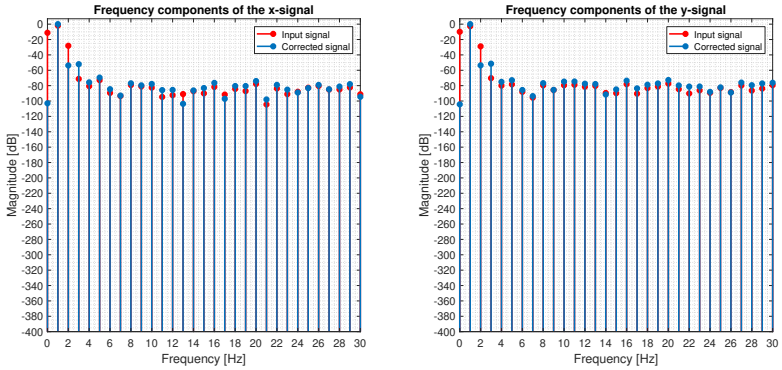


Figure G.49: DOF = 8 (8 coefficients), amount of matching points = 360

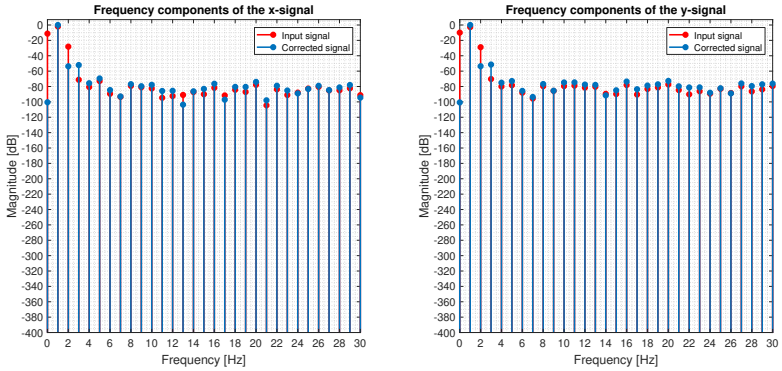


Figure G.50: DOF = 8 (9 coefficients), amount of matching points = 360

Results of nonlinear angular error correction algorithm based on geometric distortion cancelation
Sphere magnet

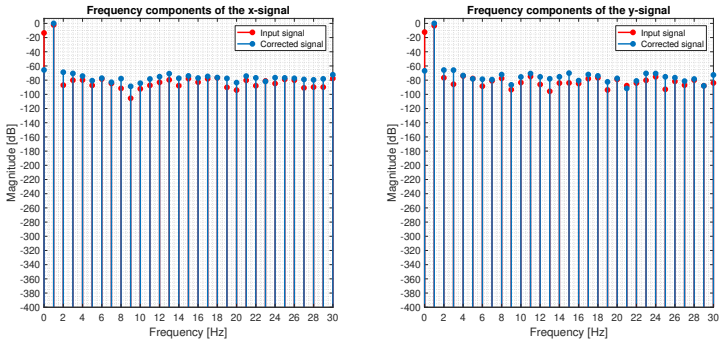


Figure G.51: Degree = 2 (6 coefficients), amount of matching points = 181*4.

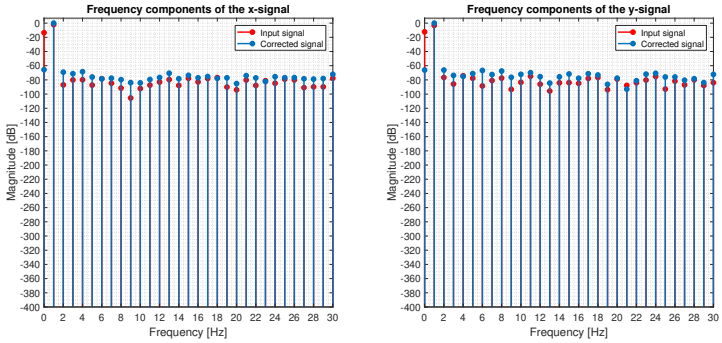


Figure G.52: Degree = 4 (8 coefficients), amount of matching points = 181*4.

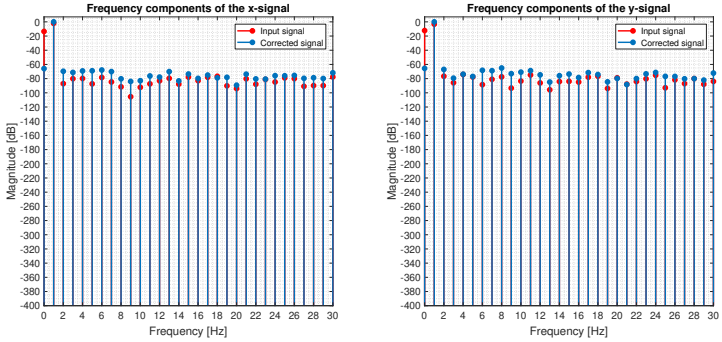


Figure G.53: Degree = 6 (10 coefficients), amount of matching points = 181*4.

Disc magnet

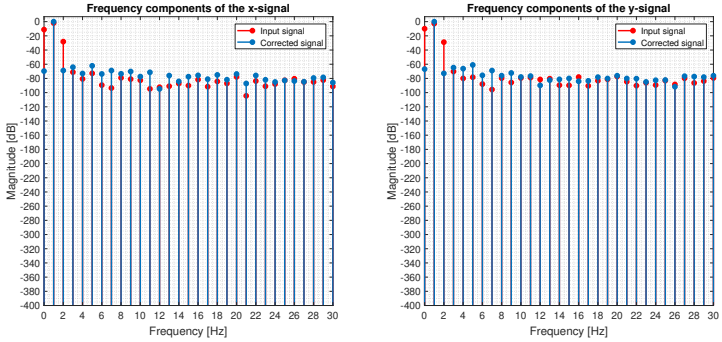


Figure G.54: Degree = 2 (6 coefficients), amount of matching points = 181*4.

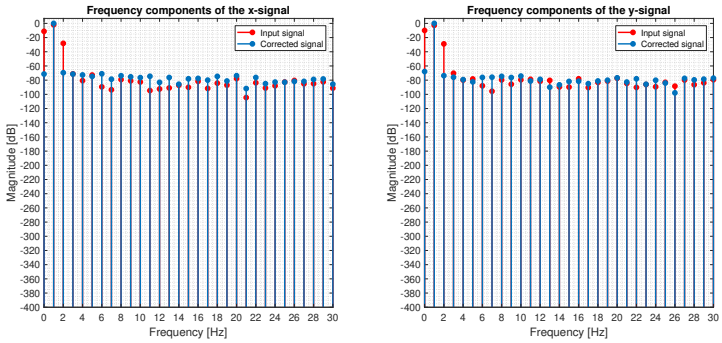


Figure G.55: Degree = 4 (8 coefficients), amount of matching points = 181*4.

