Technology

Daniel Piri

Sensor Fusion for Nanopositioning

Master's Thesis

Bibliographic information published by the German National Library:

The German National Library lists this publication in the National Bibliography; detailed bibliographic data are available on the Internet at http://dnb.dnb.de .

Imprint:

Copyright © 2014 GRIN Verlag
ISBN: 9783656865247

This book at GRIN:

https://www.grin.com/document/286249

Daniel Piri

Sensor Fusion for Nanopositioning

GRIN Verlag

Sensor Fusion for Nanopositioning

DIPLOMARBEIT

Ausgeführt zum Zwecke der Erlangung des akademischen Grades eines
Diplom-Ingenieurs (Dipl.-Ing.)

eingereicht an der
Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik
Institut für Automatisierungs- und Regelungstechnik

von
Daniel Piri, BSc.

Wien, Österreich

Wien, im November 2014

Gruppe für Industrielle Automationstechnik

Acknowledgement

This master thesis was written at the Institute for Automation and Control at Vienna University of Technology as completion to a Master's degree in Automation. The master's program focuses on modelling and control of complex dynamic systems, as well as on the accomplishment of challenges in embedded mechatronic systems. The subject of this thesis, sensor fusion, treats many master's fields and far beyond that. I applied and linked my reasearch to the multilayered field of position measurement in nano-positioning applications (e.g. in the Atomic Force Microscopy) by simplifying sensor selection, proposing an arrangement and circuitry of sensors and using a method to fuse data from multiple sensors to gain high resolution at high bandwidth.

I am extremely grateful to Prof. Dr. Andreas Kugi (Head of the Institute) for his structured, reasoned lessons and lecture notes, who inspired me to pursue studies in the field of automation and control.
My deep appreciation goes for his superior conduction to Prof. Dr. Georg Schitter, who steered me in the right direction. Without the solid fundamentals in understanding dynamic systems it would not have been possible to write the thesis at this high level. My sincere thanks goes to Shingo Ito, M.A.Sc. for interesting discussions and critical questions. Furthermore, I would like to express my thanks to DI Jürgen Steininger for supervision and for proofreading my work.

Last but not least, lots of thanks is addressed to my parents for their support and to my sweetheart Andrea for her huge patience.

Abstract

Sensors can be used to measure the position of an object. In the present thesis the effects which limit the usage of sensors in high dynamic positioning applications on a nanometer level are discussed. Various sensor principles and their properties are investigated and compared. Sensors based on the measurement of i.a. magnetic fields, illumination, or even strain are characterized, as well as their range, bandwidth, resolution, linearity and disturbance rejection is determined.

It will be shown that the simultaneous use of multiple sensors and the specific combination of sensors' data (*fusion*) enables a higher performance primarily in terms of resolution and dynamics. Several techniques for the fusion are discussed under consideration of various aspects, however the ultimate aim of sensor fusion is similar. The methods of feedforward control, complementary filtering, Kalman filtering and optimal filtering (robust control) are developed and verified on practical problems in position sensor systems. To treat various challenges in sensor filtering and sensor fusion a methodological approach, containing separable steps of

- problem formulation with well-defined prerequisits and simplifications,
- theory discussion with approach to find a solution,
- analytical proof or reasoning by statistical values out of numerical simulations,
- experiment design, and
- verification on a real time platform

are realized.

keywords: sensor fusion, nanopositioning, high dynamics, high resolution, sensor noise, Kalman filter, H_2-optimal filter, H_∞-optimal filter

Zusammenfassung

Sensoren werden vielfach verwendet um einen Abstand zu messen. In der vorliegenden Arbeit werden die Beschränkungen hinsichtlich des Einsatzes von Sensoren in hochdynamischen Präzisionsanwendungen auf Nanometer Ebene dargelegt. Verschiedene Messprinizipien, u.a. basierend auf der Erfassung von Magnetfeldern, Licht und mechanischen Spannungen werden auf Messbereich, Bandbreite, Auflösung, Nichtlinearitäten und Störunterdrückung charakterisiert.

Es wird gezeigt, dass der gleichzeitige Einsatz von mehreren Sensoren, und die spezifische Kombination von dessen Signalen (Fusion) eine Erhöhung der Positionsgenaugkeit und der Bandbreite bei dem Einsatz in Regelschleifen erlaubt. Es werden Methoden zur Filterung und Fusion vorgestellt und auf praktischen Problemstellungen angewandt. Unter anderem sind dies Komplementärfilter, Kalman Filter bzw. optimale Filter, gegliedert jeweils in

- Problemstellung inklusive Voraussetzungen,
- die theoretische Vorgehensweise mit einem Lösungsansatz,
- analytische Beweise u.A. zur Stabilität oder Folgerungen mit statistischen Mittel
- Messungen auf einer Echtzeitplatform und Rückschlüsse.

Schlagwörter: Nano-positionierung, hohe Dynamik, hohe Auflösung, Sensorrauschen, Kalman Filter, Optimale Filter

Résumé

Les capteurs peuvent être utilisés pour mesurer la position d'un objet. Dans cette thèse les effets qui limitent l'utilisation des capteurs dans les applications de positionnement dynamiques élevées au niveau nanométrique sont discutées. Différents principes de détection et leurs propriétés sont étudiés et comparés. Capteurs basés sur la mesure des champs magnétiques, l'éclairage, ou même de la dilatation sont caractérisés; ainsi que leur gamme de mesure, la bande passante, la résolution, la linéarité et l'antibrouillage est déterminé.

On va démontrer, que l'utilisation simultanée de plusieurs capteurs et la combinaison spécifique des données - la fusion des capteurs - offre une performance plus élevé, surtout en termes de résolution et de la dynamique. Plusieurs techniques de la fusion sont discutées, par rapport aux différents aspects. Les méthodes pour le réglage en boucle ouverte, filtrage complémentaire, filtrage de Kalman et filtrage optimal (commande robuste) sont développées et vérifiées sur des problèmes pratiques dans les systèmes de détection de position.

Mots-clés: fusion de capteurs, nano-positionnement, dynamique élevée, haute résolution, le bruit du capteur, filtre de Kalman, H_∞ filtre optimal

Contents

List of Figures

List of Tables

Acronyms

ADC	Analog to Digital Converter
AFM	Atomic Force Microscope
DAQ	Data Acquisition
dBc	deciBels to carrier
EKF	Extended Kalman Filter
FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array
FPU	Floating Processor Unit
GMR	Giant Magneto Resistance
LFT	Linear Fractional Transformation
LHP	Left Half Plane
LMI	Linear Matrix Inequalities
LSB	Least Significant Bit
LTI	Linear Time Invariant
MIMO	Multiple Input Multiple Output
MIPS	Mega Instructions per Second
MISO	Multiple Input Single Output
MSPS	Mega Samples Per Second
PS	Processing System
PSD	Power Spectral Density
PZT	Lead Zirconate Titanate
RHP	Right Half Plane
RMS	Root Mean Square

SIMO	Single Input Multiple Output
SISO	Single Input Single Output
SNR	Signal to Noise Ratio
SPI	Serial Peripheral Interface
ssKF	steady state Kalman Filter
THD	Total Harmonic Distortion
TIA	Trans-Impedance Amplifier

CHAPTER 1

Introduction

To introduce to the topic of sensor fusion some preliminary words from D. Simon [34] about the problems and challenges in nano-positioning and control are reproduced in followings.

In order to grow spiritually we need to listen and learn from a variety of sources (...) because we can never know who, when and how will speak to us. (...) In that sense we need to be essentially open to the data that comes into our lives from others. But if we listen to everything that is within earshot we will 'tossed back and forth by the waves, and blown here and there by every wind of teaching'. We need to reject unhealthy data in order to prevent ourselves from being misled. (...) We need to strike a healthy balance between scepticism and acceptance of the views of others. That is, we need to exercise discernment in order to allow ourselves to be influenced by beneficial information while rejecting data that may be detrimental.
The band limited frequency response of a control system is similar to spiritual discernment. A control system needs to be responsive to input commands, yet it also needs to reject input signals that are outside of its desired bandwidth. A control system that rejects all input signals is clearly ineffective. Yet a control system that responds equally to all input signals will be 'tossed back and forth by the waves, and blown here and there by every wind of input signal.' The control system needs to strike a healthy balance between acceptance of desired inputs and rejection of extraneous inputs.

Forgiveness is an essential part of ones. (...) Despite the widespread view, that forgiveness of an offense is equivalent to ignoring that offense, one could say that forgiveness consists of confronting the offense, recognizing it as the wrong that it truly is, seeking to benefit the offender, and consciously revoking any attempts at revenge. A person who refuses to forgive hurts himself more than the offender, for the unforgiving person allows a destructive root of hate and bitterness to grow inside him.
Noise suppression in control theory is similar to forgiveness. A control system that does not consider noise is incomplete. In fact, noise suppression can be considered as one of the primary purposes of feedback control. A control system design that ignores the

presence of noise might exhibit undesirable oscillatory behavior or even instability. The system might operate wonderfully in a noise-free environment, but the introduction of noise could render the system useless. A control system that is designed to perform well in the presence of noise is like (...) somebody who acknowledges the presence of sin in the world but does not allow it to ruin him. Just as (...) some deals with that sin in a constructive and active way, the control system is designed to minimize the effects of noise.

1.1 Motivation for filtering

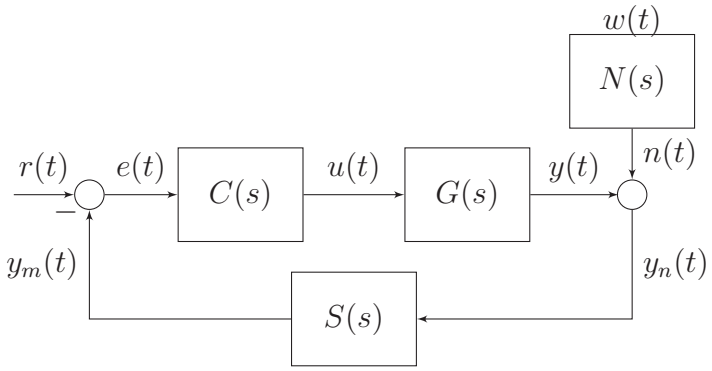


Figure 1.1: Feedback control loop with controller $C(s)$, system $G(s)$, sensor $S(s)$ and noise transfer function $N(s)$. Control input $r(t)$, measurement $y_m(t)$ and white noise $w(t)$.

Already out of the philosophical disquisition it can be recognized that sensor noise and sensor bandwidth play an important role for feedback control systems. To confirm that, a feedback control loop with controller $C(s)$, plant $G(s)$, sensor $S(s)$ according to Figure 1.1 is considered. s denotes the complex Laplace variable defined by $s = i\omega \in \mathbb{C}$. The colored noise[1] $n(t)$ arises from the sensor and its environment and noise transfer function $N(s)$ is usually identified by measurements. The sensor model $S(s)$ and that of the plant $G(s)$ are both either analytically derived or identified by measurements. The transfer function of the open loop for the presented case is given by

$$L(s) = C(s)G(s)S(s) \tag{1.1}$$

and that of the closed loop by

$$T_{ry}(s) = \frac{C(s)G(s)}{1 + L(s)} \tag{1.2}$$

[1] In contrast to white noise $w(t)$ which has theoretically unlimited bandwidth, the colored noise $n(t)$ is bandwidth limited. White noise only exists theoretically.

together with the transfer function of noise $w(t)$ to the position $y_n(t)$

$$T_{wy_n}(s) = \frac{N(s)}{1 + L(s)} \tag{1.3}$$

The two major requirements on the feedback control can be formulated intuitionally:

$$\textit{good} \text{ tracking of setpoint's magnitude} \qquad |T_{ry}(s)| \approx 1 \tag{1.4a}$$
$$\text{high suppression of noise} \qquad |T_{wy_n}(s)| \approx 0 \tag{1.4b}$$

The phase of the transfer function $T(i\omega)$, denoted by $\angle T(i\omega)$ can be obtained by solving the two trigonometric functions

$$\sin(\angle T(i\omega)) = \frac{\mathrm{Im}(T(i\omega))}{|T(i\omega)|}, \quad \cos(\angle T(i\omega)) = \frac{\mathrm{Re}(T(i\omega))}{|T(i\omega)|} \tag{1.5}$$

or as explicitly defined by equation (1.27). The requirements on the feedback control loop (1.4a) can be refined by additional criterion

$$\angle T_{ry}(\omega) \approx 0 \tag{1.6}$$

and expressed more precisely [19] by

$$\textit{good} \text{ tracking of setpoint} \qquad |T_{ry}(i\omega) - 1| \ll 1 \tag{1.7a}$$
$$\text{high suppression of noise} \qquad |T_{wy_n}(i\omega)| \ll 1 \tag{1.7b}$$

over wide range of frequencies $0 \leq \omega \leq \omega_h, \omega_h \gg 1$[2]. For a given configuration according to Figure 1.1, there is only the freedom to influence $T_{ry}(s)$ and $T_{wy_n}(s)$ by shaping the controller $C(s)$ to fulfill the requirements (1.7). By expressing (1.2) and (1.3) with general poles and zeros and substituting them into (1.7), it can be seen that (1.7a) is contradictory to (1.7b) in terms of the requirements on $C(s)$ and therefore, (1.7a) and (1.7b) can not be satisfied simultaneously. Additionally $C(s)$ must stabilize the system $G(s)$ if it is unstable [19].

The closed loop bandwidth ω_b is defined by

$$\omega_b = \min\{\omega \in \mathbb{R} : L(i\omega) = 1\} \tag{1.8}$$

i.e. where the open loop transfer function firstly crosses the 0 dB value[3]. One can argue that by shaping $C(s)$ and thus $L(s)$ the control bandwidth ω_b can be shifted towards higher frequencies. Basically, it is possible, however *extending* the control's bandwidth happens onto costs of the control signals magnitude $u(t)$.

The properties of a satisfying controller $C(s)$ (in respect to the closed loop) can be

[2] The notation $\varepsilon \ll 1$ is used to express ε is a small number, while $\omega \gg 1$ denotes usually a large frequency.

[3] It can be shown [19], that for each practical system with low-pass character $G(s)$ is strictly proper and for each realizable controller $C(s)$ is also strictly proper, so $\exists \omega_b : |L(i\omega_b)| = 1$

summarized as

$$\left.\begin{aligned} T_{ry}(s) &\approx \frac{1}{S(s)} \\ T_{wy_n}(s) &\approx \frac{N(s)}{L(s)} \end{aligned}\right\} \text{for } \omega \leq \omega_b$$

$$\left.\begin{aligned} T_{ry}(s) &\approx C(s)G(s) \\ T_{wy_n}(s) &\approx N(s) \end{aligned}\right\} \text{for } \omega > \omega_b$$

Out of the last line it can be seen that the noise $w(t)$ is transferred in full proportion to the position $y_n(t)$ for frequencies above ω_b without having any opportunity to influence it by the controller $C(s)$.

The aim of filtering the signal of a position sensor with transfer function $S(s)$ as part of a positioning system with actuation $G(s)$ and controller $C(s)$ can be summarized as follows.

1. For frequencies below ω_b positioning accuracy should be increased (possible by minimizing the effect of sensor noise).

2. Control loop's bandwidth should be extended until a certain magnitude of $u(t)$ which is still realizable.

Next to the freedom of controller design, the idea to take more sensors (m pieces) with transfer functions $S_1(s) \dots S_m(s)$ and generally unequal noise transfer functions $N_1(s) \dots N_m(s)$ to *profit of benefits of one or other* leads to investigate sensor fusion.

1.2 Motivation for sensor fusion

The desired position sensor has high resolution, high bandwidth, high linearity, low drift and it is robust against change in environmental parameters at affordable price *(Performance criteria)*.
It is imaginable that one single sensor can hardly fulfill all the listed criteria. This motivates to *fuse* sensors' data in order to emulate an ideal sensor complying the most of the performance criteria.

Since the measurement of absolute position is far more complex than measuring displacement (to my knowledge), displacement sensors are widely used and investigated. Besides that fact of complexity, for lots of applications only the relative position to an initial position is of interest (*displacement*). There are several principles to measure displacement, mostly based on different physical effects. Since different principles imply different advantages and disadvantages in properties of the sensor based on that principle, it is obvious to *combine* (consensually used *fuse*) the sensors in a way that mostly advantages of each are dominating after the fusion. Figure 1.2 illustrates the aim of sensor fusion with some possible displacement sensing principles with typical

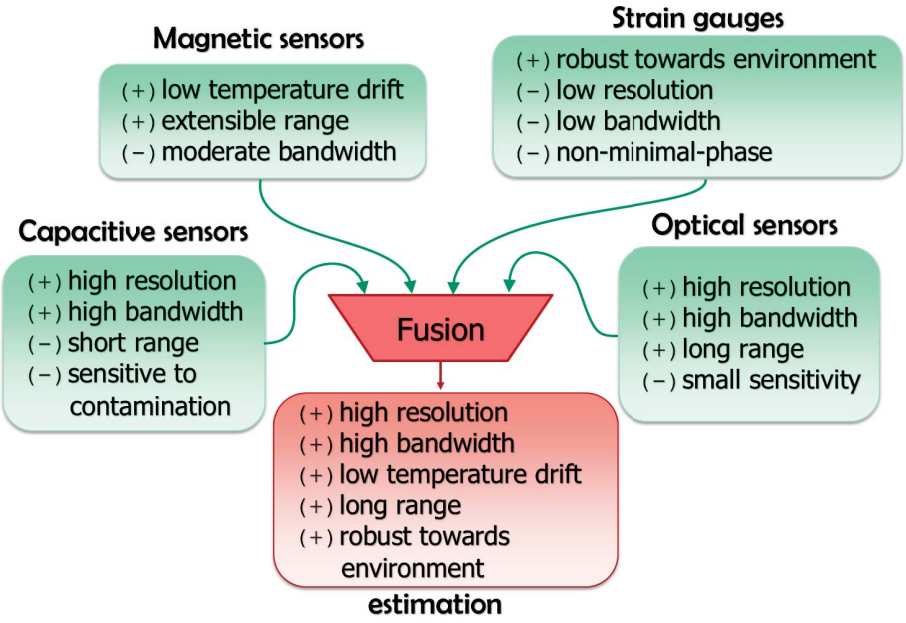


Figure 1.2: The aim of sensor fusion

properties[4]. Advantages (+) of each are wished to be own by the fused signal (called *estimation*).

Sometimes a single sensor has a bandwidth smaller than required for high dynamic actuation. Besides it, sensors often suffer from drift and deliver signals superimposed with noise, however not in equal amounts in general. E.g. one sensor has high drift, by owning also high bandwidth, compared to another one with low drift and high resolution at a very limited bandwidth. The aim of sensor fusion is to take advantages of each sensor by combining (filtering) them individually in order to provide a position estimation according to the Performance criteria.

[4] The reported properties are valid for a given configuration. It is not the essence here to formulate general statements about a class of sensors. The enumeration should be taken as an example illustrating the aim of fusion rather than a comparison between sensors.

1.3 Boundaries of the thesis

The control design is not subject of present thesis as for obtaining PID controllers with output feedback, or controllers with state feedback designed by either pole placement or by minimizing a quadratic cost function as the Linear Quadratic Regulator (LQR)[5], or the LQR augmented with a state estimator called Linear Quadratic Gaussian (LQG)[6], or other non-linear controllers[7], literature can be studied.

However, the design of Kalman filters are investigated in Chapter 3. Stationary, time-variant/extended, augmented with noise models, multiple input and various other types are discussed in Section 3.3. Besides it, other optimal filters such as the H_2- and H_∞-filters are discussed in Section 3.4. To accomplish the most of the Performance criteria for positioning systems (hereinafter only such are of interest) some norm of the noise transfer function $T_{wy_n}(s)$ (1.3) needs to be minimized. A filter which minimizes the noise energy (2-norm) is e.g. the H_2-optimal filter. Others, which are intended to minimize the noise power (∞-norm) [36] are called H_∞-filters. Both are handled in Section 3.4.4.

Mixed sensitivity filters minimizing a combination of norms are not handled in the thesis. As well as advanced methods of robust control applied on filtering problems such as e.g. the robustified Kalman filter [36] are also not handled neither constrained optimization or probability projection based filters such as the particle filter.

Due to the need for the detailed examination of sensor properties (such as noise distribution, bandwidth, resolution, etc.) in order to apply any filter, a characterization of different displacement sensors is presented in Chapter 2. For enabling the argumentation on the language of mathematics, basic concepts of linear dynamic systems and definitions enabling an evaluation of sensors are presented in Chapter 1, in followings.

1.4 Sensor

A sensor converts a physical quantity to another quantity in the electrical domain. Its operation is based on (an interaction of several) physical effects. For the design of experiments, for validation purpose, or for the examination of environmental influences it is essential to understand the entire operation principle of a sensor. Literature, wherein mostly theoretical characteristics of sensors are reported, help to design a measurement system. Experiments need to be made for examining side effects, e.g. for determining resolution or acquiring drift. Real sensors have always limited bandwidth and other undesired effects, e.g. increased sensitivity for some frequencies (*peaking*) or damping of others (insensibility, *notching*).

Key parameters of distance sensors are:

- range

[5] LQR is an optimal linear controller minimizing the 2-norm of the state estimation error (rather the mean of the squared error over all frequencies) if all states of a dynamic system are measured (or accessible).

[6] LQG is an LQR controller and an energy-optimal state estimator (Kalman filter) in one.

[7] Non-linear controllers might perform better than linear ones, but there is no guaranty in general.

- bandwidth
- resolution
- drift
- linearity

which can be quantified and which are defined in Section 1.4.1 to 1.4.5. A further parameter is *robustness*. It covers the ability to reject the impact of uncontrollable or non-deterministic environmental parameters, such as of electric, magnetic or electromagnetic fields of external sources (caused by e.g. a static discharge, mains, lightning or radio broadcast), ambient light, humidity, etc. It is defined differently for dissimilar sensors. One can surmise that an electric field mostly disturbs capacitive sensors (at least such without a *shield ring*) or ambient light influences optical sensors. Of course there are sources of disturbance influencing multiple sensors similarly e.g. they can originate from a common power supply. A summary of effects of electrical noise and precautions are given in Section 1.4.4.

1.4.1 Range

A definition for the *range of measurement* for a given sensor is not obvious. Most sensors have non-linear correlation between input and output. Generally, the limited range with linear transmission (or acceptable non-linearity) denotes the *range of linearity*, although the sensor also delivers some output with noticeable change beyond it (excluding the range clipping or saturation). In followings, range stays for the *total range* (unless otherwise noted) where the sensor delivers any output.

The range is measured in units of meters for displacement sensors. Usually there is a trade-off between sensitivity and range for some sensors, e.g. for photodiode based optical proximity sensors. In some configurations, a lower amplification of the photocurrent (lower sensitivity) increases the range of measurement and simultaneously lowers resolution and bandwidth, too.

1.4.2 Bandwidth

The bandwidth is identified with the frequency range wherein a sensor is designed to operate. It is denoted by $\mathscr{F} = [f_l, f_c]$, where f_l is the lower limit of frequency (in some cases 0 Hz) and f_c is the upper limit. Usually, there are two simultaneous definitions for f_c in context of sensors operating in a control loop.

1. f_c is the frequency at *half power* where the magnitude of power firstly drops by $10\log(1/2) \approx 3\,\mathrm{dB}$. It is equivalent to a drop of amplitude to 70.7 % since $20\log(1/\sqrt{2}) \approx 3\,\mathrm{dB}$. In this case it is denoted by $f_{-3\,\mathrm{dB}}$.

2. Sensors with noticeable dead time mostly causes phase lag $\varphi(f)$. In most cases, sensors will be used in closed loop with controller and actuator. The frequency f_c is defined at which firstly the phase reserve of open loop $\varphi_r = 70°$ to negative feedback ($-180°$) is consumed. Since amplifiers, actuators and other parts in the

closed loop also causes phase lag in general and the remaining 110° has to be divided between those and the sensor, i.e. that of sensor is limited de facto to approximately $\varphi(f_c) > -30°$ phase lag in the context of this thesis. In this case the frequency is denoted by f_φ.

1.4.3 Resolution

The resolution of a sensor is a key value for almost all applications in nanopositioning. Assuming gaussian (normal) distribution of sensors' noise, the $\pm 3\sigma$-resolution is defined as the equivalent range in displacement wherein 99.7 % of samples fall within [12]. A single value for resolution is always related to a measurement bandwidth. Due to diversity of applications where sensors are used for measuring dynamic actuation different bandwidth may be the subject of interest. Therefore, as in [11] wherever it is possible, resolution is reported versus bandwidth.

The resolution of a sensor is primary defined by electrical noise. A noise density is directly captured over the frequency by spectrum analyzers. By integrating the density with respect to a frequency range the band power and also the noise voltage related to that bandwidth can be obtained (1.12) which is in proportion to the resolution by the sensitivity factor.

1.4.4 Noise

The resolution of sensors is limited by electrical noise. Intrinsic noise mainly consists of thermal noise in resistors, flicker (1/f) noise arising from resistance fluctuations in a current carrying layer, and shot noise caused by generation and recombination processes in semiconductors. All of the mentioned noise sources are uncorrelated, so the total noise power[8] is

$$p(t) = v^2(t) = \sum_{i=1}^{M} v_i^2(t) \tag{1.10}$$

where v_i^2 is the theoretical noise power of the i-th (altogether M) noise source.

The huge drawback of acquiring the noise voltage (or noise power) in the time domain is, that it strongly depends on the measurement bandwidth. Only noise voltages with the same measurement bandwidth are comparable, and there is no information about *how much it would be lower when limiting the bandwidth of measurement further.* Usually, the density of noise power $p(f)$ is called PSD and it is captured in the frequency domain in units of power per 1 Hz.It is as useful to acquire the noise power density, as out of the PSD the noise power and hence, the noise voltage for any bandwidth can be calculated (which is included in the acquired PSD).
The limited bandwidth of a measurement is denoted by the interval

$$\mathscr{F} = [f_l, f_u] \qquad 0 \leq f_l \leq f_u < \infty \tag{1.11}$$

[8] It is intended to call a squared noise voltage v^2 as noise power p if the reference impedance = 1 Ω.

and the noise power for the measurement bandwidth $\mathscr{F}$ denoted by $p(\mathscr{F})$ is given by the integral over the density as follows

$$p(\mathscr{F}) = \int_{\mathscr{F}} p(f)\, df \overset{(1.11)}{=} \int_{f_l}^{f_u} p(f)\, df \tag{1.12}$$

$p(\mathscr{F})$ is also called the cumulative noise power or band power because of its calculation [12].

To obtain the noise voltage $v(\mathscr{F})$ linked to the bandwidth $\mathscr{F}$, the square root is taken

$$v(\mathscr{F}) = \sqrt{p(\mathscr{F})R} \tag{1.13}$$

with R as the reference impedance. Care should be taken to calculate the noise voltage with the same R as the power is measured on. To my knowledge, the reference impedance is often $50\,\Omega$. $v(\mathscr{F})$ is also called the cumulative noise voltage in analogy to $p(\mathscr{F})$.

Practical considerations

The PSD $p(f)$ is acquired in units of $1\,\mathrm{W/Hz}$ (SI-units) or logarithmically scaled in $1\,\mathrm{dBW/Hz}$ (referred to $1\,\mathrm{W}$) or in $1\,\mathrm{dBm/Hz}$ (referred to $1\,\mathrm{mW}$). By taking the square root of $p(f)$ in units of $1\,\mathrm{W/Hz}$ according to (1.13) the noise voltage density (or noise current density) $v(f)$ in units of $1\,\mathrm{V}/\sqrt{\mathrm{Hz}}$ ($i(f)$ in units of $1\,\mathrm{A}/\sqrt{\mathrm{Hz}}$) can be calculated. As in practice, due to the finite resolution of the acquired PSD, the integral in (1.12) can be replaced by the sum. Assuming the vectors $\mathbf{f} \in \mathbb{R}^N$ of N frequency points and corresponding vector $\mathbf{p} \in \mathbb{R}^N$ of N power spectral density points,

$$\mathbf{f} = [\underbrace{f_l \dots f_u}_{N}] = [f_1, f_2, \dots, f_{N-1}, f_N], \qquad \mathbf{p} = [\underbrace{p(f_l) \dots p(f_u)}_{N}] = [p_1, p_2, \dots, p_{N-1}, p_N] \tag{1.14}$$

the following equation summarizes (1.12)…(1.14) expressing the cumulative noise voltage $v(\mathscr{F})$ in units of $1\,\mathrm{V}$ out of the PSD measurement in SI-units of $1\,\mathrm{W/Hz}$

$$v(\mathscr{F}) \approx \sqrt{\sum_{i=1}^{N-1} p_i \; R \; (f_{i+1} - f_i)} \tag{1.15}$$

with linearly or logarithmically spaced vectors $\mathbf{f}$ and $\mathbf{p}$. It is advisable to chose the logarithmic spacing when the noise voltage is plotted versus the logarithmic frequency axis to have an even density of information. An efficient way to compute (1.15) can be done by the cumulative sum function `cumsum()` in MATLAB[9].
The conversion between typical units of the PSD can be accomplished with the following correlations

$$p(\mathscr{F})\big|_{\mathrm{dBm/Hz}} = p(\mathscr{F})\big|_{\mathrm{dBW/Hz}} + 30\,\mathrm{dB} = 10\log(p(\mathscr{F})\big|_{\mathrm{W/Hz}}) \tag{1.16}$$

$$p(\mathscr{F})\big|_{\mathrm{W/Hz}} = 10^{\frac{p(\mathscr{F})|_{\mathrm{dBm/Hz}}}{10}}\, 10^{-3} = 10^{\frac{p(\mathscr{F})|_{\mathrm{dBW/Hz}}}{10}} \tag{1.17}$$

[9] Matlab (Mathworks, Massachusetts, USA)

Regarding the information content, the density (either of power or of voltage) plotted over the frequency gives an impression of the spectral distribution of noise, as the frequency and magnitude of single-frequency disturbances are separable[10], which effect the measurement. However, the cumulative noise voltage (1.15) of a sensor, is in proportion to its resolution by the so called sensitivity factor.

Challenges

Especially the very low frequency part of flicker noise (approximately below 3 Hz) contributes to limit the resolution and it is very time-consuming to acquire adequately. Moreover, it cannot be totally separated from drift.

1.4.5 Drift

Drift is a comparatively long-term change of a sensor signal. It may be unidirectional or periodic. If such a change is observable at constant and homogeneous temperature distribution ($\nabla\vartheta = \text{const.}$[11] in time) temperature drift is excluded. In this case drift is assumed to arise from other sources than a change in temperature. E.g. for lots of sensors aging produces also a drift on a long-term scale. A drift is commonly measured in the time domain at constant temperature (de facto with temperature recording) over several hours to days and reported as peak-to-peak value with respect to the acquisition time range.

To determine the temperature drift of a sensor system, it gets uniformly heated up and cooled down by simultaneously recording the sensor's signal $\mathbf{v} = [v_1, v_2, \ldots, v_n]^T$ and the temperature $\boldsymbol{\vartheta} = [\vartheta_1, \vartheta_2, \ldots, \vartheta_n]^T$. The change in the sensor's signal and the change in temperature are compared and in case of a high correlation e.g. $|\operatorname{corr}(\mathbf{v}, \boldsymbol{\vartheta})| > 0.95$ the temperature dependence TD can be given e.g. in units of $1\,\mathrm{V/°C}$[12].

Methodology

Long term recordings of voltages and currents are made using multiple 6.5 digit multimeters with logging capability as the TrueVolt® 34461A[13]. They are taken because of having a high resolution of $1\,\mu\mathrm{V}$ (at 0.25 Hz BW), sampling rates from 10 Hz down to 1/60 Hz by storing 10^4 samples over 16 minutes to almost 7 days, optional filters for mains, auto zero comparison, synchronization capability, etc.

1.5 External disturbances

In addition to intrinsic noise sources (as summarized in Section 1.4.4), there is a variety of non-essential noise sources of external disturbances, often recognizable on PSD measurements. These are typically the power line and its harmonics (usually dominant below 1 kHz), AM broadcast stations at several 100 kHz, FM and TV broadcast at

[10] Often external disturbances, such as mains and its harmonics, appear as spikes on the spectrum.
[11] ∇ denotes the gradient of some quantity in directions of space.
[12] However, correlation does not imply causation in general.
[13] Agilent Technologies, Santa Clara, CA, USA

several 100 MHz or even radar at some GHz [39]. Some of these are as distinct that they appear in a spectral measurement above the intrinsic background noise as e.g. spikes of mains and its harmonics with up to +10 dB (in power) above noise floor for strain gauge sensors as shown in Figure 2.18. There are several ways to lower the effect of these non-essential noises, summarized in followings (based on hints in [39]).

1. A voltage of an isolated equipment or an operator can couple to a sensor via small stray capacitances. Therefore, voltages should be measured with low impedance and currents with high impedance. An additional metal shielding connected to earth lowers the entering of travelling charges to the sensitive circuit and thus, it lowers capacitive coupling.

2. A rapidly changing current in a nearby circuit or mains alternating current produces a magnetic field which induces a voltage in surrounding conductors according to law of induction. To lower the effects, either parallel wires close to each other (this configuration reduces the area of the pick-up loop), or twisted pair wires (it alters the orientation of pick-up loop and induced currents cancel out each other), or coaxial cables (effect of cage of Faraday), or perhaps a magnetic shield (for high frequency even a thin metal enclosure is adequate due to the Skin-effect) should be used.

3. By allowing parallel paths (distributed in space) for a current to flow, a huge pick-up loop (ground loop) can be formed enabling induction of currents. These currents cause a voltage drop on a wire with finite conductivity, which is superimposed to a signal. This can happen e.g. by the unwanted earthing of the same potential of a circuit at multiple locations. Therefore, grounding at one single point (stair-point) should be used.

4. Microphonics provides a path for mechanical noise to appear as electrical noise in a circuit [39]. The capacity of a coaxial cable is a function of its geometry so mechanical vibrations are transformed into the electrical domain.

5. Voltage created by dissimilar metal junctions (as used in thermocouples) can falsify measurements by adding varying voltages depending on environmental parameters.

One can see how careful a measurement needs to be done in order to provide reproducible results. In practice it is impossible to consider everything what influences the measurement. The only way to overcome the problem is to assess the effect of *the one and other*. An analytical investigation by keeping a close eye to the neglected effects is indispensable.
Next to the properties of sensors, their usage is illustrated in one emphasized application in the field of nano-positioning, in followings.

1.6 Atomic Force Microscopy

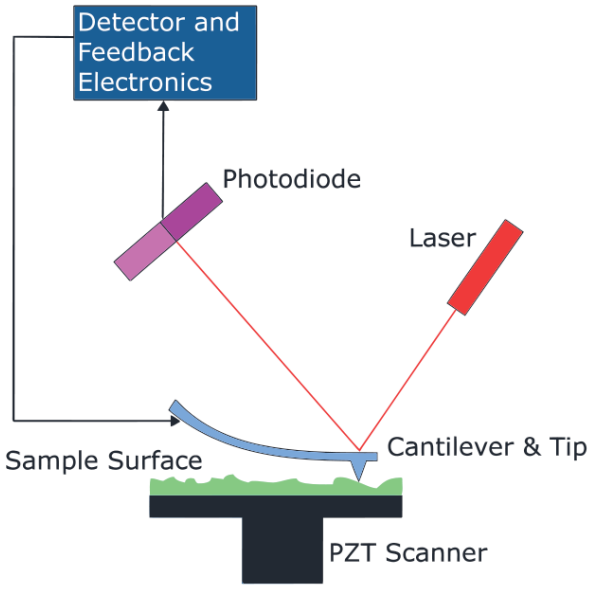


Figure 1.3: Operation principle of an Atomic Force Microscope (AFM) [9]

An Atomic Force Microscope (AFM) allows to image topography of a sample even on the atomic scale. Individual molecules' position, shape, charge and other properties can be acquired with resolution comparable to electron beam microscopy with much less effort in sample preparation [9].

It differs from optical microscopes since it does not acquire an image by focusing light onto a surface. Inside an AFM a very sharp tip approaches the surface of the sample. This tip is *mounted* on the free end of a one side chucked flexible cantilever. The attractive and repulsive forces interact with the tip on atomic level, bending the cantilever. As illustrated in Figure 1.3 the deflection of the cantilever can be measured by the optical beam deflection method (containing a laser source and a photodiode array detector). By the lateral moving of the sample together with the deflection information of the cantilever a three dimensional map of the sample's topography can be acquired.

In order to produce the lateral moving scanner tables (positioning stages, nanopositioners) are needed with actuation ranging from several nanometers to hundreds of micrometers in multiple directions at high speeds and at high accuracy. It is a key issue to measure position or displacement with very low time delay at high bandwidth for closed loop nanopositioners [12],[48], although there exist some positioners operating in open-loop without a need for a feedback [20], [47].

1.7 Dynamic system

A Linear Time Invariant (LTI) Single Input Single Output (SISO) dynamic system can be represented in continuous time with n differential equations of first order in the state-space representation

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t) \tag{1.18a}$$

$$y(t) = \mathbf{c}^T\mathbf{x}(t) + du(t) \tag{1.18b}$$

with state vector $\mathbf{x}(t)$ initial conditions $\mathbf{x}(0) = \mathbf{x}_0$, time derivatives $\dot{\mathbf{x}}(t) = \partial\mathbf{x}(t)/\partial t$, one-dimensional input $u(t)$, one-dimensional output $y(t)$ and matrices of appropriate dimension, system matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, input matrix $\mathbf{b} \in \mathbb{R}^n$, output or measurement matrix $\mathbf{c}^T \in \mathbb{R}^n$ and feed-through matrix $d \in \mathbb{R}$ can be written[14].
Parameter t denotes time dependence equivalent to subscript k for time discrete systems. Matrices with lack of both are constant in time, called LTI.

1.7.1 Transfer function

The Laplace transform is defined by the bijective one-side integral transformation

$$X(s) = \mathcal{L}(x(t)) = \int_0^\infty x(t)e^{-st}\,\mathrm{d}t, \quad s = i\omega \in \mathbb{C} \tag{1.19}$$

which transforms the real time signal $x(t)$ into the complex frequency plane $X(s)$ [19].