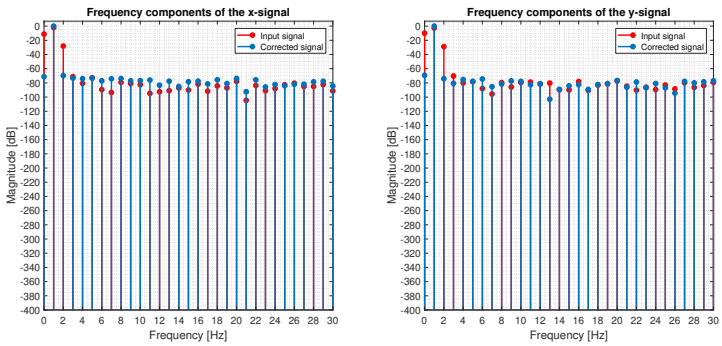


Figure G.56: Degree = 6 (10 coefficients), amount of matching points = 181*4.

G.8.3 Misalignment compensation: worst SOA case

The discussion of the algorithm results in the following figures are related to the algorithms *2D-DFT, Approx, LoG, and Conv.* Results of other candidates in the shown GUI graphs were not covered in the thesis and therefore not discussed.
Worst SOA case setup: z-distance 2 mm, 1mm x- and y-translation, x-tilt 5°. Applied ADC resolution: 12 bit.
Sphere magnet

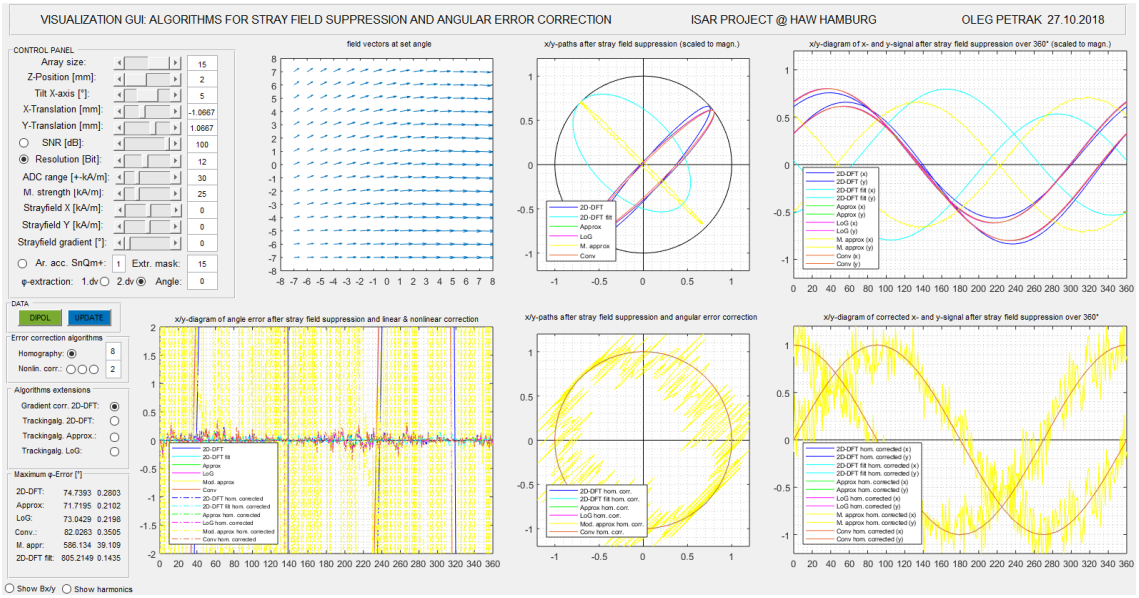


Figure G.57: Linear angular error correction based on homography approach. A perfect correction is obtained already by homography DOF of 6, see the graphs in the lower part of the GUI. The calculated x- and y- signals exhibit a perfect progression. Only the quantization noise is visible in the angular error graph. The angular error from investigated algorithms was corrected from approximately 70 ° to 0.2° - 0.3°. *Correction setup: DOF = 6, amount of matching points = 360.*

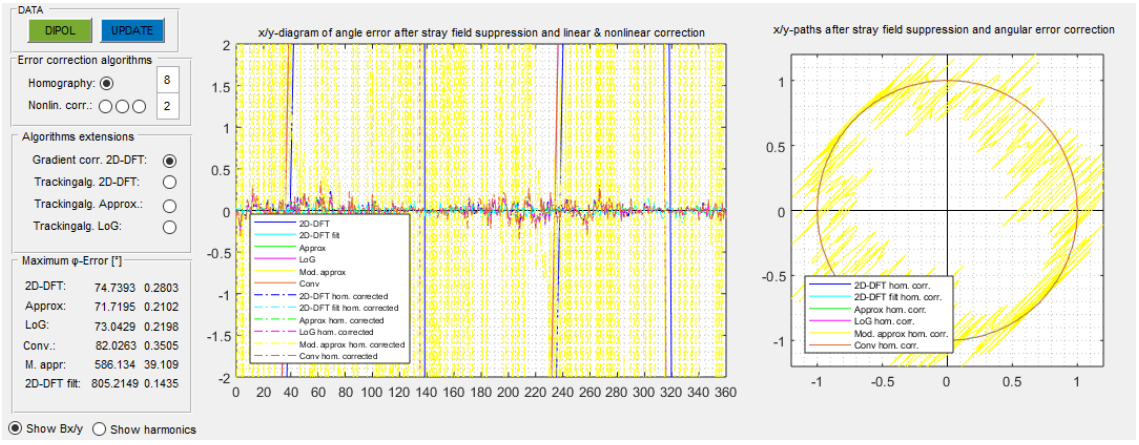


Figure G.58: Linear angular error correction based on homography approach. Angular error window and algorithms control panel Figure G.57.

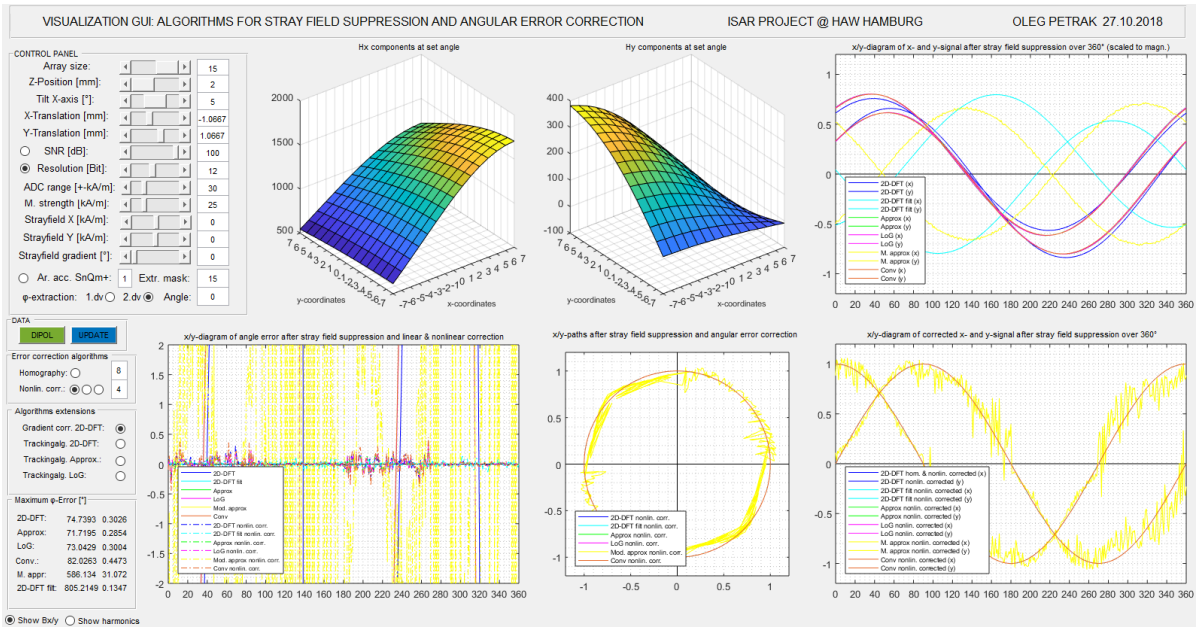


Figure G.59: Nonlinear angular error correction based on geometric distortion. Also here a perfect correction is obtained, already by a degree of 2, see the graphs in the lower part of the GUI. The calculated x- and y- signals exhibit a perfect progression. Besides the quantization noise also a small impact of of transitions of x-/y-signals are visible. However, its impact is in an absolutely tolerable range. The angular error from investigated algorithms was corrected from approximately 70° to 0.4° - 0.3°. *Correction setup: Degree = 2 (6 coefficients), amount of matching points = 181*4.*

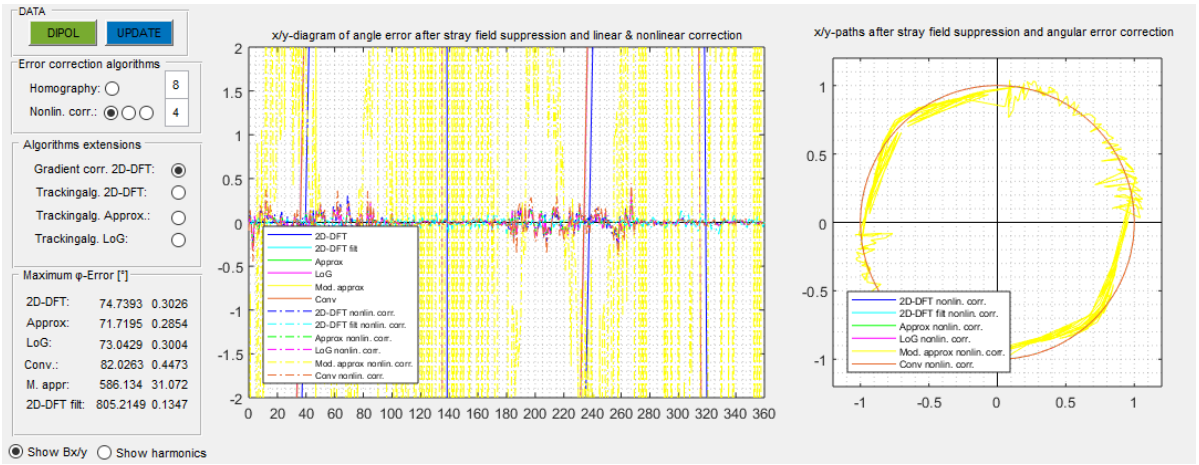


Figure G.60: Nonlinear angular error correction based on geometric distortion. Angular error window and algorithms control panel from Figure G.59.