By Laplace transforming (1.18), expressing $X(s)$ and substituting (1.18b) $Y(s)$ can be written

$$Y(s) = \mathbf{c}^T(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{x}_0 + \left(\mathbf{c}^T(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} + d\right)U(s) \tag{1.20}$$

with $U(s) = \mathcal{L}(u(t))$, $Y(s) = \mathcal{L}(y(t))$ as Laplace transformed of input and output respectively. $G(s)$ denotes the transfer function of (1.18) since

$$G(s) = \frac{Y(s)}{U(s)} = \mathbf{c}^T(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{b} + d \tag{1.21}$$

with identity matrix $\mathbf{I}$. It is the representation of the relation between input and output of an LTI system with zero initial conditions $\mathbf{x}_0 = \mathbf{0}$ and it is for many practical systems a broken rational function. For every system given in state space representation (1.18) transfer function $G(s)$ according to (1.21) can be expressed, although, generally not the opposite way.

[14] Superscript T denotes transpose of a matrix, bold capitals ($\mathbf{A}$) indicates matrices, bold lower case symbols ($\mathbf{b}$) denotes column vectors ($\mathbf{c}^T$ row vectors) and italic letters (u) are representing scalars. The notation to indicate that real valued matrix $\mathbf{M}$ has g rows and h columns by assigning it as element of space of real numbers of dimension $(g \times h)$ denoted by $\mathbf{M} \in \mathbb{R}^{g\times h}$ is legal if some conditions about its eigenvalues [19] are met.

A broken rational transfer function

$$G(s) = \frac{b(s)}{a(s)} = \frac{\sum_{i=0}^{q} b_i s^{q-i}}{\sum_{i=0}^{r} a_i s^{r-i}} \tag{1.22}$$

with polynomials $b(s)$ and $a(s)$ is called *proper* if the order of nominator q and order of denominator r satisfy condition $q \leq r$. $G(s)$ is *strictly proper* if $q < r$ which is a *convenient* property in some cases, it excludes a direct feed-through of input to output ($d = 0$ in state space realization). It is consensual to *realizable*.

Poles p of the transfer function $G(s)$ are zero places of the characteristic polynomial of the system, $p = \{s : a(s) = 0\} \in \mathbb{C}$, while *zeros* of transfer function $G(s)$ are zero places of nominator polynomial $z = \{s : b(s) = 0\} \in \mathbb{C}$, respectively.

Eigenvalues of the matrix $\mathbf{A}$ are poles of the system $p = \mathrm{eig}(\mathbf{A})$. Eigenvalues are the solution of the equation $\det(\mathbf{A} - s\mathbf{I}) = 0$ where $\det(\cdot)$ denotes the determinant of a square matrix. Next to the evaluation of eigenvalues of the system in state space realization (1.18), the corresponding transfer function (1.21) can be visualized on a Bode diagram[15].

A dynamic system is called *bounded-input bounded-output stable* or *BIBO stable* if corresponding transfer function (1.22) has only poles in the Left Half Plane (LHP), i.e. poles with negative real part $\mathrm{Re}(p) < 0$.

A transfer function is called *non-minimal-phase* if at least one zero with positive real part $\exists z_i : \mathrm{Re}(z_i) > 0$ exists. This property implies that the step response shows an initial reversed answer than the excitation signal, which is in case of sensors mostly undesired.

Rating of transfer functions

Sometimes it is required to quantify the *variability* of a transfer function. Hence, a norm has to be defined. The infinity-norm (denoted by $\|\cdot\|_\infty$) of a transfer function $G(s)$ expresses the maximum of magnitude over all frequencies ω (in absolute values)

$$\|G(i\omega)\|_\infty = \max_\omega |G(i\omega)| \tag{1.23}$$

The variability of a transfer function, denoted by the Δ operator, is defined as the difference between its maximum and minimum for a limited frequency range $\mathscr{F} = [f_1, f_2]$ according to

$$\Delta G(\mathscr{F}) = 20 \log \left(\|G(i\omega)\|_\infty \, \|G^{-1}(i\omega)\|_\infty \right), \quad \left(\frac{\omega}{2\pi}\right) \in \mathscr{F} \tag{1.24}$$

in units of 1 dB, assuming that $G^{-1}(s) = 1/G(s)$ exists ($G(s)$ must be a broken rational function (1.22), however $G^{-1}(s)$ does not need to be realizable $\equiv$ strictly proper). Figure 1.4 illustrates the relations.

[15] after Hendrik Wade Bodé

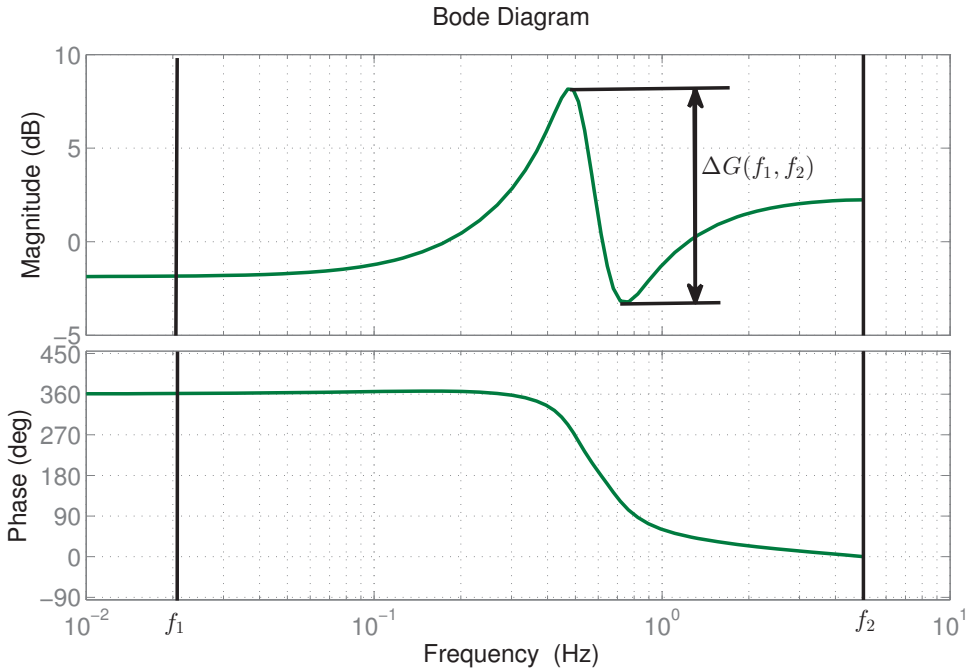


Figure 1.4: Illustrative example to definition of transfer function's *smoothness*

A transfer function $T(s)$ is called *ideal* in the context of sensors, in the frequency range $\mathscr{F}$ if

$$T(i\omega) \approx \text{const.}\ \forall\omega : \left(\frac{\omega}{2\pi}\right) \in \mathscr{F} \iff \Delta T(\mathscr{F}) \ll 1\,\text{dB (small)} \tag{1.25}$$

1.7.2 Bode diagram

A Bode diagram consists of two plots below each other. On the upper one, absolute values of the magnitude of a transfer function are plotted

$$G(f) = 20\log(|G(s)|),\ s = i2\pi f \qquad G(f) \text{ in units of } 1\,\text{dB} \tag{1.26}$$

versus a logarithmically scaled frequency on the ordinate (horizontal axis), in units of 1 Hz. The phase is given by the four quadrant inverse tangent refining (1.5) to

$$\varphi(f) = \operatorname{atan2}(x, y) \begin{cases} \arctan(y/x) & x > 0 \\ \arctan(y/x) + \pi & y \geq 0, x < 0 \\ \arctan(y/x) - \pi & y < 0, x < 0 \\ \pi/2 & y > 0, x = 0 \\ -\pi/2 & y < 0, x = 0 \\ \text{undef.} & y = 0, x = 0 \end{cases} \tag{1.27}$$

with $x = \operatorname{Im}(G(s))$ and $y = \operatorname{Re}(G(s))$

It is drawn in the lower plot versus the same logarithmically scaled frequency axis, as can be seen e.g. in Figure 1.4. In case of sensors, the transfer function is commonly

interpreted as *sensitivity over the frequency*.

One real pole in the Left Half Plane (LHP) ($p < 0$) causes a slope of transfer function's magnitude by $-20\,\mathrm{dB/decade}$ of frequency increase above $2\pi|p|\,\mathrm{Hz}$, while LHP poles with multiplicity of m causes a slope by $-20m\,\mathrm{dB/decade}$. However zeros z with multiplicity m causes a rise of transfer function's magnitude by $+20m\,\mathrm{dB/decade}$ above $2\pi|z|\,\mathrm{Hz}$.

Considering only the phase it can not be stated whether the change by $-90m^\circ$ (phase lag) is caused by an LHP pole or Right Half Plane (RHP) zero as well as it is not clear whether the change by $+90m^\circ$ (phase lead) is coming from a RHP pole or LHP zero of multiplicity m, respectively. As consequence, it can be concluded, that the Bode diagram does not acquire whether a system is BIBO-stable or not, this can be seen e.g. out of the step response in the time domain or by the evaluation of the corresponding poles in state-space realization.

Methodology

Acquiring Bode diagrams can be done directly using a system analyzer (rather than recording signals in time domain). The system analyzer measures the root mean square (RMS) of the sensor's output in a varying narrow frequency band (resolution bandwidth RBW) while exciting the system with a sine of increasing frequency at constant amplitude. For presented Bode diagrams a network analyzer HP4395[16] is used in *swept sine* mode.

Attention needs to be taken for each component of the overall system to excite only one frequency at a time. In general, due to non-linearity of components such as amplifiers or actuators the Total Harmonic Distortion (THD) is acquired in some cases and the expressiveness of the Bode diagram is characterized by it. For some characterized sensors, especially at high frequencies power spectrum of third and fifth harmonics are suppressed by 7...10 deciBels to carrier (dBc) only. In such case the presented transfer function should be taken with care, since it does not describe the real (non-linear) behavior of the (assumed) linear sensor anymore.

Transformations

A time signal $g(t)$ is related to its spectral representation $G(f)$ by the linear bijective Fourier transform defined by

$$G(f) = \mathcal{F}(g(t)) = \int_{-\infty}^{\infty} g(t)e^{-i2\pi ft}\,\mathrm{d}t \tag{1.28}$$

In the case $g(t)$ is the impulse response of a sensor, $G(f)$ is its transfer function. Both hold the same information. The transformation is denoted by $g(t) \circ\!\!-\!\!\bullet\, G(f)$. Table 1.1 summarizes other important correspondences[17].

The continuous Fourier transform (1.28) converts the time domain signal $g(t)$ of infinite duration into a continuous spectrum $G(f)$ composed of an infinite number of sinusoids. For real-valued time signals the spectrum is symmetric around 0 frequency,

[16] former Hewlett Packard, Agilent Technologies, Santa Clara, CA, USA
[17] The distribution $\delta(f)$ denotes the Dirac delta function

so $G(f)$ holds no new information for $f < 0$. This, and practical considerations, such as the finite signal duration motivates to use the Discrete Fourier Transform (DFT) instead of (1.28). The finite length time signal g_k, $k = 0, 1, \ldots N - 1$ is transformed into the complex, discrete frequency domain by the N-point DFT

$$G_k = \sum_{l=0}^{N-1} g_k e^{-i2\pi lk/N} \tag{1.29}$$

which is usually computed by the M-point Fast Fourier Transform (FFT) for $M = 2^{\mathbb{N}^+} > N$. Usually, some windowing function (e.g. the hanning window) needs to be taken to pretend no periodicity for a transient signal, and the technique of zero padding is used to achieve higher spectral resolution ($M/2$ frequency points instead of $N/2$).

time domain		frequency domain
1	$\circ\!\!-\!\!\bullet$	$\delta(f)$
$\delta(t)$	$\circ\!\!-\!\!\bullet$	1
$\mathrm{rect}(t)$	$\circ\!\!-\!\!\bullet$	$\dfrac{\sin(f\pi)}{f\pi}$
$\mathrm{tri}(t)$	$\circ\!\!-\!\!\bullet$	$\dfrac{\sin^2(f\pi)}{f\pi}$
$\sin(2\pi f_0 t)$	$\circ\!\!-\!\!\bullet$	$\dfrac{1}{2i}\big(\delta(f - f_0) - \delta(f + f_0)\big)$

Table 1.1: Fourier correspondences [19]

CHAPTER 2

Sensor characterization

In order to validate analytically known models of sensors and to investigate major influences of environment, an actuator has to be taken to which sensors can be aligned in order to measure displacement as illustrated in Figure 2.1.

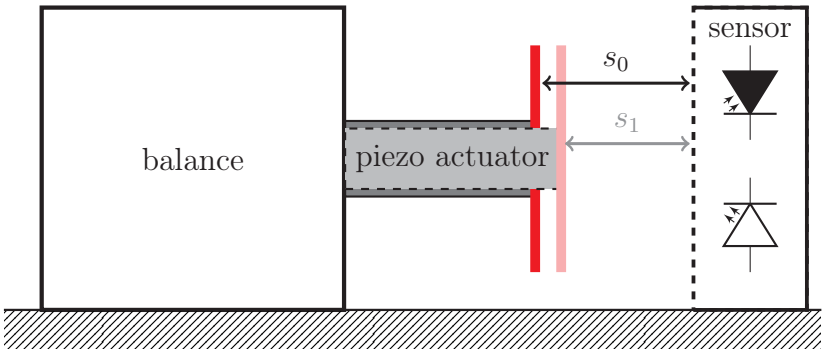


Figure 2.1: Test setup consisting of piezo actuator (on the one side fixed to a balance with large mass) with an actuated platform mounted on it (red indicated plate) and a fixed displacement sensor (symbolically an optical proximity sensor) measuring displacement $s_1 - s_0$.

2.1 Test setup

For an expected sensor bandwidth of 200 kHz and above [17], low-mass high-stiffness piezoelectric actuators (commonly known as Lead Zirconate Titanate (PZT) actuators or piezos) are investigated, because they are supposed to be able to excite the platform at such high frequencies by several hundred nanometers at least. The configuration in

Figure 2.1 is not the only possible arrangement of sensor and its counterpart[1], it illustrates only one possible method of displacement measurement using optical proximity sensors.

The piezoelectric effect[2] can be observed as an electric potential in response to compression of an anisotropic quartz crystal, such as PZT. The inverse effect, when applying a voltage on electrodes of such crystals, an elongation or contraction occurs which is the principle of piezoelectric actuators. Similar to the materials that permanent magnets are made of, molecules of piezoelectric materials with the same polarization direction are grouped in Weiss domains. For manufacturing piezoelectric materials the Weiss domains are aligned in a high electrostatic field and remain in this while cooling them down from above the Curie temperature to room temperature, in order to get a remnant polarization. Therefore, when using PZT actuators, care should be taken not even to approach Curie temperature because depolarization occurs and actuators are destroyed [17].

2.1.1 Mechanical considerations

Ceramic materials, such as PZT of which piezoelectric actuators consist of, have a very high stiffness, in order of 200 N/μm for the NAC2025[3]. Thus, they transmit vibrations of the balance to the platform and vice versa. Due to the own mass of the actuator, which is also partly accelerated during actuation, remarkable forces are induced into the balance while driving the actuator especially at high frequencies. If the balance is not able to damp excitation forces adequately, they get reflected and are superimposed in the actuator causing resonances. For a piece of aluminium acting as balance, it turned out that many resonances over a wide frequency range occur, i.e. the frequency response of the actuator is superimposed with peaks (*nodes and anti-nodes*), as it can be seen in Figure 2.2. The figure compares the same actuator NAC2025 mounted consecutively on two different balances. These two are

1. a piece of aluminium bar with
   $(40 \times 40 \times 535)$ mm size and 2.31 kg weight

2. and a heavy granite stone with
   $(35 \times 15 \times 50)$ cm size and 71 kg weight.

It can be clearly seen in Figure 2.2 that taking that of granite (2) brings benefits if low order models have to be fitted to the recorded frequency response.

The granite balance suppresses the excitation forces of the piezoelectric actuator sufficiently. The first resonance mode of the piezo is observable at 41 kHz with +10 dB raise and higher modes at ≈140 kHz and ≈300 kHz. Residual vibrations of the balance

[1] Also the sensor can be actuated and its counterpart can be fixed. Usually magnetic sensors are less heavy than the permanent magnet or coil which acts as source of magnetic field, therefore sometimes the sensor is actuated.
[2] Observed by Pierre Curie in 1880 [26]
[3] Noliac, Berlin, Germany.

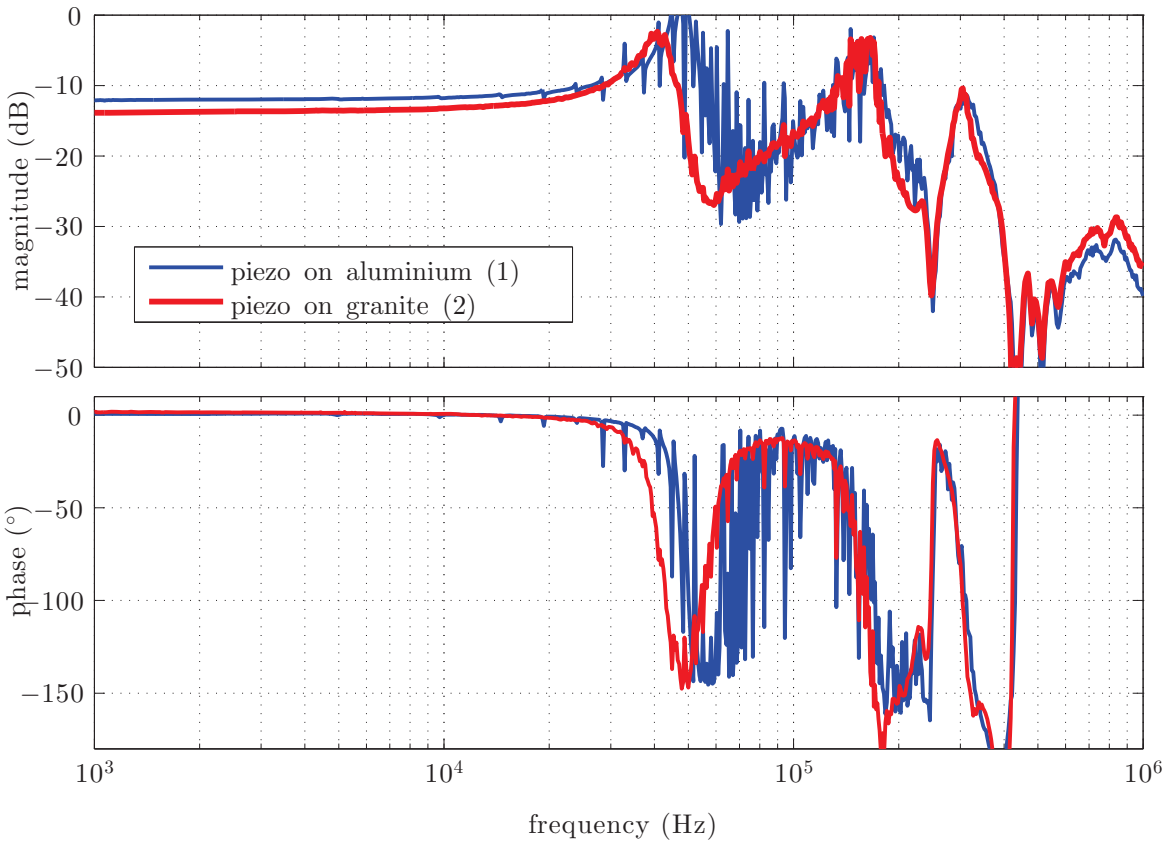


Figure 2.2: Frequency response of the piezoelectric actuator NAC2025 (voltage to displacement) on different balances.

are at least 30 dB damped for $f < 40\,\text{kHz}$ as it can be seen in Figure 2.6. The theoretical maximal force seems to be as high as $F = 390\,\text{N}$ at $A = 10\,\text{nm}$ displacement at $f = 1\,\text{MHz}$ frequency with $m = 1\,\text{g}$ of actuated mass and acceleration according to $a = A(2\pi f)^2 \sin(2\pi f t)$ which might be the reason for the excitement of granite block due to $1/71000$ mass ratio.