Disc magnet

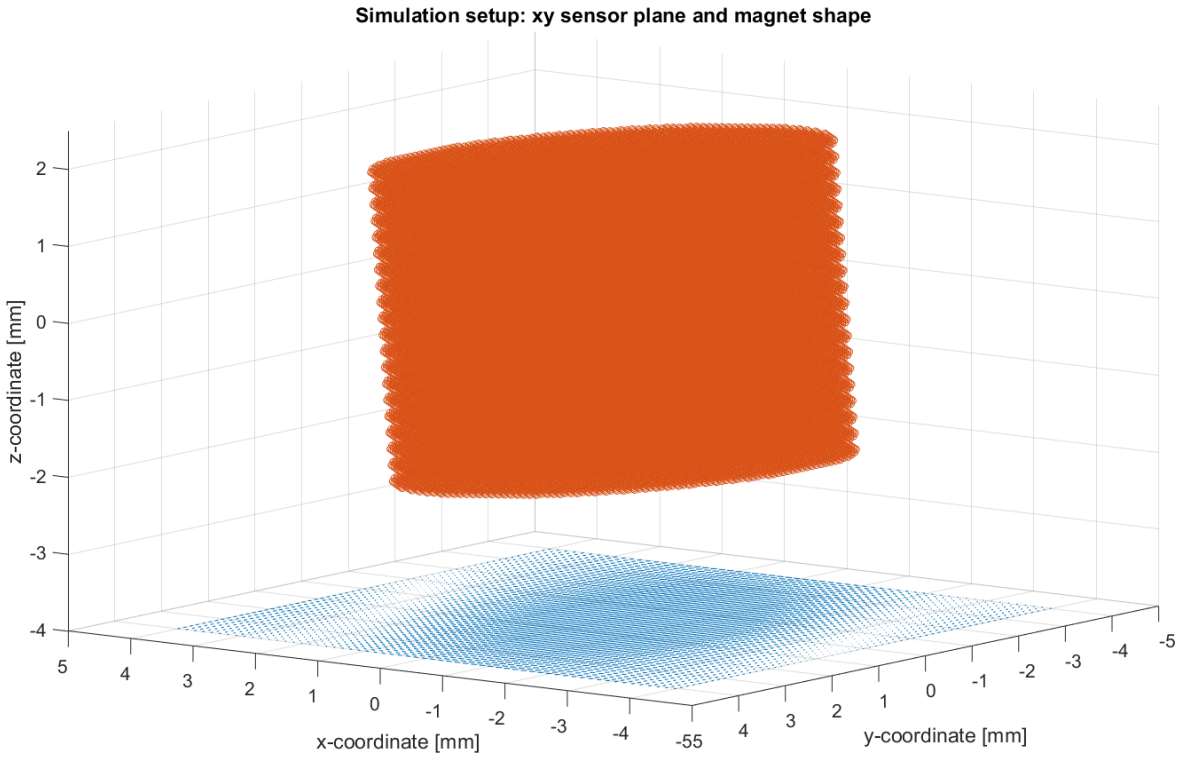


Figure G.61: Visualization of mechanical misalignment of disc encoder magnet. Worst SOA misalignment case: 5° x-tilt and 2 mm z-distance.

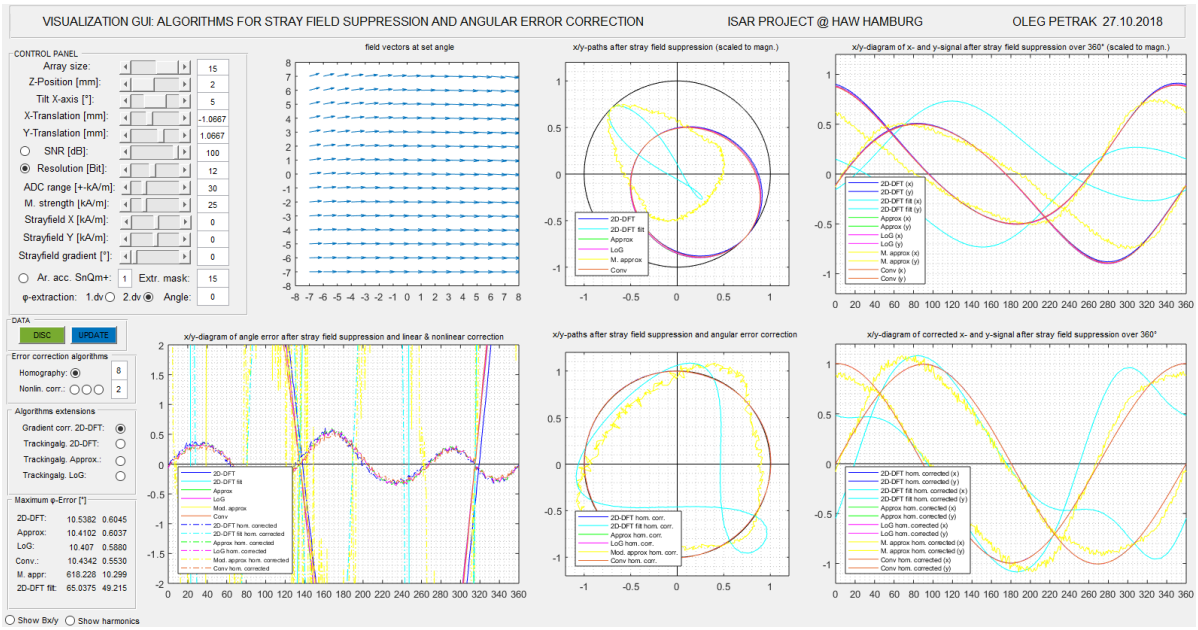


Figure G.62: Linear angular error correction based on homography approach. The progressions of x- and y-signals are not sinusoidal. This is an evidence for the occurrence of harmonics. For the worst misalignment case within the SOA applied on the disc magnet it is not possible to correct the angular error perfectly by homography DOF 8. However, the remained angular error is within the maximum error range of $\varphi_{err} \leq 1°$. The angular error from investigated algorithms was corrected from approximately 10° to 0.6°. *Correction setup: DOF = 8, amount of matching points = 360.*

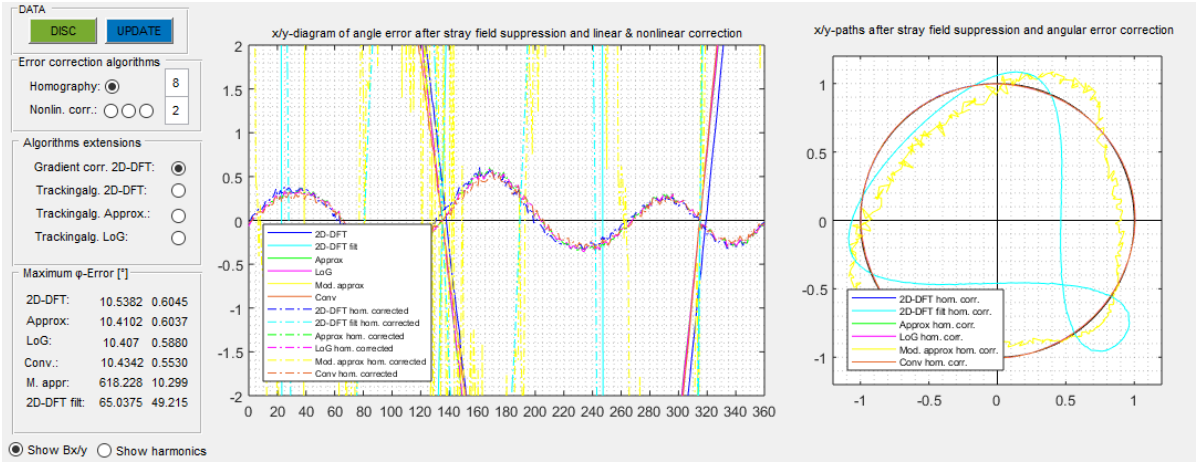


Figure G.63: Angular error window and algorithms control panel from Figure G.62.