With a maximum of stationary deflections of $8 \ldots 20\,\mu\text{m}$ [43] stacked piezos (Figure 2.4) have first resonance frequency ($\omega = \sqrt{k/m}$) in the range of 150 kHz to 30 kHz depending on their mass m and stiffness k. Measurements confirm that piezo excites the balance during actuation. The energy of oscillation is partly introduced into the balance, propagates as a longitudinal wave and gets reflected from the counter-side (parallel plane to that where the piezo is mounted on) while getting damped (as outlined in [17] and partly in [10]).

On the one hand, it can be shown empirically that the material, the shape and dimensions of the balance influence how many resonances with which magnitude occur. On the other hand, the connection between piezo actuator and balance plays a role. There is a widespread literature about how to glue *optimally* a PZT actuator on different surfaces [17], [48]. It can be observed, that major differences in the number, position and magnitude of resonances evolves depending on the glue which was used to

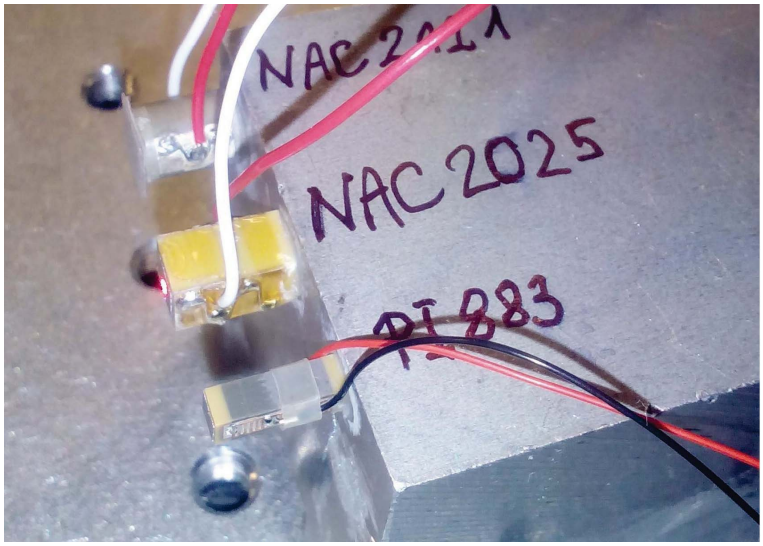


Figure 2.3: Stacked piezo actuators of different manufacturers prepared for characterization by a vibrometer

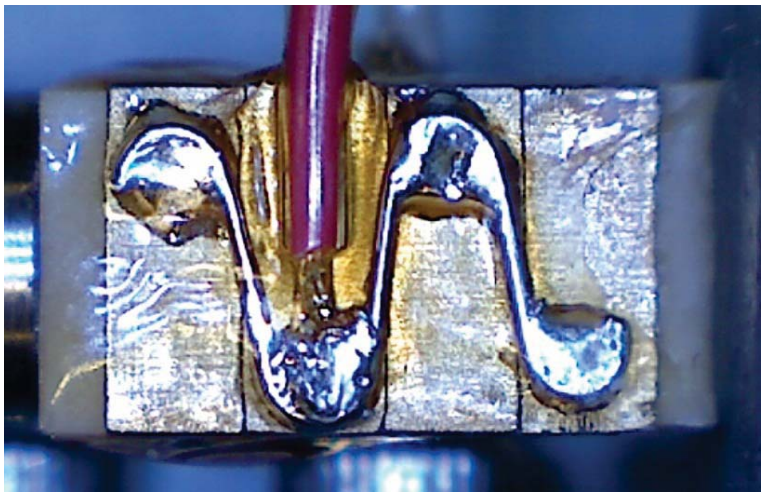


Figure 2.4: Stacked piezoelectric actuator NAC2025 consisting four elements in mechanically serial and electrically parallel connection. Dimensions $(5 \times 5 \times 10)$mm.

fix the actuator to the balance. According to my observations,

1. a glue with low stiffness (*soft glue*) adds additional damping to the frequency response and produces fewer and less distinct resonances. The first dominant resonance of the piezo ($\omega \approx \sqrt{k/m}$) is shifted towards higher frequencies. Soft glues are e.g. a double sided adhesive tape or e.g. *Palmatex*.

2. A glue with high stiffness (*hard glue*) tends to damp resonances fewer. The first resonance mode of the piezo is shifted towards lower frequencies. Hard glues are all thin-film glues, especially instant adhesives, cyanoacrlyate adhesives or epoxy glues.

Longitudinal wave theory

Theoretical considerations with sound or pressure waves (identical) show that although only small differences in material parameters between aluminium and granite persist, especially propagation speed for pressure waves c_p shows a remarkable difference as can be taken from Table 2.1. c_p should be related for the same sized balances inversely proportional to its eigenfrequency and thus to resonances observable on the piezo. The larger and longer the balance, the lower the eigenfrequency and coincidentally resonances measured on the free end of the piezo are less distinct which is desired and can be observed. This is assumed to happen while the wave is propagating over a longer path while it is getting damped. The increased mass of the balance might probably not be neglected, too. c_p and c_s are the propagation velocities for pressure and shear waves, respectively. They are given for a homogeneous three-dimensional solid by

$$c_p = \sqrt{\frac{K + \frac{4}{3}G}{\rho}}, \quad c_s = \sqrt{\frac{G}{\rho}} \tag{2.1}$$

with bulk modulus of the elastic material K, which describes the material's resistance to uniform compression ($K = \rho \ dP/d\rho$) [45]. Next to granite, concrete is of interest due to the lowest c_p of materials listed in Table 2.1.

A detailed modelling of the stacked piezo actuator and especially the glue layer between the balance and the actuator either or both numerically by Finite Element Method analysis (FEM) or analytically by solving Partial Differential Equations (PDE) of distributed springs and mass elements deflects the focus of the thesis.

material	ρ (kg/m^3)	E (GPa)	G (GPa)	c_p (m/s)	c_s (m/s)
aluminium	2700	70	25.5	6317	3073
granite	2650..2750	60..66	26	5117..5467	2180..2254
steel	7750..8050	210	79.3	5847..5959	3139..3199
concrete	2240..2500	30..48	15..27	3651..4880	1753..2719

Table 2.1: Mechanical parameters of some materials: specific density ρ, Young's modulus E, shear modulus G, propagation velocity for shear waves c_s and for pressure waves c_p (taken from [25], [45])

2.1.2 Amplifier

The simplest model of a piezoelectric actuator is a capacitor C_p where the voltage across it v_c corresponds to its elongation [20]. In order to drive a stacked piezoelectric actuator up to the maximum of its elongation several hundred Volts are necessary, which is provided by amplifier. Unfortunately, amplifiers have a finite output resistance R_o which forms with the capacitance of the piezo C_p a first-order low-pass filter, as it is modeled in Figure 2.5. v_c becomes lower with increasing frequency while simultaneously current i_c increases until a certain limit caused by the amplifier.

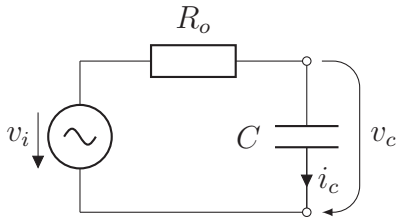


Figure 2.5: Equivalent circuit of a voltage amplifier and a piezoelectric actuator

At typical nominal parameters $R_o = 50\,\Omega$ (WMA-300 high voltage amplifier[4]) and $C_p = 400\,\text{nF}$ (NAC2025 10 mm long stacked piezo) the cut-off frequency ($-3\,\text{dB}$) is as low as

$$f_c = \frac{1}{2\pi R_o C_p} = 8\,\text{kHz} \tag{2.2}$$

The actuator is wished to be driven at constant amplitude of voltage v_0 over wide frequencies,

$$v_c = v_0 \cos(\omega t) + v_0 \tag{2.3}$$

as the voltage v_c is in good proportion to the elongation of the piezo above a lower frequency limit, however below that the effect of creep dominates (generally, the electrical charge in the actuator is in good proportion to its elongation over a wide range of frequencies [17]). As this is de facto not possible due to the limited bandwidth of the amplifier (detailed explanation in Section 2.1.2), the frequency response of the total system (amplifier and actuator) is measured by a sensor which is considered as ideal (reference measurement). The system's frequency response is corrected than by the reference measurement afterwards to pretend as if the actuator would be excited at constant amplitude of voltage over a wide range of frequencies. The reference measurement is provided by a vibrometer OFV-5000+OFV534[5] based on the optical Doppler-effect, hereinafter called vibrometer.

Because of the low pass model (Figure 2.5) the transfer function of the actuation (consisting of the amplifier and the piezoelectric actuator) from voltage v_i to the displacement measured by the vibrometer, shows for the driving voltage $v_i(\omega) = v_0 \cos(\omega t) + v_0$ with constant amplitude and offset $v_0 = 25\,\text{V}$, in the frequency range $[2\,\text{kHz}, 1\,\text{MHz}]$ a decay of $-20\,\text{dB/decade}$ superimposed with resonances of the piezo as it can be seen by the red colored frequency response in Figure 2.6. The first resonance mode is observable at $f_{r1} = 41\,\text{kHz}$.

The green colored response represents the vibrations of the granite balance, acquired by the vibrometer, too. There are possibly different sources of influence why the last one is not evenly low, close to the lower limit of the used network analyzer HP4395 with a dynamic range of 100 dB. At low frequencies $\mathscr{F}_1 = [10\,\text{Hz}, 500\,\text{Hz}]$ some peaks up to 30 dB variability ($\Delta T(\mathscr{F}_1) = 30\,\text{dB}$ according to definition (1.24)) are observable, originating possibly from a resonating environment. In the middle frequency range

[4] Falco Systems BV, Amsterdam, The Netherlands
[5] Polytech, Waldbronn, Germany.

$\mathscr{F}_2 = [4\,\text{kHz}, 11\,\text{kHz}]$ resonance and anti-resonance pairs (nodes and antinodes) are present causing the pattern visible repetitivly in the frequency response. The isolation (defined as damping of excitation forces initiated by piezo actuator) can be read out as difference between the two responses, and it lowers with increasing frequency, particularly it is 28 dB at f_{r1} where the first resonance mode of the piezo occurs.

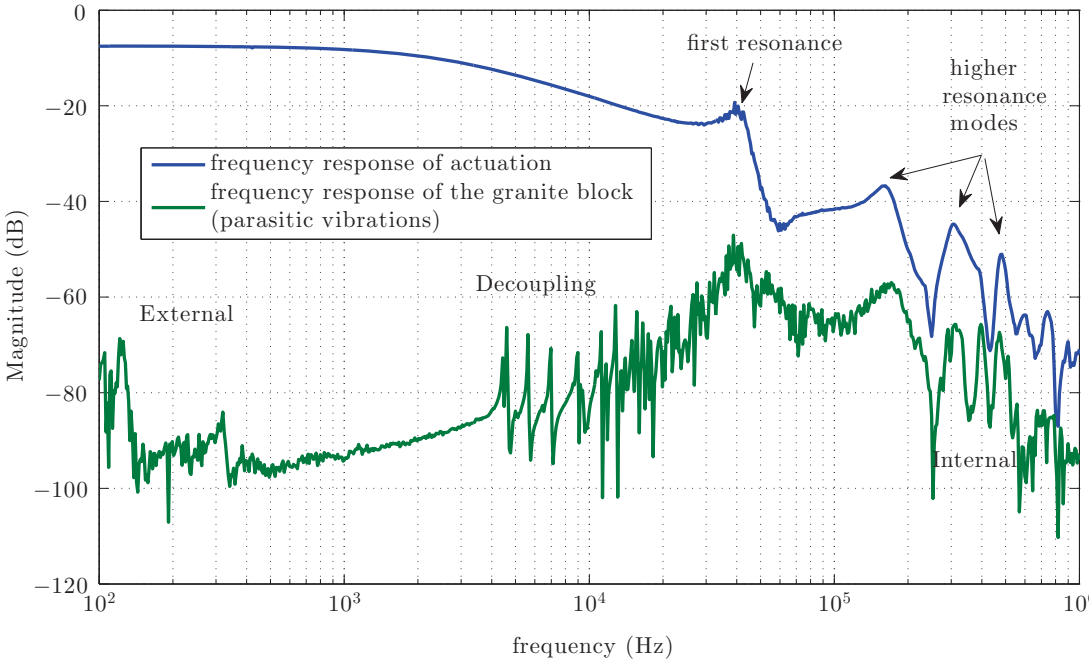


Figure 2.6: Frequency response of the actuation system (including amplifier and piezoelectric actuator) as well as the vibrations of the granite block.

Despite all of the above mentioned imperfections the granite block is used as balance for mounting piezo actuators on it. Usually, in AFMs there is far less counter-mass to damp excitation forces initiated by actuators, although displacement is referred to a relative coordinate system which might resonate relative to the base (inertial system).

2.1.3 Non-linearities

Piezoelectric materials show creep after changing the applied voltage, i.e. during the period when the voltage is already constant. It is noticeable in a slow drift of the elongation. It is in order of 1 to 2 % each time decade of elongation within 100 ms (logarithmic settle), so after a few hours the displacement can be as large as 10 % of the total displacement [26]. Its origin is supposed to be molecular friction [17]. For characterizing sensors by using piezoelectric actuators, care has to be taken to exclude frequencies below several Hertz to exclude the effect of creep.

It can be minimized further by controlling the charge of the piezo during actuation (charge controlled amplifier), however as long as a reference measurement can be done using a vibrometer, sensor characterization is also possible using voltage amplifiers.

However, it should be noted that besides the drift of the actuator the vibrometer is also afflicted with drift because of its operation principle. Basically, a vibrometer uses the optical Doppler effect for velocity measurement, however the position is not determined by integration but by fringe counting. To my knowledge there are uncertainties in this method and also thermal issues causing a drift.

It can be observed, that non-linearities (quantified by the THD) can be lowered further by exciting the piezo less than 20 % of nominal voltage, de facto $\approx 30\,\mathrm{V}_{PP} \approx 10\,\mathrm{V}_{RMS}$. This restriction reduces actuation by 80 % and unfortunately Signal to Noise Ratio (SNR) drops by 16 dB. So it becomes more difficult to acquire sensor dynamics particularly at high frequencies, where additional attenuation from piezo actuator also happens (see Figure 2.2).

2.2 Data Acquisition

For converting analogue signals of different sensors to a digital data an ADC type AD7985[6] has been chosen. This successive approximation ADC with 16-bit resolution capable to sample up to 2.5 Mega Samples Per Second (MSPS) has a conversion time of $t_{conv} = 320\,\mathrm{ns}$ only.
In order to implement data fusion (filters, offset correction, sensor multiplexing, etc.) a 32-bit ARM Cortex-M4 processor based microcontroller the STM32F407[7] with a Floating Processor Unit (FPU) with 210 Mega Instructions per Second (MIPS) @ 168 MHz is used. The interconnection of elements of the data acquisition system is shown in Figure 2.7.

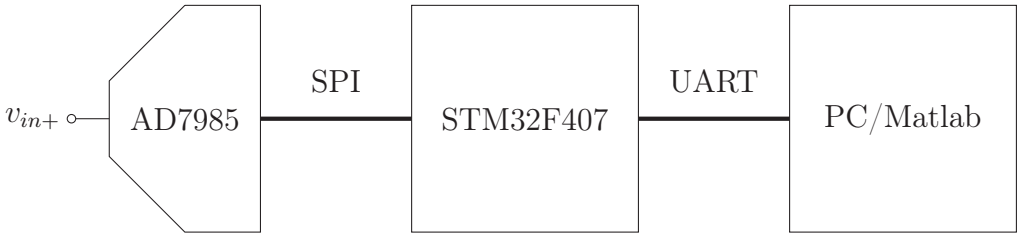


Figure 2.7: Data acquisition system consisting of an ADC (AD7895) which digitizes the analogue signals, a real time processing system (STM32F407) which executes filtering algorithms and a PC with data evaluation in MATLAB.

2.2.1 Analog to Digital Converter

The ADC and the Processing System (PS) (STM32F407 microcontroller) are interconnected by the Serial Peripheral Interface (SPI) bus. Only the ADC (bus slave) transfers data to the PS (bus master) after initialization has been done. The master reads 16 bits of conversion result with a delay of $t_{d1} = 182\,\mathrm{ns}$ within $t_{tr} = 381\,\mathrm{ns}$ at the highest possible clock frequency of the PS at $f_{clk} = 45\,\mathrm{MHz} = 16/t_{tr}$ according to Figure 2.8.

[6] Analog Devices Inc., Norwood, Massachusetts, USA
[7] STMicroelectronics N.V., Genf, Switzerland

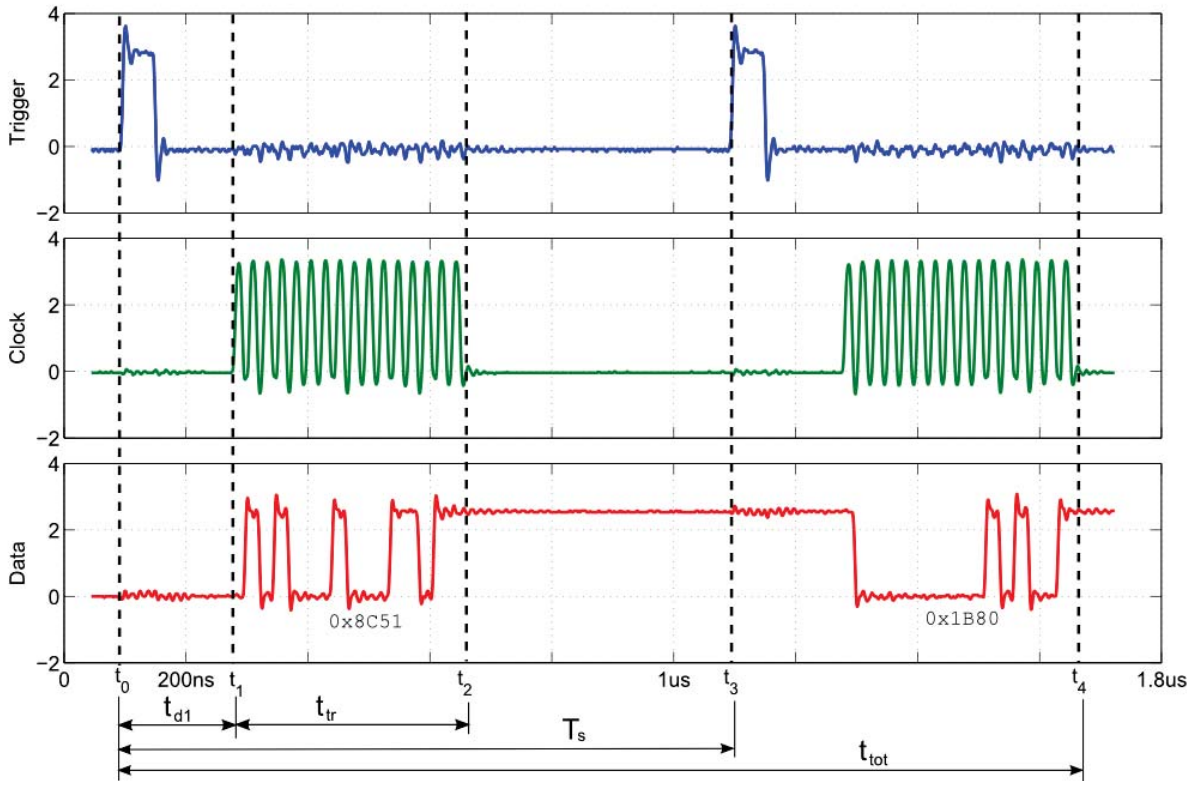


Figure 2.8: Timing diagram of the ADC on the SPI bus

Due to the fact, that the AD7985 transmits the previous conversion result at actual trigger, the total delay makes $t_{tot} = 1.56\,\mu$s at $f_s = 1/T_s = 1\,$MHz $= 1\,$MSPS sampling rate with the presented architecture.