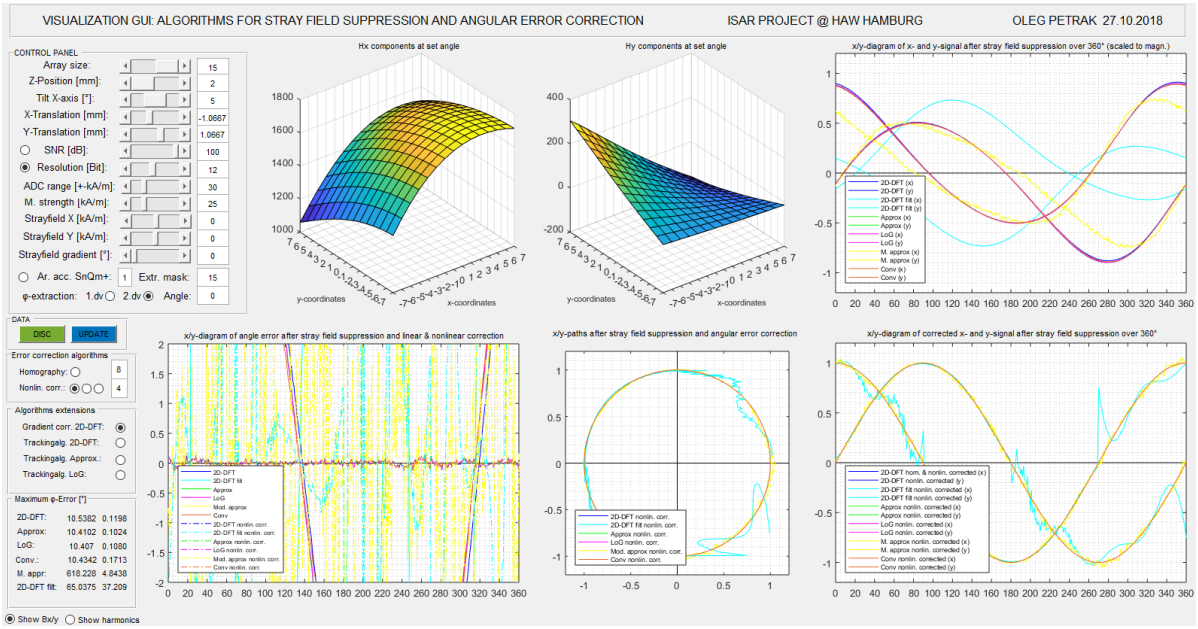


Figure G.64: Nonlinear angular error correction based on geometric distortion. The progressions of x- and y-signals are not sinusoidal. This is an evidence for the occurrence of harmonics. For the worst misalignment case within the SOA applied on the disc magnet it becomes possible to correct the angular error almost perfectly. A tiny error is still observable but is in the close range with quantization noise error. The angular error from investigated algorithms was corrected from approximately 10° to 0.1°. *Correction setup: Degree = 4 (8 coefficients), amount of matching points = 181*4.*

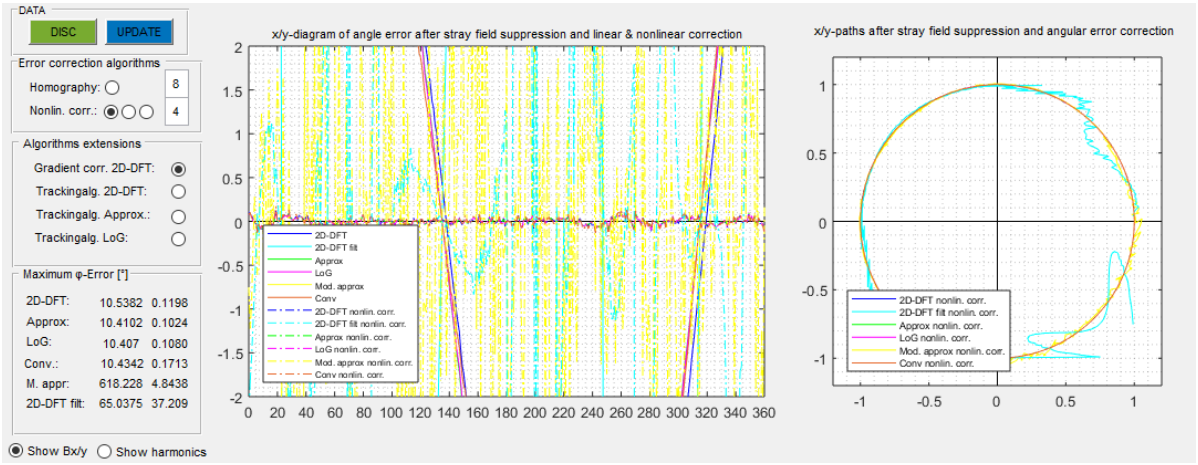
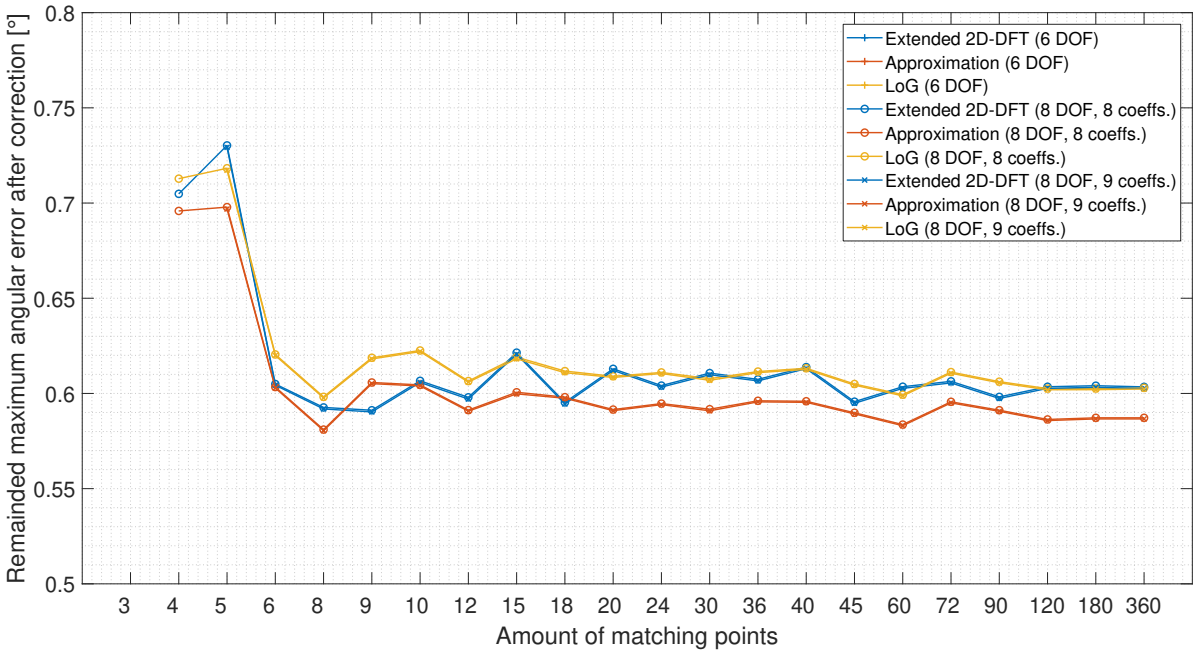