The theoretical closed loop bandwidth can be as high as

$$f_{cl} = \frac{1}{10}\frac{1}{t_{tot}}\bigg|_{\text{AD7985}} = 64\,\text{kHz} \tag{2.4}$$

by assuming 10 samples per cycle of a sine for proper interpolation as adequate (5 times more than the Nyquist sampling).

It is verified on the present PS, that the task to compute the Kalman filter for a third order model of actuation makes a closed loop bandwidth of only $f_{cl} = 50\,$kHz possible for the assumption (2.4). To compute a Multiple Input Single Output (MISO) Kalman filter of fifth order model (as presented in Section 3.3.3 with two sensors) f_{cl} drops to 10 kHz for the same assumption (2.4).

According to the data sheet of the ADC [4] the theoretical lowest delay $t_{tot} = 500\,$ns could be achieved at $f_{clk} = 84.2\,$MHz implying an $f_{cl} = 200\,$kHz. The STM32F407 PS with $\max(f_{clk}) = 45\,$MHz clock of the SPI bus and present implementation of software (using the standard peripheral library of STM in the program language C) reduces f_{cl}

by a factor of 4 at least[8] compared to an implementation on a Field Programmable Gate Array (FPGA) where mostly only the delay of ADC and data transmission counts[9].

By using STM32 there is almost unlimited flexibility in shaping and adapting filters and tuning the fusion algorithm, even during operation. The wide variety of standard peripheral interfaces included in STM32 and its simplicity in programming compared to e.g. a Xilinx FPGA and the description language VHDL allows a comparably short development time. Although FPGAs are more suitable for real time filtering and control tasks and they are used, to my knowledge, in high-end applications or in standard products sold in large quantities where the development time is justified, but even in individual applications where excessive computation is needed, such as by implementing a moving horizon observer as in [29], the effectiveness of filtering can be shown using the microprocessor STM32F407 capable up to 210 MIPS.

Triggering

The triggering of a conversion can be initiated by

(a) the PS internally via a timer, at sampling rates in range $[0, 1]$MHz

(b) externally at any frequency in range $[0, 0.8]$MHz

Option (b) is useful for supernyquist sampling (undersampling), which is used to mix down the frequency of the measurement to the base-band when using an AC-measurement bridge. This is equivalent to an envelope detector. By using a second ADC triggered at 90° phase shift to the first ADC, an IQ-demodulator can be built, providing in-phase and quadrature components, as if a lock-in amplifier would be used.

Universal Asynchronous Receiver/Transmitter

As indicated in Figure 2.7 the data transfer between PS and the personal computer (PC) or notebook with MATLAB installed on it, is not time crucial anymore. It is realized with asynchronous bit transfer at 115200 bits/s over a TTL to a virtual-COM adapter. Data is recorded in MATLAB without an additional software interface, based on the standard java input/output methods and functions on which MATLAB is based on. The reading from a standard serial input can be done more effectively by addressing the appropriate ports directly by an executable, e.g. compiled from a C-code on the used operating system.

2.2.2 Signal conditioning

In order to use the full input range of the ADC, in the present case $0\,\mathrm{V} \leq v_{in+} \leq 4.095\,\mathrm{V}$ with sensors of different range of output voltages (mostly symmetrical, non-differential)

[8] The amount of performance degradation depends on the additional control algorithm.

[9] By assuming no comparable delay in saving/buffering of the data in connection with ADCs sampling at several MHz. However if the ADCs are sampling at several hundreds of MHz or even at several GHz probably conversion delays (serial to parallel) dominates.

and to avoid a voltage drop due to capacitive input currents into the ADC a high-frequency ultra low-offset operational amplifier AD8021[10] is used in a non-inverting summing stage configuration as it can be seen in Figure 2.9 with parameters in Table 2.2.

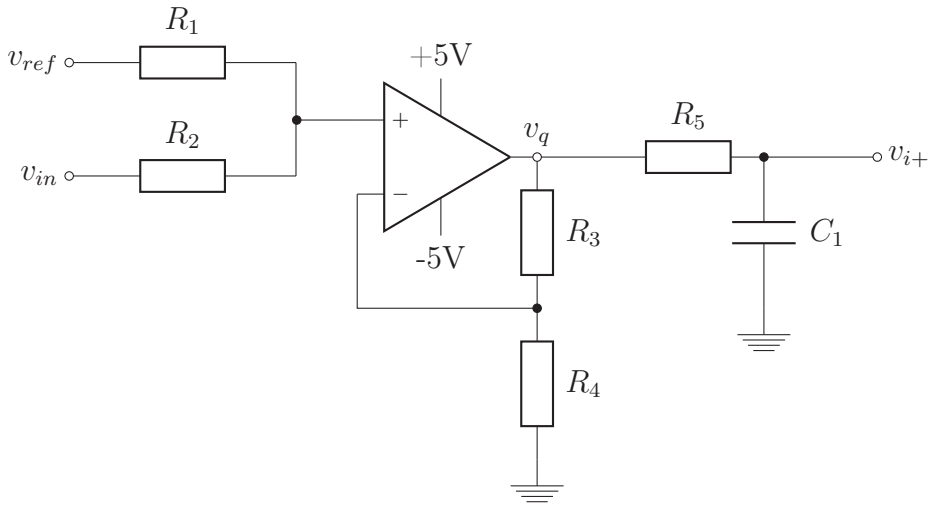


Figure 2.9: Signal conditioning circuit

R_1	R_2	R_3, R_4	v_{ref}	v_{in}	R_5	C_1
20 kΩ	5 kΩ	10 kΩ	4.096 V	$[-1, 1]V$	18 Ω	3 nF

Table 2.2: Parameters of analogue circuit for data acquisition

Due to the non-ideal behavior of the summing circuit a small peaking (up to 10 dB) is always present as it can be seen around 4.5 MHz in Figure 2.10. To provide equal gain and as few phase lag as possible up to 1 MHz, R_5 and C_1 are chosen to from a first order low-pass filter with a cut-off frequency of $f_{-3\,\mathrm{dB}} = 5$ MHz. Only in the case when the Nyquist theorem can not be guaranteed an additional anti-aliasing filter is connected with $f_c < 500$ kHz.

2.2.3 Performance of conversion

A histogram of ADC values captured over more than 6 seconds confirms a non markable drift of the signal conditioning circuit during that time, as can be seen in Figure 2.11. Moreover, the assumption over a normally distributed noise of the Data Acquisition (DAQ) including the signal conditioning circuit is affirmed, since the quantization noise of the ADC would be uniformly distributed. The $\pm 3\sigma$-resolution of the DAQ is 2.31 mV for 20×10^3 samples and a sampling rate of 3 kSPS.

When applying a sine with frequency of 10 kHz from a high performance signal generator[11] to v_{in} the non-linearity of ADC with the analogue circuit (Figure 2.9) can

[10] Analog Devices Inc., Norwood, Massachusetts, USA
[11] SDG5162, Siglent Technologies Europe GmbH, Hamburg, Germany

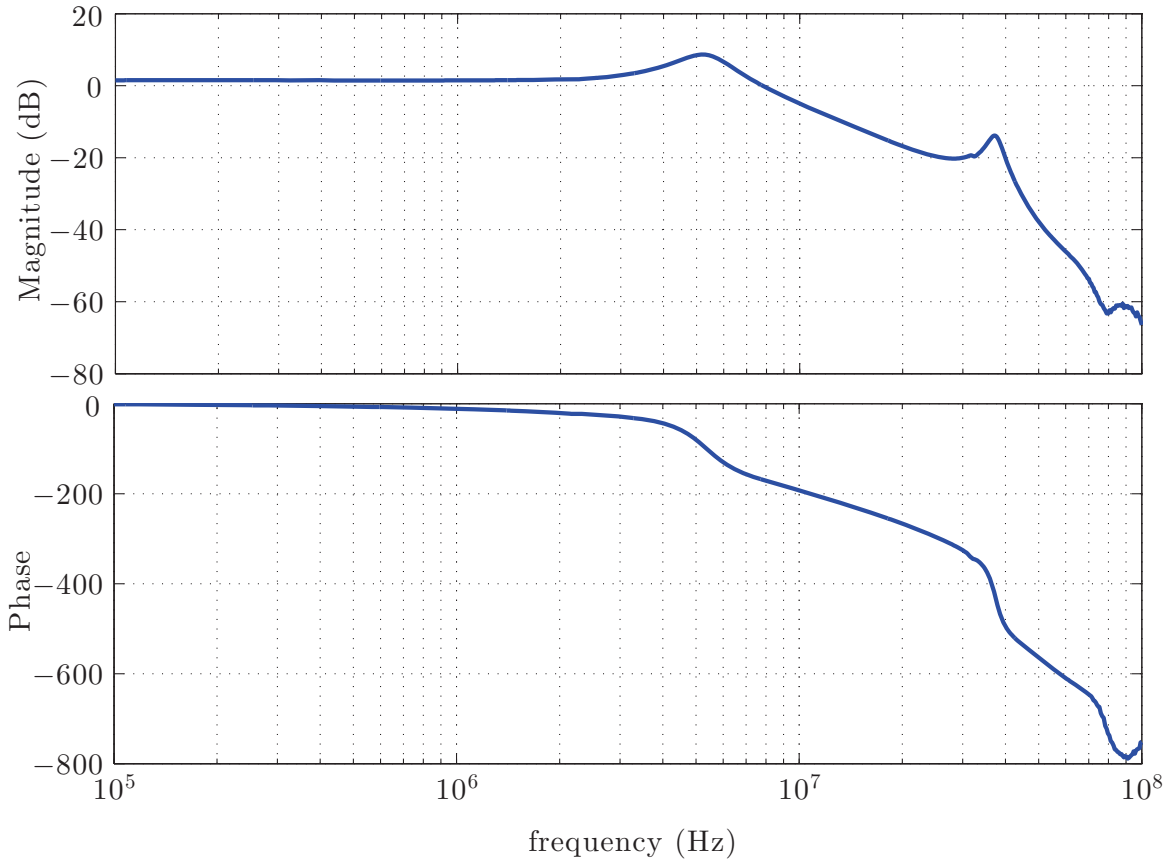


Figure 2.10: Transfer of the analogue signal conditioning circuit. $\mathcal{L}(v_{i+}(t))/\mathcal{L}(v_{in}(t))$. Parameters as in Table 2.2.

be measured and compared to that in the data sheet. The THD (as shown in Figure 2.12) calculated out of the fundamental and the first 5 harmonics shows a value of −66 dB which verifies that acquisition system performs as desired. THD over frequency is reported in [4].

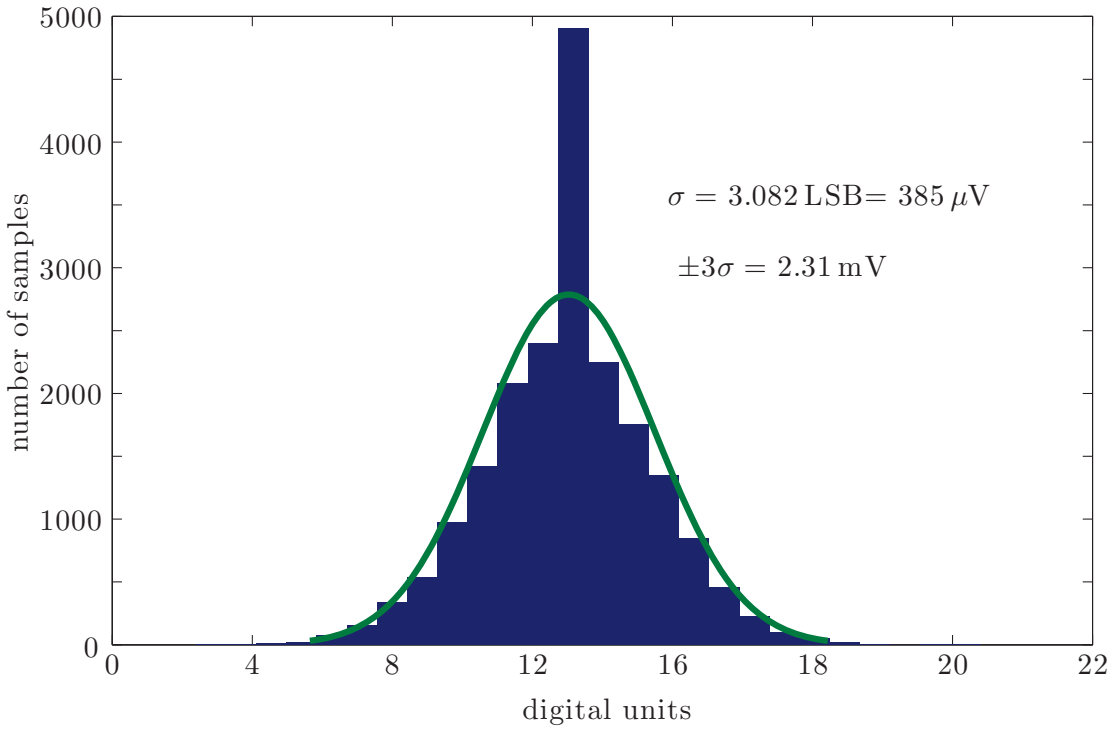


Figure 2.11: Histogram of the quantized voltage at $v_{in} = 0\,\mathrm{V}$ computed out of 2×10^4 samples sampled at 3 kHz, other parameters as in Table 2.2.

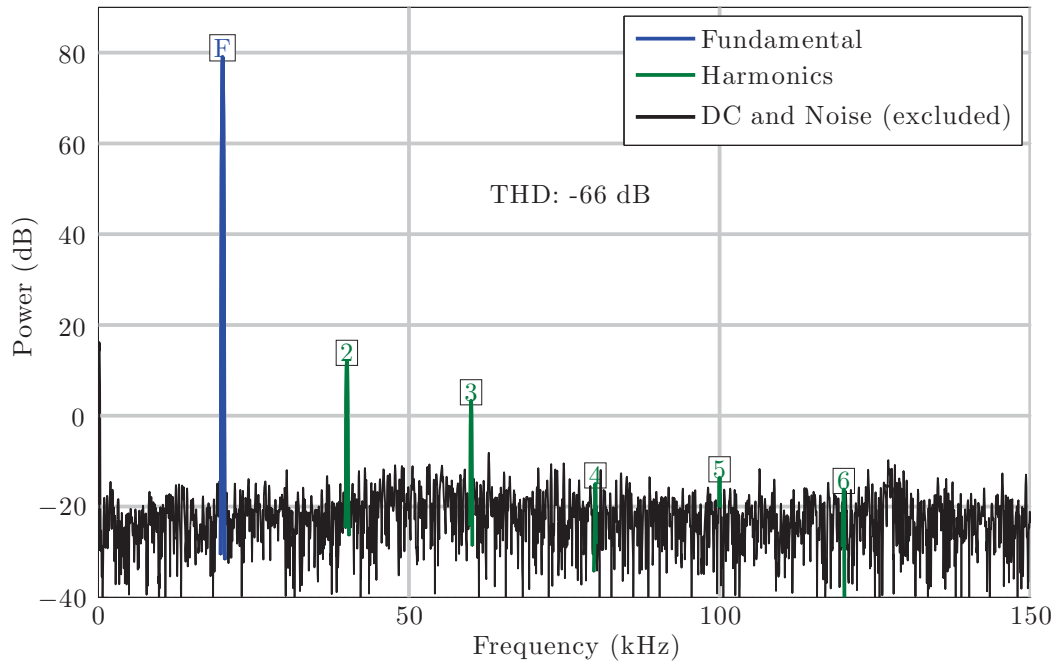


Figure 2.12: Total harmonic distortion analysis: frequency content of a 10 kHz 2 Vpp sine sampled at 338.5kSPS.

2.3 Strain gauges

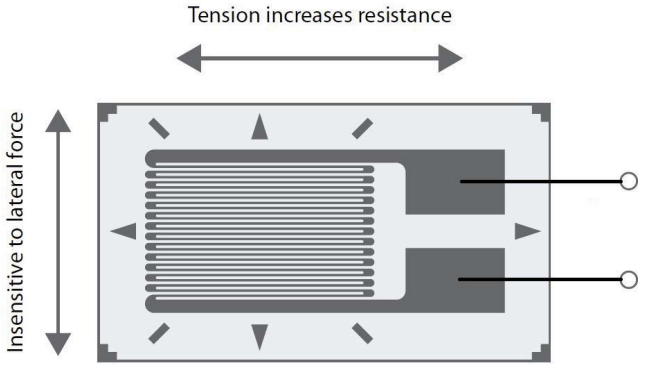


Figure 2.13: Photo of strain gauge foil (taken from [16])

2.3.1 Operation principle

Strain gauges consist of a thin isolating foil with meander-shaped resistive layer on it as illustrated in Figure 2.13. By stretching them along the sensitive direction they increase resistance in order of 0.1 %.

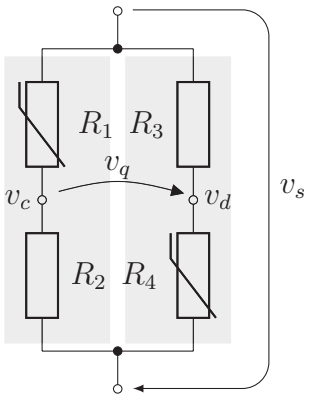


Figure 2.14: Strain gauges in Wheatstone bridge with active elements R_1 and R_4, pairwise placed sidewards on the stacked piezo actuator, with supply voltage v_s and bridge voltage v_q.

Figure 2.14 and 2.15 shows the configuration which can be used when strain gauges are applied to piezoelectric actuators. A Wheatstone bridge with two active elements R_1, R_4 and two fix valued R_2, R_3 is used[12]. An equal change in resistance of the active elements is mapped in the bridge voltage v_q. The passive elements (fix valued) are not sensible to strain (since they are perpendicular to the direction of actuation) however they have the same temperature coefficient as active elements to ensure no change

[12] From now on, the term *strain gauge* or *strain gauge sensor* refers to the Wheatstone bridge consisting of four strain gauge elements.

in v_q[13] in response to a change of temperature uniformly affecting all four elements. Unfortunately, the temperature compensation is only given for a balanced state of the bridge i.e. if $v_q = 0$.

2.3.2 Thermal properties

Sometimes strain gauges are applied on large metal parts (*carriers*) to measure strain or deformation as in a load cell [16]. Since they are in close contact to the carrier, the same temperature of both is presumed. Sometimes, the resistive film in a strain gauge is made of an alloy with negative temperature coefficient to compensate for resistance change caused by the thermal expansion of the carrier to which it is applied (self-compensated strain gauge) [16]. Thermal expansion coefficients of steel $\alpha_{st} \approx 10^{-5}K^{-1}$ and of aluminium $\alpha_{al} \approx 2 \times 10^{-5}K^{-1}$ are approximately equivalent. Therefore, possibly a quarter or a half bridge (consisting of one or two grids) is adequate for lots of applications when using carriers out of steel or aluminium. The temperature change does not significantly influence the bridge voltage v_q.

However, when applying strain gauges on ceramic materials (as piezoelectric actuators consist of) no temperature compensation is ensured, since the expansion coefficient of quartz $\alpha_{qu} = 10^{-4}K^{-1}$ is much larger than that of steel or aluminium and additionally the heating effect of the piezo actuator (due to inner losses) causes a a rise in temperature even until 40 °C (measurement) above the ambient temperature.

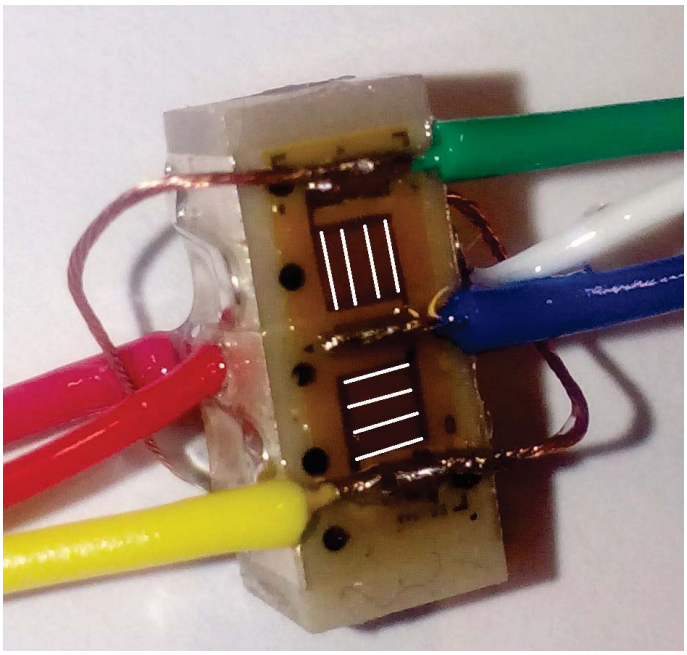


Figure 2.15: Strain gauges applied on a piezoelectric actuator. Size of foil: (3×8) mm and of actuator: $(5 \times 5 \times 12)$ mm

[13] The bridge voltage v_q is referred as sensor's output.

type (manufacturer)	`FAET-A6347Q` (Micro-Measurements)
grid resistance	$R_i = 350\,\Omega$
supply voltage	$v_s = 20\,\mathrm{V}$
instrumentation amplifier	`INA111` (Burr-Brown)
amplification	$G = 56\,\mathrm{dB}$ ($R = 73\,\Omega$)
sensitivity	$S = 238 \times 10^3\ \mathrm{V/m}$

Table 2.3: Parameters of strain gauges and amplifier

2.3.3 Dynamic properties

In order to acquire the frequency response of the strain gauge sensor, the response of the system with actuation and sensor is compared to a reference measurement with the vibrometer (made immediately after the first) and the response of the actuation is eliminated. Only the difference as well in magnitude as in phase is drawn in Figure 2.16 called the sensor's frequency response. It is, like the sensor's response would be corrected for constant amplitude of excitation. Therefore, $f_{-3\,\mathrm{dB}} \approx 20\,\mathrm{kHz}$ and it *slightly* depends on the amplification factor set for the instrumentation amplifier. The reason for this is still uninvestigated.

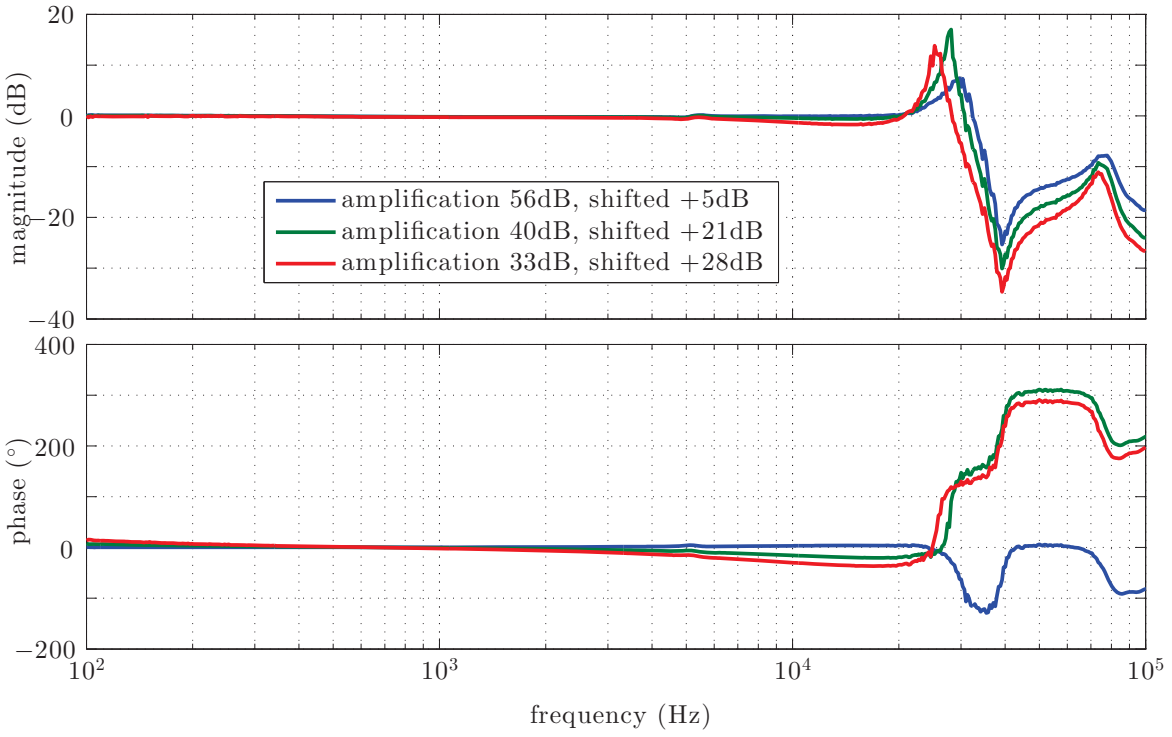


Figure 2.16: Frequency response of the strain gauge sensor at parameters as in Table 2.3.

2.3.4 Resolution

To determine resolution of strain gauges the PSD of noise is acquired while the piezo actuator is not actuated. The creep of the PZT material (as discussed in Section 2.1.3) does not effect the measurement, because the lower frequency limit of the measurement is set to 10 Hz. The PSD in (logarithmic units) is captured over a linearly spaced frequency axis (it is a limitation of the used network analyzer HP4935) while setting up the required resolution bandwidth (RBW) and video bandwidth (VBW) for each decade of frequency according to Table 2.4 and merging measurements afterwards, as can be seen on Figure 2.17.

To determine resolution, according to Section 1.4.4, square root of noise power density is summed up to obtain cumulative noise voltage $v(\mathscr{F})$ for the frequency range $\mathscr{F} = [f_l, f_h]$, according to (1.15). For $f_l = 10\,\text{Hz}$ and $f_h = [10\,\text{Hz}, 1\,\text{MHz}]$ the cumulative noise voltage $v(\mathscr{F})$ is shown in Figure 2.18 and the sensor resolution (since it is proportional to $v(\mathscr{F})$ by the sensitivity factor) in Figure 2.19. According to that, the $\pm 3\sigma$-resolution for e.g. a 20 kHz bandwidth limited measurement is $r(20\,\text{kHz}) = 22\,\text{nm}$.

f_1	f_2	RBW	VBW
10 Hz	1 kHz	3 Hz	10 mHz
1 kHz	10 kHz	10 Hz	30 mHz
10 kHz	100 kHz	30 Hz	100 mHz
100 kHz	1 MHz	300 Hz	1 Hz

Table 2.4: Resolution bandwidth (RBW) and video bandwidth (VBW) as used for noise spectrum measurement. Obtained empirically.

2.3.5 Non-linearities

Schitter et al. in [31] compares strain gauges and optical sensors. It is shown, that a strain gauge can bring non-minimum-phase behavior into the piezo actuator system. It is observable as

> *(...) the response to a guidance step[14] causes the system to react initially in the opposite direction before it settles to the specified steady state value* [31].

A step signal (steep transition) is a special case of a rectangle signal, which is defined by the decreasing odd Fourier coefficients (1, ⅓, ⅕, ⅐ ...) as expressed by

$$\text{rect}(t) = \sum_{m=0}^{\infty} \frac{1}{2m+1} \sin(2\pi f(2m+1)t) \tag{2.5}$$

containing an infinite number of frequency components. Due to the low bandwidth of the test setup $f_{-3dB} = 2\,\text{kHz}$ (Figure 2.6) a step input is filtered as much, that non-minimum-phase behavior could not be verified.

[14] *Natura non facit saltus.*

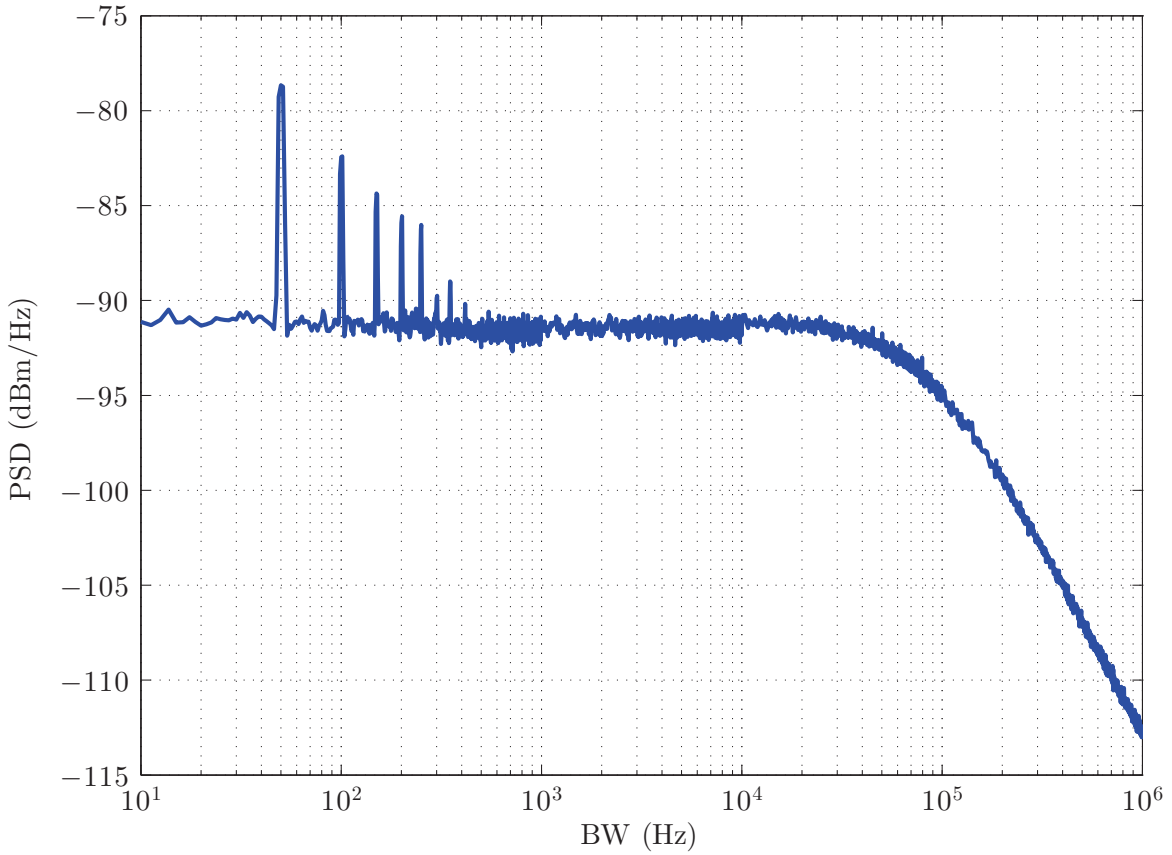


Figure 2.17: PSD of sensor noise for strain gauges on logarithmic frequency scale concatenated out of 5 measurements with linearly placed data points.

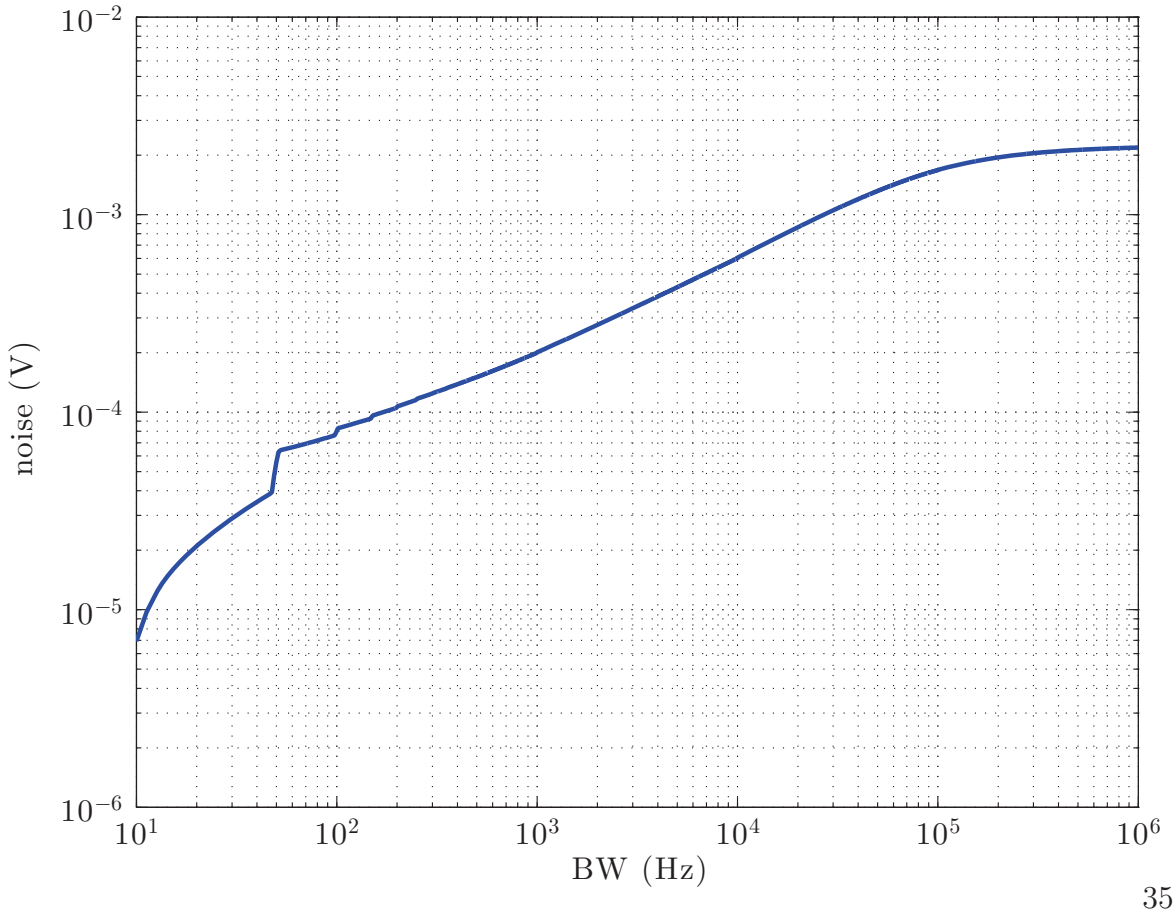


Figure 2.18: Cumulative noise voltage of strain gauges.

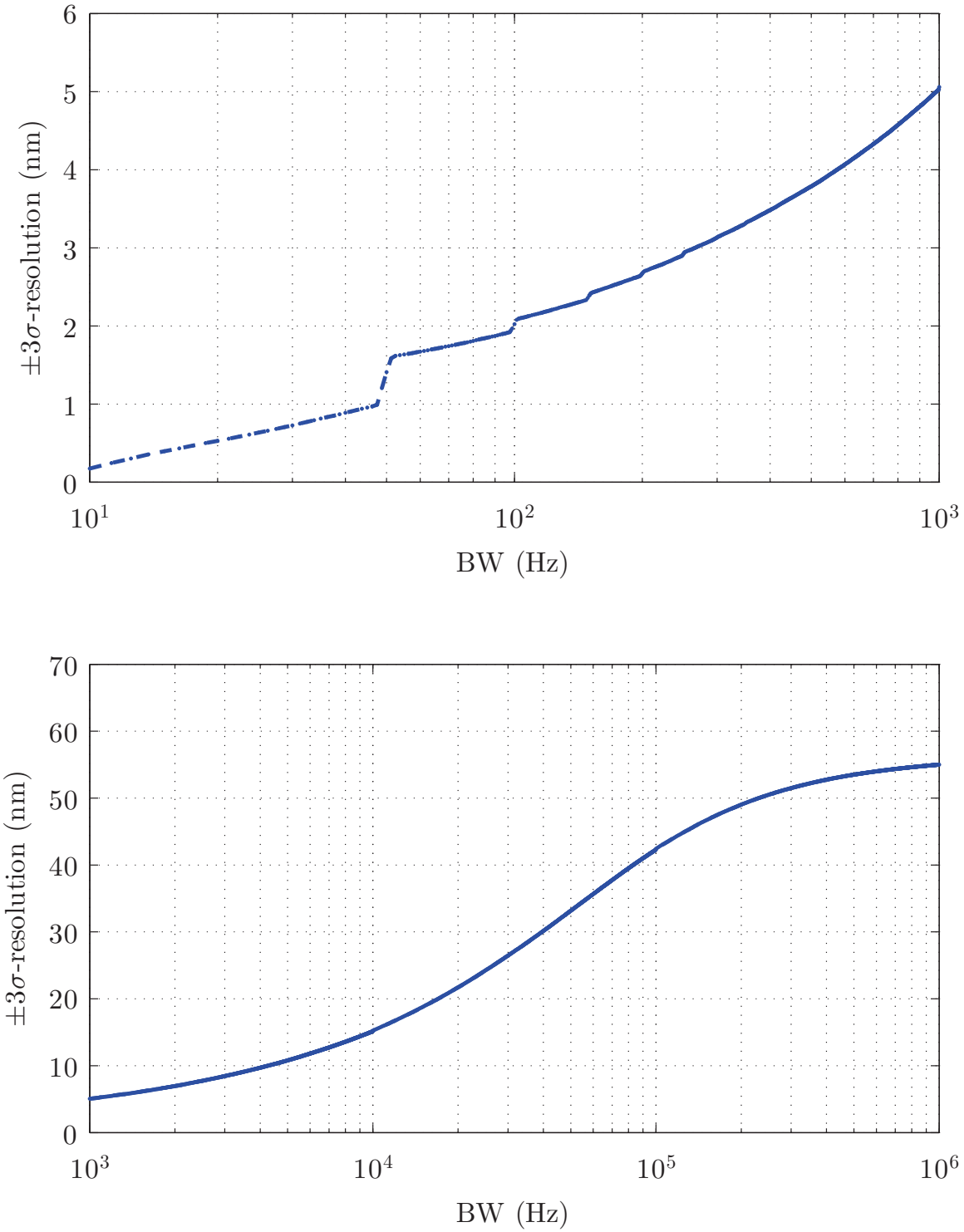


Figure 2.19: $\pm 3\sigma$-resolution of strain gauges with instrumentation amplifier at parameters of Table 2.3 versus a partitioned frequency axis.