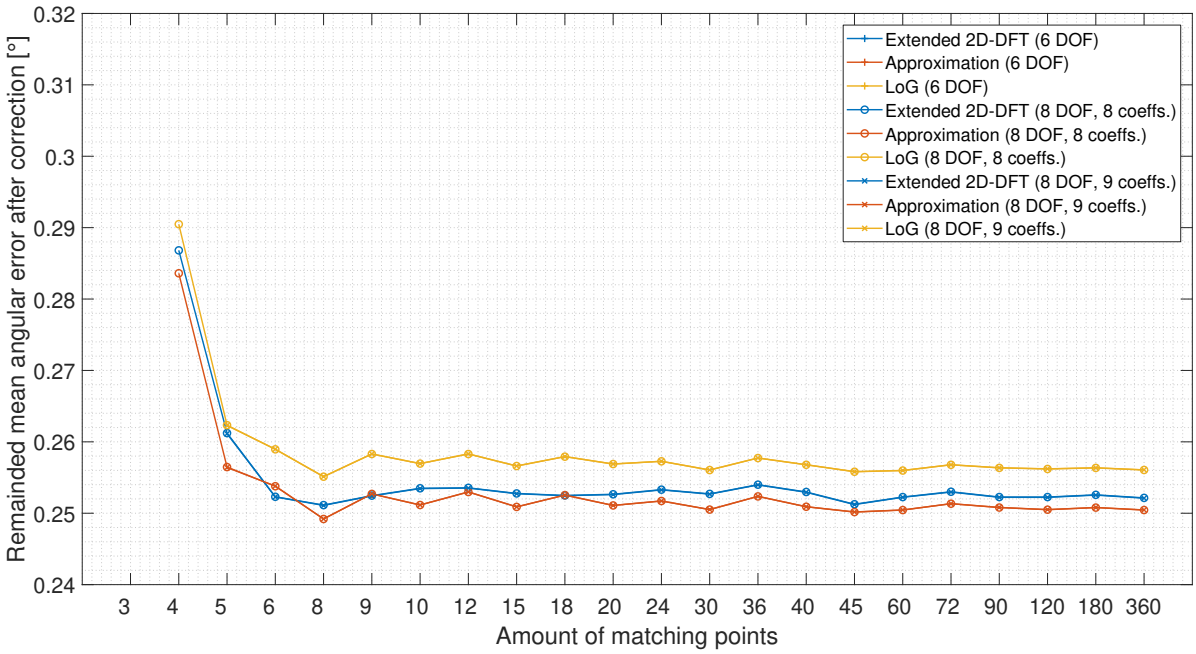
Figure G.65: Angular error window and algorithms control panel from Figure G.64.

Results of linear angular error correction algorithm based on homography approach: amount of matching points

Disc magnet



(a) Maximum angular error



(b) Mean angular error

Figure G.66: Investigation of linear angular error correction algorithm based homography approach: misalignment compensation (worst SOA case). 6 DOF not shown. *Simulation setup: 15 x 15 array, 1 mm x- and 1 mm y-translation, 5° x-tilt, 2 mm z-distance, 12 bit ADC, disc magnet.*