The creep of the glue used to install strain gauges reduces accuracy. The effect can be minimized using special designed glues for this purpose. Some of them have to pause several hours [16] at relatively high temperatures (100-300°C) after applying them, which is not always allowed in case of piezoelectric actuators due to depolarization effects. Possibly mainly this effect is observable in the measurement in Figure 3.9.

2.4 Optical sensors

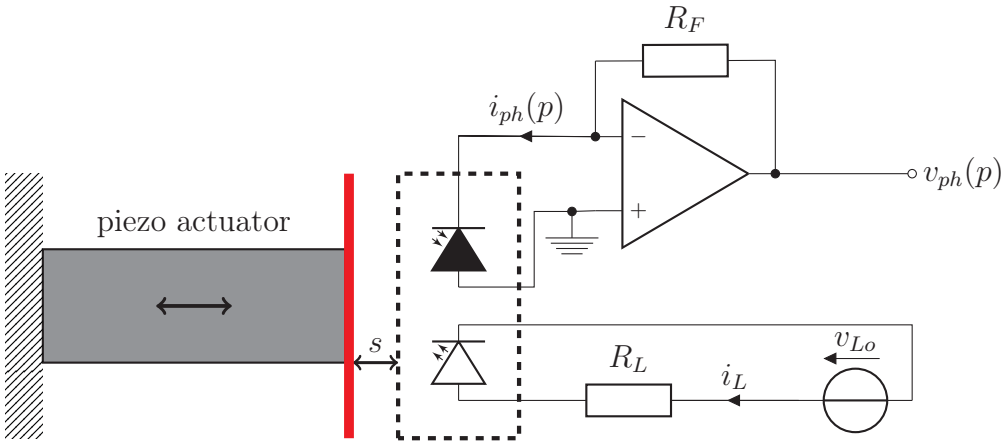


Figure 2.20: Characterizing optical proximity sensor HDSL9100 with photodiode (operating principle)

2.4.1 Operation principle

A unique aspect of the usage of optical proximity sensors is that measuring light intensity is usually not the primary intent. The optical proximity sensor consists of a Light Emitting Diode (LED) and a photodiode aligned to the same direction as indicated in Figure 2.20 in the dotted rectangle. The emitted light gets reflected from the reflective surface carried by the actuator (red indicated plate) and enters the photodiode letting flow the photocurrent i_{ph}. The varying gap s between the actuator and the sensor modulates i_{ph}.

2.4.2 Photodiode

In a wide variety of applications photodiodes[15] are used for monitoring light [42]. Due to a linear transmission between photons entering the pn-junction of a semiconductor and electrons emitted, a current to voltage amplifier, so called Trans-Impedance Amplifier (TIA) is preferred to amplify the small current $i_{ph}(p)$ to a voltage that can be handled $v_{ph}(p)$[16] as it can be seen in Figure 2.20.
Depending of the area of the photodiode, quantum efficiency and other parameters, the photocurrent is very small, $i_{ph}(p_1) \approx 50\,\text{nA}$ at $p_1 = 100\,\text{lux}$ illumination (poor office light) for the optical proximity sensor HDSL9100 (Agilent Technologies, Santa Clara, CA, USA). In order to transform it to a practical range of voltage[17] it needs to be amplified by the transimpedance $\approx 10^7\,\Omega = 140\,\text{dB}\Omega$. Therefore, special considerations regarding a low noise amplification need to be adhered when designing analog circuit and placing components.

[15] Based on the inner photoelectric effect, discovered by Albert Einstein [38].
[16] Parameter p announces a dependence of photon flow, i.e. intensity of light.
[17] e.g. $[0, 5\,]V$

	par1	par2
supply voltage	$\pm 15\,\mathrm{V}$	$\pm 15\,\mathrm{V}$
transimpedance $20\log(R_F)$	$120\,\mathrm{dB}\Omega$	$140\,\mathrm{dB}\Omega$
sensitivity S_{max}	$8 \times 10^3\,\mathrm{V/m}$	$54 \times 10^3\,\mathrm{V/m}$
total range	$2.3\,\mathrm{mm}$	$470\,\mu\mathrm{m}$

Table 2.5: Parameters used for characterizing photodiode based proximity sensor

If the proximity sensor (as it can be seen in Figure 2.20 indicated with dotted rectangle) containing the photodiode (indicated with the black diode symbol) is perfectly in contact with the reflective surface carried by the piezo actuator (indicated by the red plate) covering the photodiode, only the dark current flows $i_{ph}(0) = i_{dark}(\vartheta)$.
The major effect of operation in the *near range* is the scattering of light, therefore, the reflectivity of the surface is not as important as in the *far range* as reported in [33] and proved by measurements. With increasing distance s the illumination p on the photodiode increases, under certain circumstances saturating it. The phenomenon of saturation can be seen in Figure 2.22, the static response in Figure 2.21.

2.4.3 Optimization

In order to characterize the sensors over a wide range of displacement s, a long stroke actuator (PID controlled positioner with 2 mm range and closed loop bandwidth $f_{cl} < 1\,\mathrm{Hz}$ according to definition (1.8)) and a short stroke actuator (high dynamic stacked piezo actuator with known dynamics ($\approx 10\,\mu\mathrm{m}$ displacement at 1 Hz, $f_{-60\,\mathrm{dB}} > 200\,\mathrm{kHz}$) are used. As it is known from [33] and is verified by measurements, there is a non-linear (and not unique) relation between displacement s and illuminance p at the photodiode, effecting the photocurrent $i_{ph}(p)$. The aim of this characterization is to find operating point s_{op} at which actuated by $\pm 10\,\mu\mathrm{m}$ at maximum (limits of the piezoelectric actuator), the gradient of the photocurrent is the highest in order to have a large change in measurement on response to a displacement.

$$s_{op} = \max_s \frac{d}{ds}\, i_{ph}\big(p(s)\big) \tag{2.6}$$

As it can be observed out of measurements, the optimum also depends on the transimpedance R_F (denoted by $s_{op} = g(R_F)$ with $g(\cdot)$ as some unknown, probably non-linear function). Instead of finding global maxima in dependence of two variables, some fixed values for R_F are set and $s_{op}(R_F)$ is determined empirically. Around this operating point $s_{op}(R_F)$, the piezo actuator moves only several μm (additionally decreasing by $20\,\mathrm{dB/decade}$ of frequency increase as presented in Section 2.1.2) causing a very low relative change in the intensity of received light Δp modulating $i_{ph}(p_1 + \Delta p)$, respectively.

2.4.4 Interference

It became apparent that the stacked piezo actuator strongly influences the recorded dynamics of the photodiode. A fully covered photodiode ($p = 0 \implies i_{ph} = i_{dark}$) is

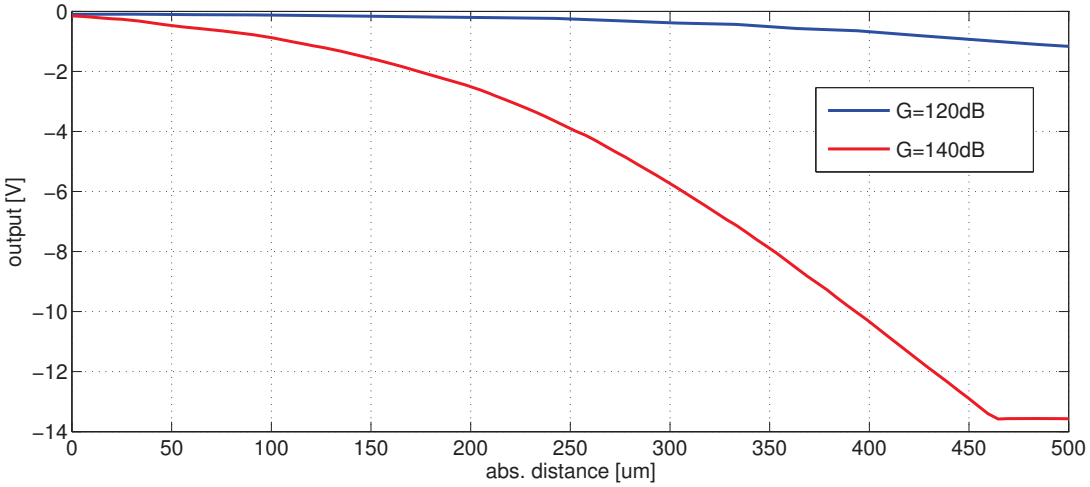


Figure 2.21: Stationary response to displacement of optical proximity sensor HDSL9100 with TIA at different transimpedances

approached to the piezo actuator during oscillation (sweeping) and transfer function $G_s(s) = \mathcal{L}(v_{ph}(t))/\mathcal{L}(s(t))$ is captured at different distances s_i as can be seen in Figure 2.23 for $s_i = \{1, 5, 10\}$ mm. The assumed real movement of the actuator (measured with vibrometer) is plotted with blue (labelled with REF). It has a slope of 20 dB/decade above 3 kHz and has the first resonance at 41 kHz as introduced in Section 2.1.3 in Figure 2.2. An ideal sensor would deliver the same, except from a frequency independent gain (due to non-equal sensitivity of sensors in general).

The transfer function of the TIA ($G_{TIA}(s) = \mathcal{L}(v_{ph}(t))/\mathcal{L}(i_{ph}(t))$ shows less than 3 dB of variability[18] in magnitude in the frequency range $\mathscr{F} = [500\,\text{Hz}, 1\,\text{MHz}]$, i.e.

- $\Delta G_{TIA}(\mathscr{F}) < 3\,\text{dB}$[19],
- however the fully covered sensor has $\Delta G_s(\mathscr{F}) \geq 30\,\text{dB}$[20]

it can be concluded, that there is an interference between actuator and sensor which is not of optical transmission. Possibly capacitive and inductive coupling between actuator and sensor are present. This explains a higher sensitivity of $G_s(s)$ closer to the surface of the actuator (at lower values of s) as visible in Figure 2.23. Furthermore, the sensor's transfer function $G_s(s)$ pretends to have a high pass property at all measured distances. Because of this and that of high variability it can be concluded that the real movement of the actuator (measured with the vibrometer) is not acquired sufficiently by the sensor $G_s(s)$. In the case when the photodiode is covered ($p = 0 \implies i_{ph} = i_{dark}$) the transfer function $G_s(s)$ should be 0 over all frequencies.

Therefore, different techniques are empirically validated to minimize undesired influence of actuator. The sensor case and surroundings are shielded and grounded in order to minimize capacitive coupling as well as magnetic shields (steel plates) are

[18] According to definition of variability of a transfer function by (1.24) in Section 1.7.1
[19] Measured with approximately constant amplitude of input current of $\approx 5\,\mu$A over the frequency range $\mathscr{F}$.
[20] Acquired at parameters `par2` from Table 2.5.

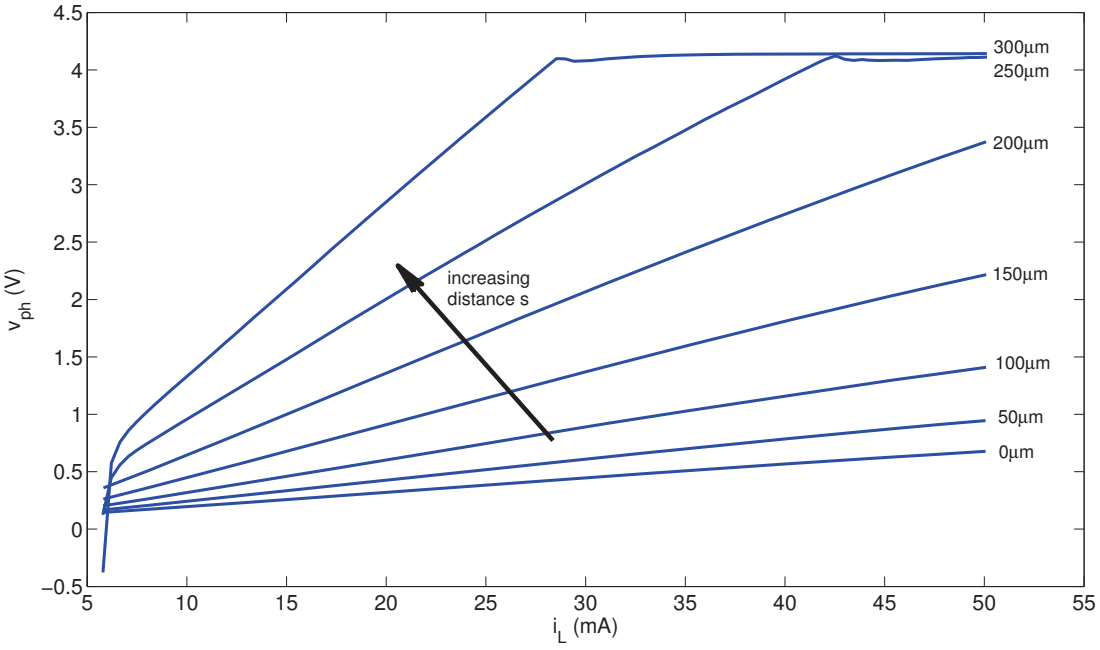


Figure 2.22: Effect of saturation (TIA and photodiode) noticeable on the knee of $v_{ph}(s, i_L)$ at various distances s and LED currents i_L of optical proximity sensor HDSL9100. The non-linear relation, noticeable on the uneven density of $v_{ph}(s, i_L)$ in response to isodistante spacing $\Delta s = 50\,\mu$m can be seen.

placed between the actuator and the sensor with a small hole through it enabling the measurement. Modified orientation of the sensor is investigated to lower the unwanted coupling. All this together does not significantly help to reduce interference.
However, the prolongation of the piezo actuator seems promising. Lightweight spacers are mounted on top of the actuator in order to keep the sensible proximity sensor farther away from the actuator. As it can be already seen in Figure 2.23, $\|G_s(s)\|$ is much lower for an increased distance s. However, the spacers have a small stiffness and a not negligible weight, so they are introducing additional resonances next to that of piezo itself. For the sensor fusion it is necessary that multiple sensors (consequently at different locations) measures the same amount of displacement as in amplitude as in phase, which might be violated when mounting an additional *mass-spring element*.

Other actuators, such as a Lorentz actuator (voice-coil actuator) with precautions can be possibly used to characterize a photodiode based proximity sensors. However because of large actuated mass in general, they are usually not able to move at such high frequencies (e.g. >100 kHz), therefore, characterization is done by optical modulation.

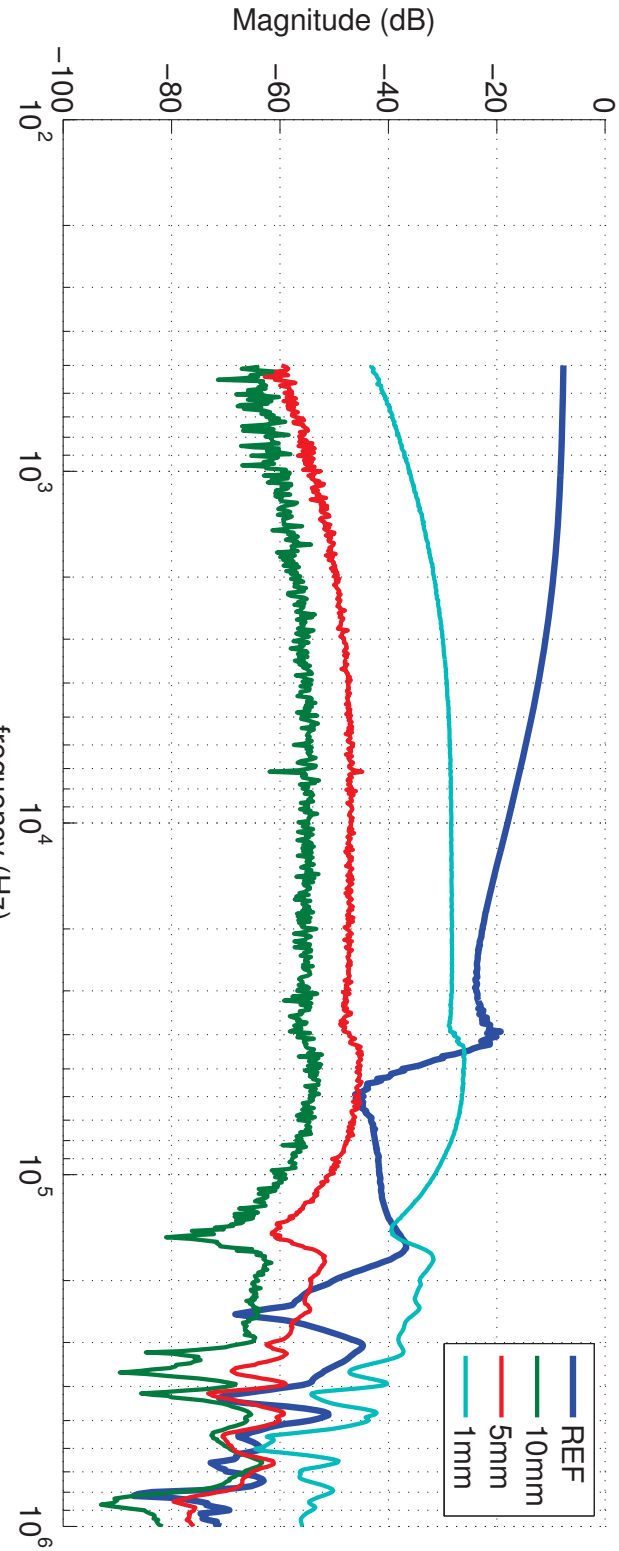


Figure 2.23: *Pretended* transfer function of optical proximity sensor HDSL9100

2.4.5 Optical modulation

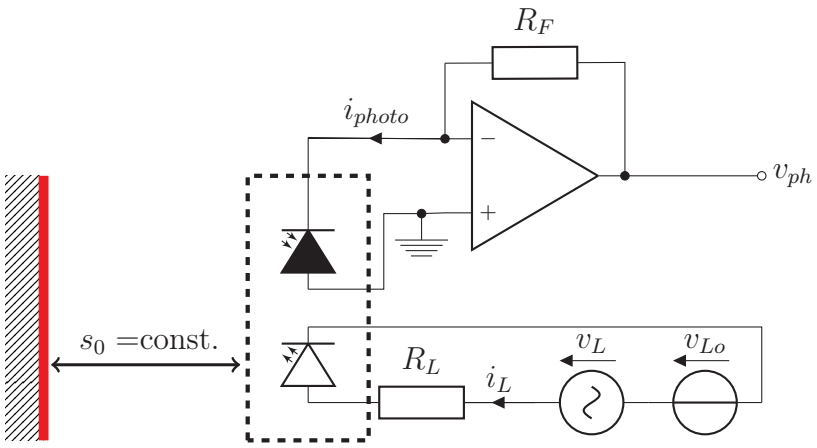


Figure 2.24: Characterizing photodiode by modulated illumination

Assuming a similar response caused by an actuated optical mirror by Δs than by modulated light by Δp[21] (*same effect assumption*) characterization of sensor dynamics is done by optical modulation. If the sensor is faced to a professional positioning stage (as e.g. usually in an AFM) the assumption about same effect might be given, in contrast if the sensors is faced to the top of a stacked piezoelectric actuator in tight closeness to it operating at high voltages, this assumption is not met. The transfer function of the sensor (acquired by modulation of light) $G_m(s) = \mathcal{L}(v_{ph}(t))/\mathcal{L}(i_L(t))$ is characterized, with LED current i_L. The LED forces photons to travel to the reflective surface at fixed distance s_0, than back into the photodiode letting i_{ph} flowing, which is amplified by the TIA as visible in Figure 2.24.