Disc magnet

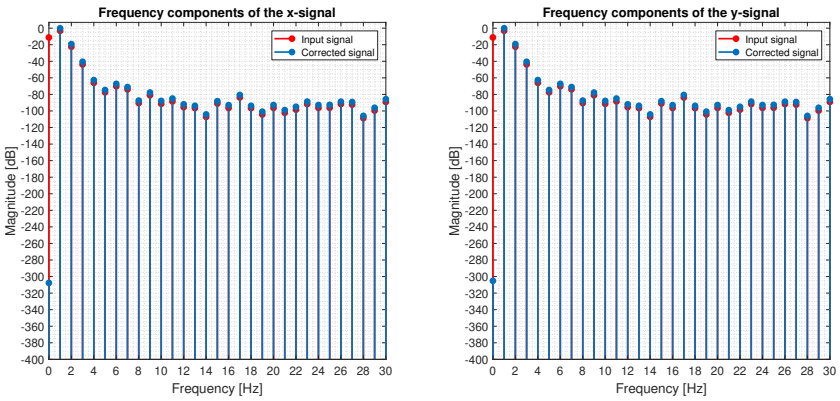


Figure G.67: DOF = 6 (6 coefficients), amount of matching points = 360

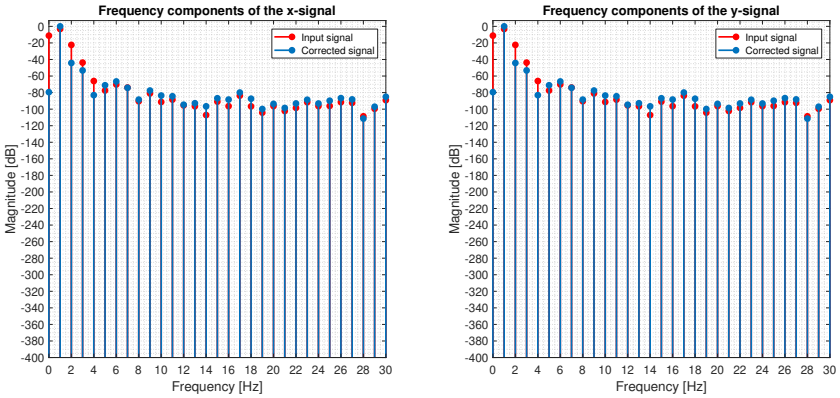


Figure G.68: DOF = 8 (8 coefficients), amount of matching points = 360

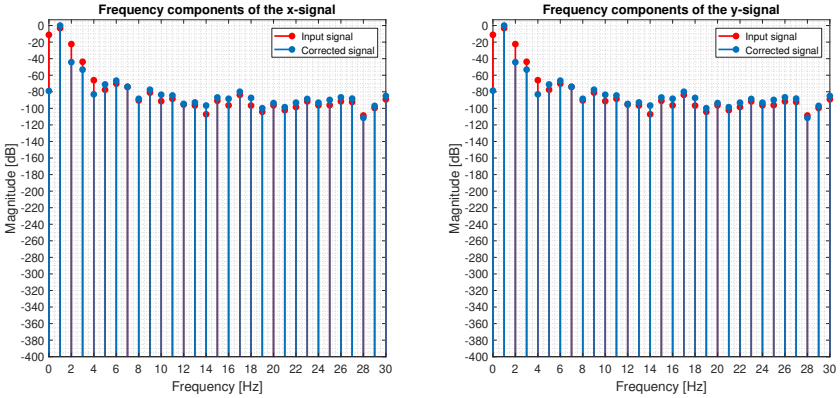
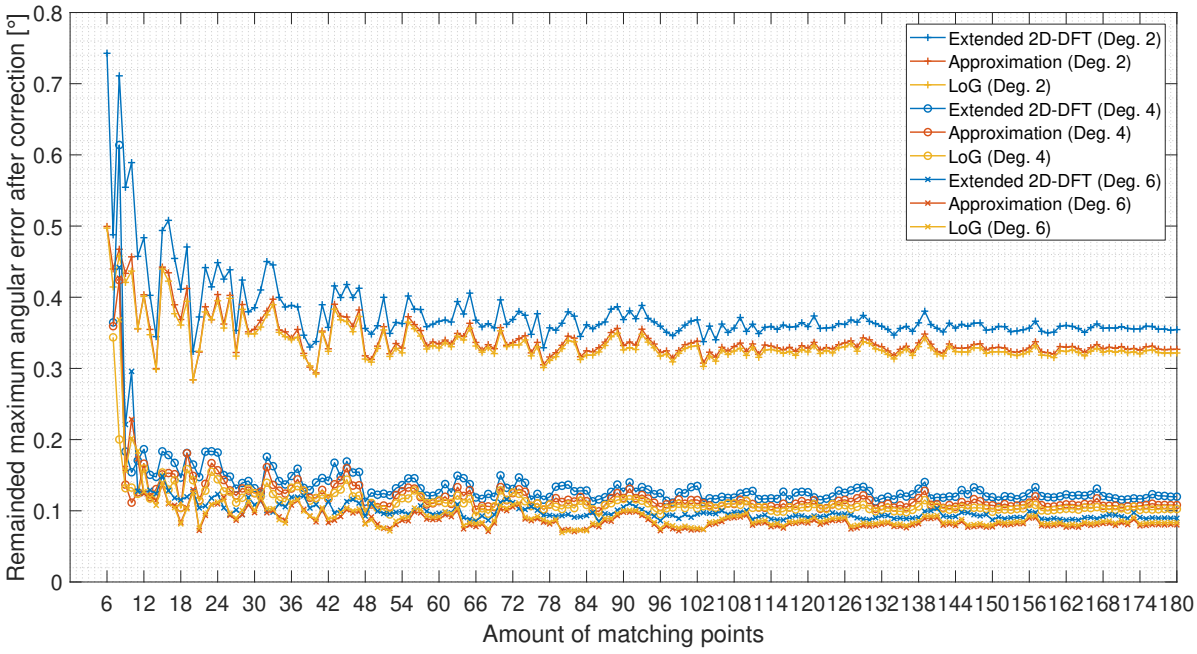


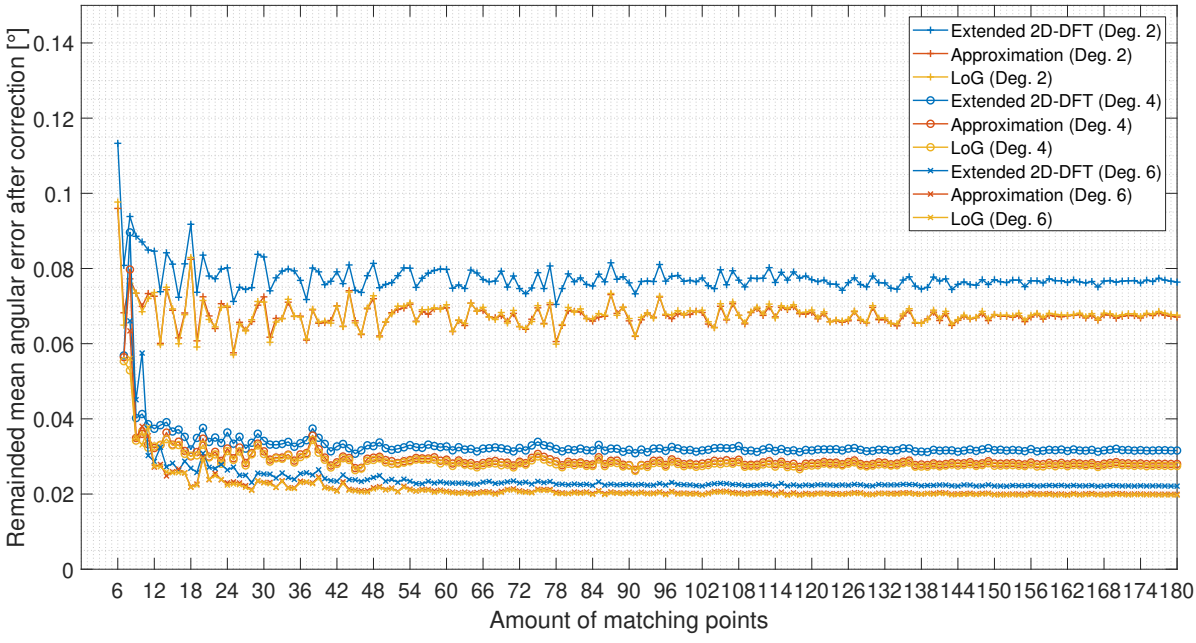
Figure G.69: DOF = 8 (9 coefficients), amount of matching points = 360

Results of nonlinear angular error correction algorithm based on geometric distortion cancelation: amount of matching points

Disc magnet



(a) Maximum angular error



(b) Mean angular error

Figure G.70: Investigation of nonlinear angular error correction algorithm based on geometric distortion cancelation: misalignment compensation (worst SOA case). *Simulation setup: 15 x 15 array, 1 mm x- and 1 mm y-translation, 5° x-tilt, 2 mm z-distance, 12 bit ADC, disc magnet.*

Results of nonlinear angular error correction algorithm based on geometric distortion cancelation: harmonics cancelation caused by encoder magnet misalignment

Disc magnet

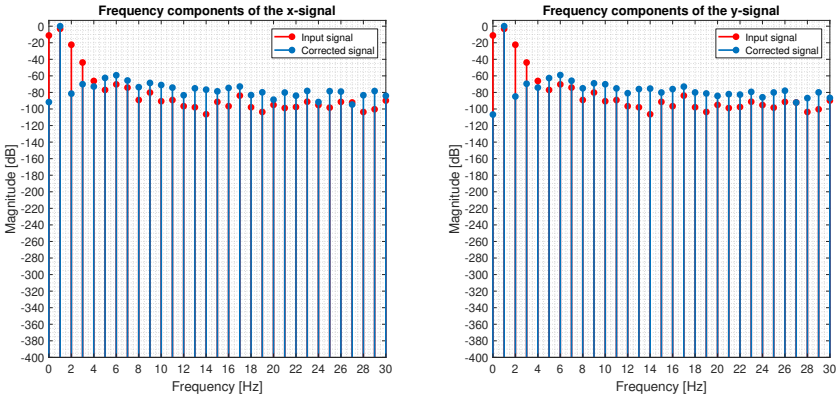


Figure G.71: Degree = 2 (6 coefficients), amount of matching points = 181*4.

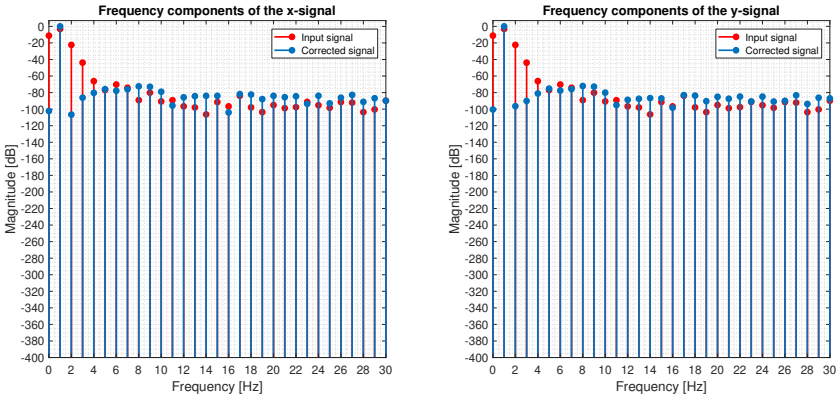


Figure G.72: Degree = 4 (8 coefficients), amount of matching points = 181*4.

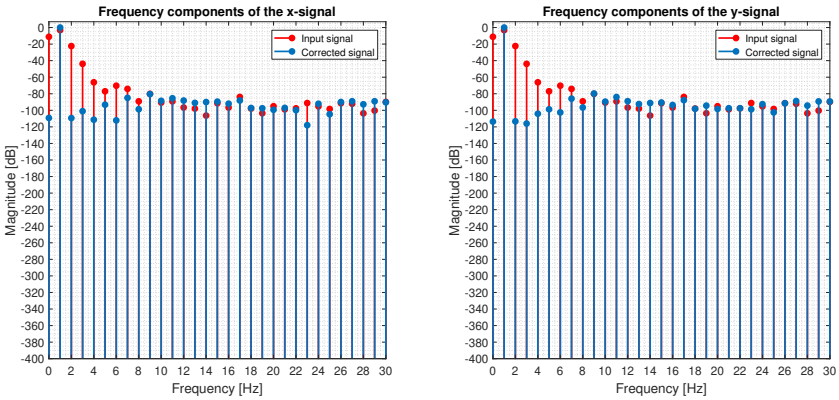


Figure G.73: Degree = 6 (10 coefficients), amount of matching points = 181*4.

Appendix H: Sensor array demonstration system

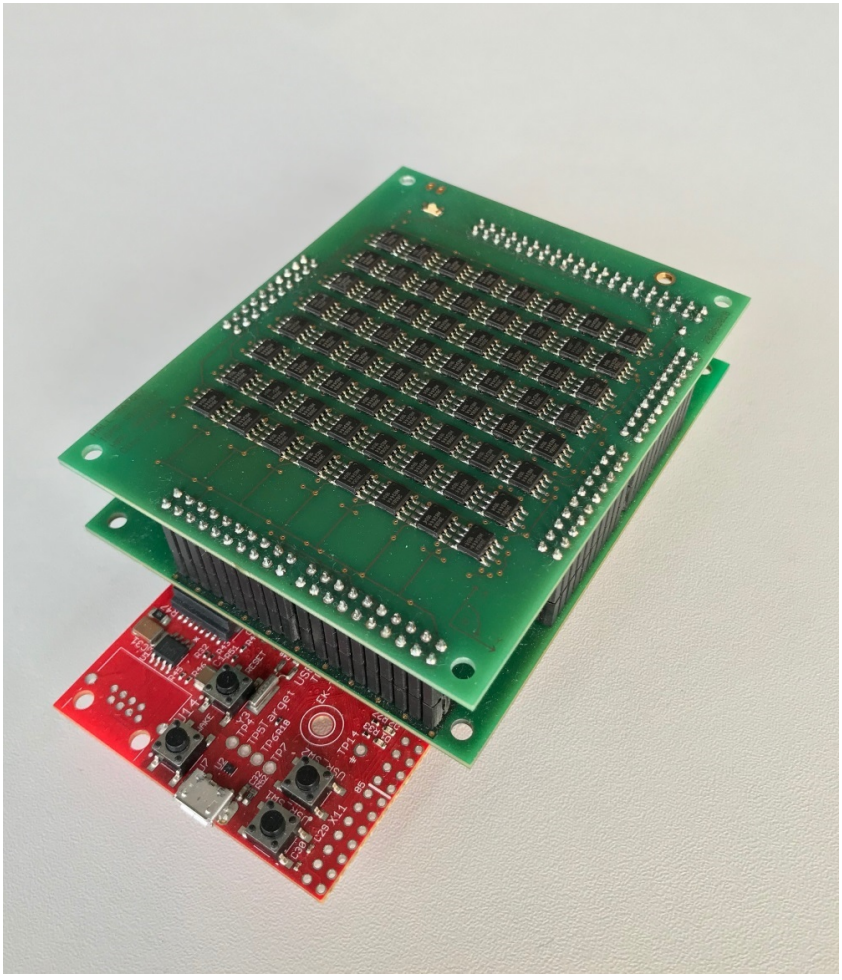


Figure H.1: Sensor array demonstration system. **Top module:** Designed sensor matrix consisting of 8 x 8 Hall sensors from Melexis of type MLX90380LDC-BAB-100-SP-ND [26] (5V Vdd, X/Y magnetic axis - equal sensitivity, 40 mT, low bandwidth). **Medium module:** Data acquisition and galvanic decoupling module. Used ADCs: 2 x 200 ksps 16 bit 8-channel SAR ADCs of type LTC1867CGN#PBF from Linear Technology. Communication with the μC module is realized via 2 x SPI interfaces. **Lower module:** Texas Instruments Launch Pad TM4C129XL, 120 MHz 32Bit ARM Cortex 4 CPU. The communication with the terminal programm for application of algorithms is realized via UART interface communication.

Versicherung über die Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §16(5) APSO-INGI ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht.

Hamburg, 26. Dezember 2018	
Ort, Datum	Unterschrift