Practical considerations

It is not as simple to realize

$$i_L(t) = i_{Lo} + i_a \cos(\omega t) \tag{2.7}$$

due to non-linear characteristic of the LED. The exponential rise of the current through a diode when applying a linearly increasing voltage on it, motivates to use a resistor R_L in series to the LED to make the current i_L to rise more flat, to

$$R_L = \frac{v_L}{i_L} = 100\,\Omega \qquad \text{for } v_L > 1.1\,\text{V} \tag{2.8}$$

The offset voltage v_{Lo} is set to drive the current $i_{Lo} = 20\,\text{mA}$. For realization of (2.7) a bias-tee is used, which equivalent circuit can be seen dashed in Figure 2.25. It enables a low impedance path for direct current (DC) from v_{Lo}[22] to v_q since $i_{dc} = (v_{Lo} - v_q)/R_b$

[21] At least for *small* Δp.
[22] From the connector where the potential v_{Lo} is labelled.

as well as a low impedance path for high frequency (HF) currents from v_L to v_q since $i_{ac} = (v_{Lo} - v_q)/(i\omega C_b)$ but neither noteworthy DC nor HF currents flow across[23].

The sensor's transfer function $G_m(s)$ for various s_0 and empirically found $i_a = 1\,\mathrm{mA}$ can be seen in Figure 2.26.

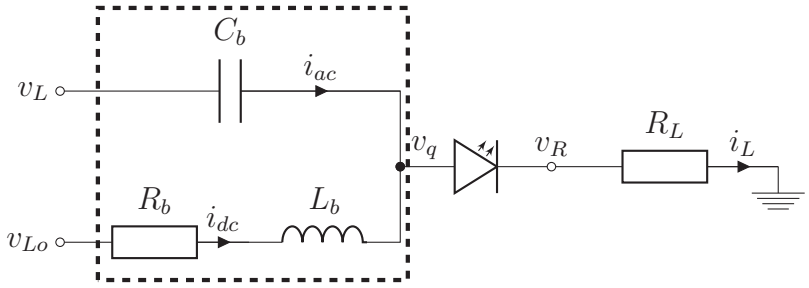


Figure 2.25: Equivalent circuit of bias-tee

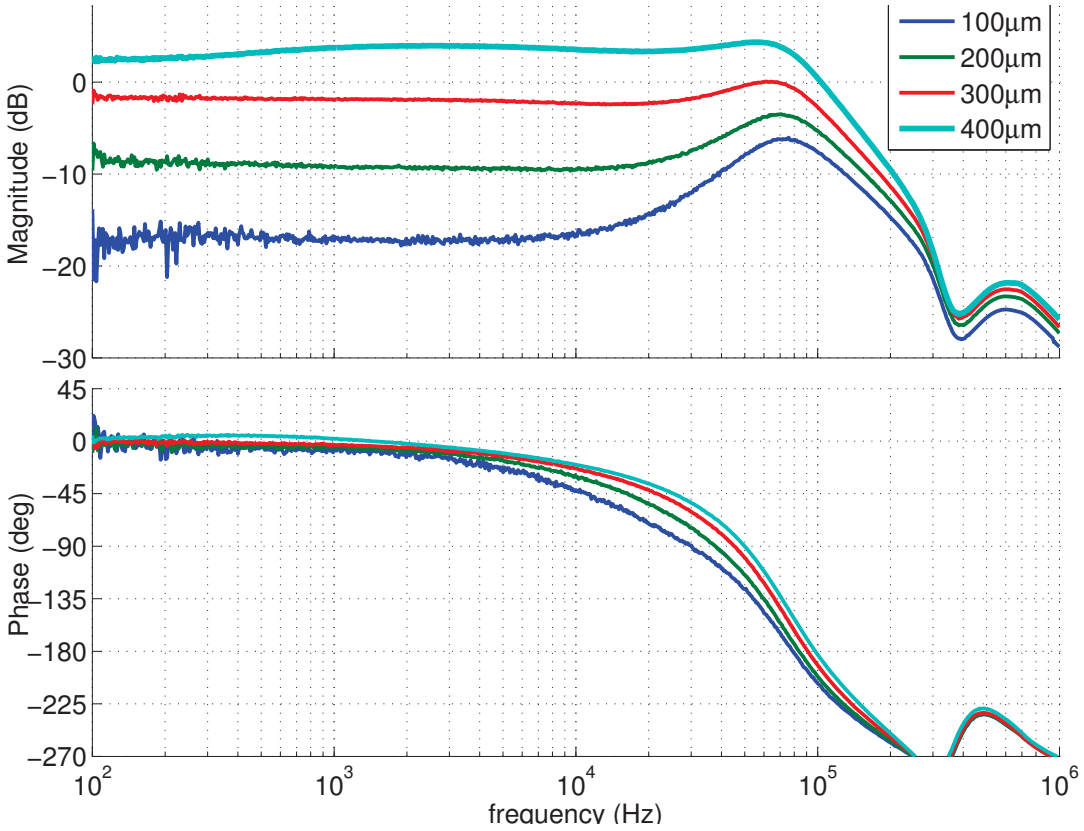


Figure 2.26: Bode plot of optical proximity sensor HDSL9100 acquired by modulated light at different distances to mirror $s = \{100, 200, 300, 400\}\mu\mathrm{m}$

According to the transfer function in Figure 2.26, the optical proximity sensor HDSL9100 can be used at a certain position where no saturation of photodiode occurs (here $s_0 = 400\,\mu\mathrm{m}$) to measure displacement up to $f_{-3\,\mathrm{dB}} = 70\,\mathrm{kHz}$ with $\pm 3\sigma$-resolution of $r(70\,\mathrm{kHz}) = 14\,\mu\mathrm{m}$.

[23] From connector labelled with v_{Lo} to connector labelled with v_L.

2.5 Capacitive sensor

According to [13] the most commonly used sensors in nanopositioning systems are capacitive and eddy-current sensors. Both enables non-contact measurement of displacement, however they require a conductive surface of the target.

2.5.1 Operation principle

Commercially available capacitive displacement sensors mostly utilize a varying gap δ to change their capacitance C (and thus the output in some proportion) approximately expressed by the relation

$$C(\delta) = \frac{\varepsilon A}{\delta} \tag{2.9}$$

in response to a displacement by $\Delta\delta$ with $\delta = \delta_0 + \Delta\delta$ and ε as permittivity of air and A as effective surface of the electrodes.

Compared to strain gauges or optical sensors, capacitive sensors have a high sensitivity in a very limited range of measurement due to the $1/\delta$ relation for small δ. The linearity of the output mostly depends on the geometry as well as constructional parameters (such as the guard ring which eliminates the influence of stray fields to the measurement) as well as on the converter electronics which linearises $C(\delta)$ and maps to an output voltage $v_q \propto \Delta\delta$.

2.5.2 Practical considerations

Usually, capacitive sensors are affordable for the high resolution and high bandwidth they can provide. Table 2.6 gives an impression of performance for commercially available displacement sensors of MicroEpsilon[24]. The $\pm 3\sigma$-resolution is calculated out of the reported RMS-resolution enabling a comparison to other sensors introduced already[25].

tpye	range (absolute)	bandwidth	resolution (RMS)	$\pm 3\sigma$-resolution
series 2800	60 μm	1 kHz	160 pm	0.96 nm
series 2800	500 μm	1 kHz	7 nm	42 nm
series 5504	50 μm	100 kHz	1.9 nm	11.4 nm
series 5501	250 μm	100 kHz	17.9 nm	107 nm

Table 2.6: Range, resolution and bandwidth of capacitive sensors from MicroEpsilon

Capacitive sensors are very sensitive to tilt. According to experiments it is hard to align a short range sensor without having a guidance or bearings, such as a flex structure for a postmount adjustment. A simple radial fixing screw might cause deformation of the sensor's head which is undesirably mapped in the output.

[24] Lowell, Massachusetts, USA

[25] Since there is no explicit information about the distribution of noise, the assumption RMS-noise = 1σ of a normal distribution is made.

2.6 Magnetic sensors

A wide variety of effects can be used to measure the magnetic field. The relevant ones, utilized by sensors of today's technology are summarized in [5] and presented partly in followings.

- The Hall effect reposes on the phenomena that a voltage is being generated in an appropriate material that conducts a transverse current and is exposed to a magnetic flux. Usually, due to very low voltages (several mV) they are rarely used for precise field measurement, however they are commonly used in magnetic controlled switches [5].

- Fluxgates are currently the most sensitive devices for very low magnetic fields, however they require a massive iron core with several coils [44], so they are too large in general to get used for a position measurement e.g. in an AFM.

- Sensors based on the GMR effect produce according to [30] up to 8 times more change in electrical resistance compared to other common types of Anisotrop Magneto Resistance (AMR) sensors. The principle of operation is a quantum effect which occurs in thin, stacked layers of ferromagnetic and non-magnetic materials if they are exposed to a magnetic field [28], [23]. Such sensors are commercially available as integrated devices on a silicon chip for an affordable price. Therefore this sensory principle is investigated in followings.

2.6.1 GMR as displacement sensor

A similar aspect of using magnetic sensors is that measuring the magnetic field is usually not the primary intent. As for optical sensors already discussed in Section 2.4, the displacement s influences in some manner the sensitive quantity, here the local magnetic field strength $\mathbf{H}$ inside the sensor. Due to the principle of operation the magnetic flux $\mathbf{B}$ through a small window (area) inside the sensor is mapped on the output.

The idea to use magnetic sensors for displacement measurement requires an inhomogeneous distribution of a magnetic field $\nabla\mathbf{H} \neq \mathbf{0}$ in order that the sensor measures a change in the local flux density as response to a displacement. The intentionally generated field can be produced by permanent magnets or coils. It is possible to actuate either the sensor in a stationary inhomogeneous magnetic field, or to actuate the source of the magnetic field while the sensor is at fixed location. (*actuated sensor* and *actuated magnet* principle, respectively)
Regarding the magnetic field measured by the sensor, both cases are equivalent. The source of the magnetic field is also irrelevant for the sensor. $\mathbf{H}$ is highly probably a superposition of the intentionally generated field and that of external sources (disturbances) such as the Earth's magnetic field, mains, etc.
Inside such a sensor there is often a Wheatstone-bridge made out of four identical elements sensible to the magnetic field.

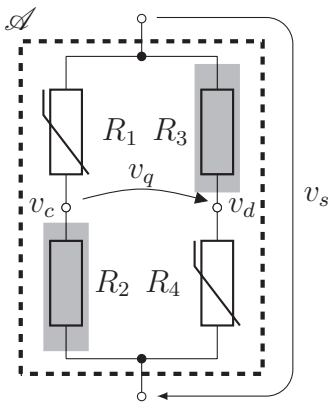


Figure 2.27: Equivalent circuit of a GMR sensor (Wheatstone-bridge)

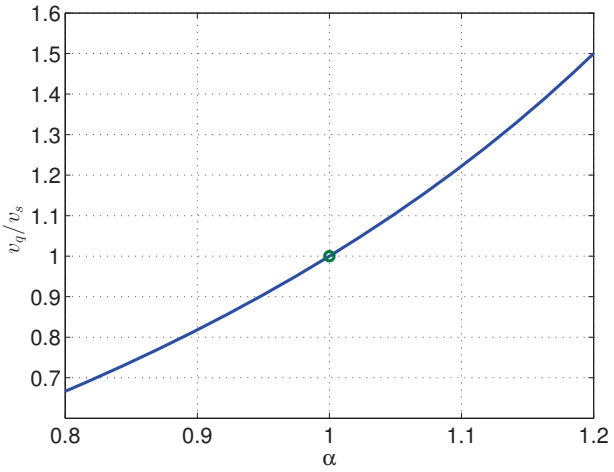


Figure 2.28: Sensitivity v_q/v_s in parameter of α for $R_2 = R_3 = R$, $R_1 = R_4 = (1-\alpha)R$

Wheatstone-bridge

By arranging four sensitive elements (indicated by $R_1 \dots R_4$) inside a sensor in a configuration according to Figure 2.27 a Wheatstone bridge is formed. Two of them R_2 and R_3 have to be covered with Mu-metal in order to make them insensible to the magnetic flux density $\mathbf{B}_l$ linked to the area $\mathscr{A}$[26]. It is assumed, that $\mathbf{B}_l(\mathscr{A})$ is homogeneous within $\mathscr{A}$, to effect all four elements (at least R_1 and R_4) uniformly. By making the simplification

$$R_2 = R_3 = R, \quad R_1 = R_4 = (1-\alpha)R \tag{2.10}$$

where α is some function of $\mathbf{B}_l(\mathscr{A})$ and other constructional parameters of the sensor, (denoted by $\alpha = g(\mathbf{B}_l(\mathscr{A}), m)$ with unknown function $g(\cdot)$ that has to be found empirically) the sensitivity S_W of the Wheatstone-bridge can be derived to

$$S_W = \frac{v_q}{v_s} = \frac{R_2}{R_1+R_2} - \frac{R_4}{R_3+R_4} \overset{(2.10)}{=} \frac{\alpha}{2-\alpha} \tag{2.11}$$

which is shown for $\alpha \in [0.8, 1.2]$ in Figure 2.28. The Wheatstone bridge is balanced if following criterion is met.

$$\frac{R_1}{R_2} = \frac{R_3}{R_4} \implies v_q = 0 \tag{2.12}$$

The GMR sensor used for characterization is the AA002 from NVE[27]. It is obvious to take the highest possible supply voltage v_s to maximize bridge voltage v_q as it can be seen from (2.11). v_s has an upper limit of 48 V because of internal dissipation of the sensor [28], so a symmetrical supply (to ground) of ±24 V is chosen. It can be gathered from [28] that constructional parameter m influences at which magnetic flux density the

[26] Symbolized by the dashed rectangle in Figure 2.27.
[27] Non Volatile Electronics, Minneapolis, MN, USA.

sensor saturates, causing further non-linearities. Therefore, it is necessary to pair the sensor with appropriate geometry (m) to a magnetic field configuration, which is object of the empirical optimization process.

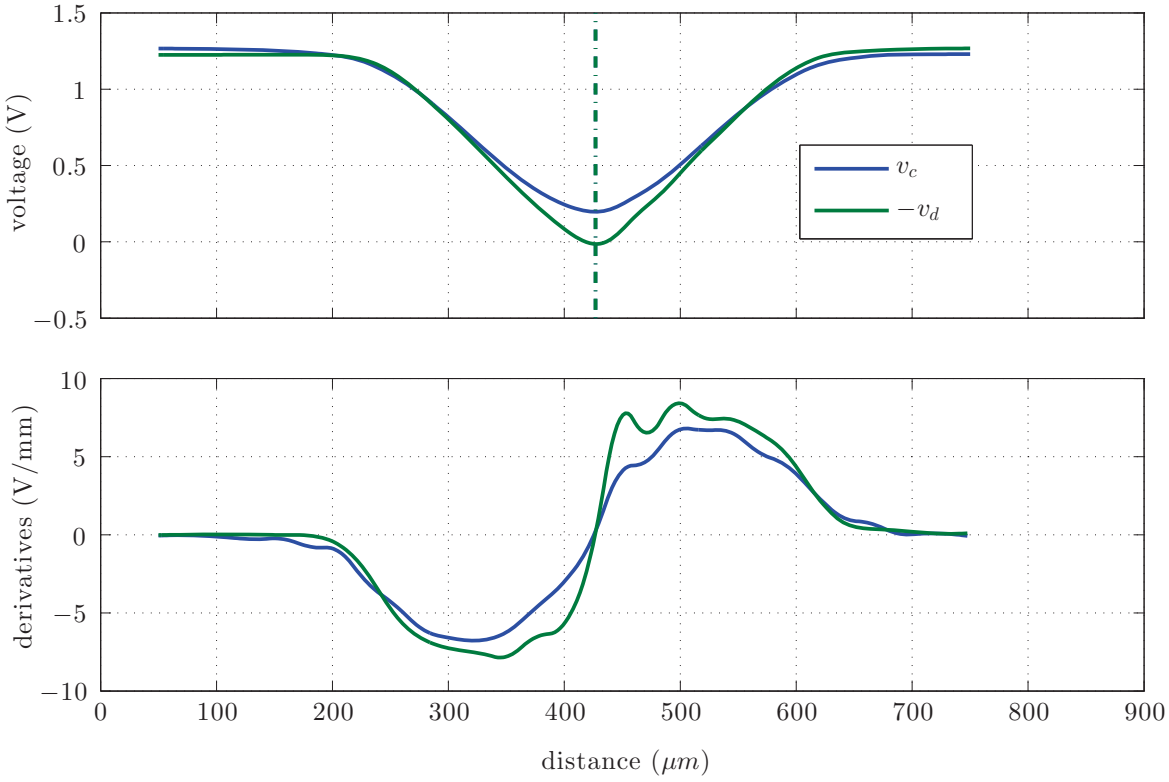


Figure 2.29: Stationary output of the GMR sensor as response to an actuated permanent magnet.

The stationary output $v_q = v_c - v_d$, presented in the upper part of Figure 2.29, results as response to the displacement of the permanent magnet NdBFe40 ($\oslash 4 \times 8$) mm by $\Delta s = 700\,\mu$m. The spatial derivation of v_q is called sensitivity of the sensor, it is shown in the lower part of Figure 2.29 and it is denoted by $S_S = \partial/\partial s\ v_q$. S_S shows a maximum of 6380 V/m at $z_1 = 1.5$ mm (which is another degree of freedom). In Cartesian coordinates, with base vectors $\mathbf{s} = [1,0,0]^T, \mathbf{z} = [0,1,0]^T, \mathbf{x} = [0,0,1]^T$, the output voltage v_q along the $\mathbf{z}$ axis as well as in $\mathbf{s}$ (axis of measurement) is presented as a 3D-plot in Figure 2.32.

The field configuration with the permanent magnet is simulated with Finite Element Methods using FEMM[28]. The field generated by the permanent magnet $\mathbf{H}$ is shown grayscaled in Figure 2.30 with vector plot of flux density $\mathbf{B}$. The sensor is located at the red indicated line, so the projection of $\mathbf{H}$ into the sensor's preferred direction of sensitivity is given by $\mathbf{z}^T\mathbf{H} = H_t$ as it can be seen in Figure 2.31. It shows, that by actuating in direction of $\mathbf{s}$ around the line of symmetry[29] by placing the sensor far off the magnet[30] (z_1) the tangential component of magnetic field strength H_t crosses 0 A/m when the sensor is on the axis of symmetry, thus the output voltage $v_q(s_0)$ is lowest with sensitivity $S_S = \partial/\partial s\ v_q(s_0) = 0$.

[28] David Meeker, Ph.D., Waltham, MA, USA.
[29] The value 2 mm on the abscissa in Figure 2.31 refers to the line of symmetry.
[30] To prevent high fields at sharp edges e.g. on corners of ferrous materials.

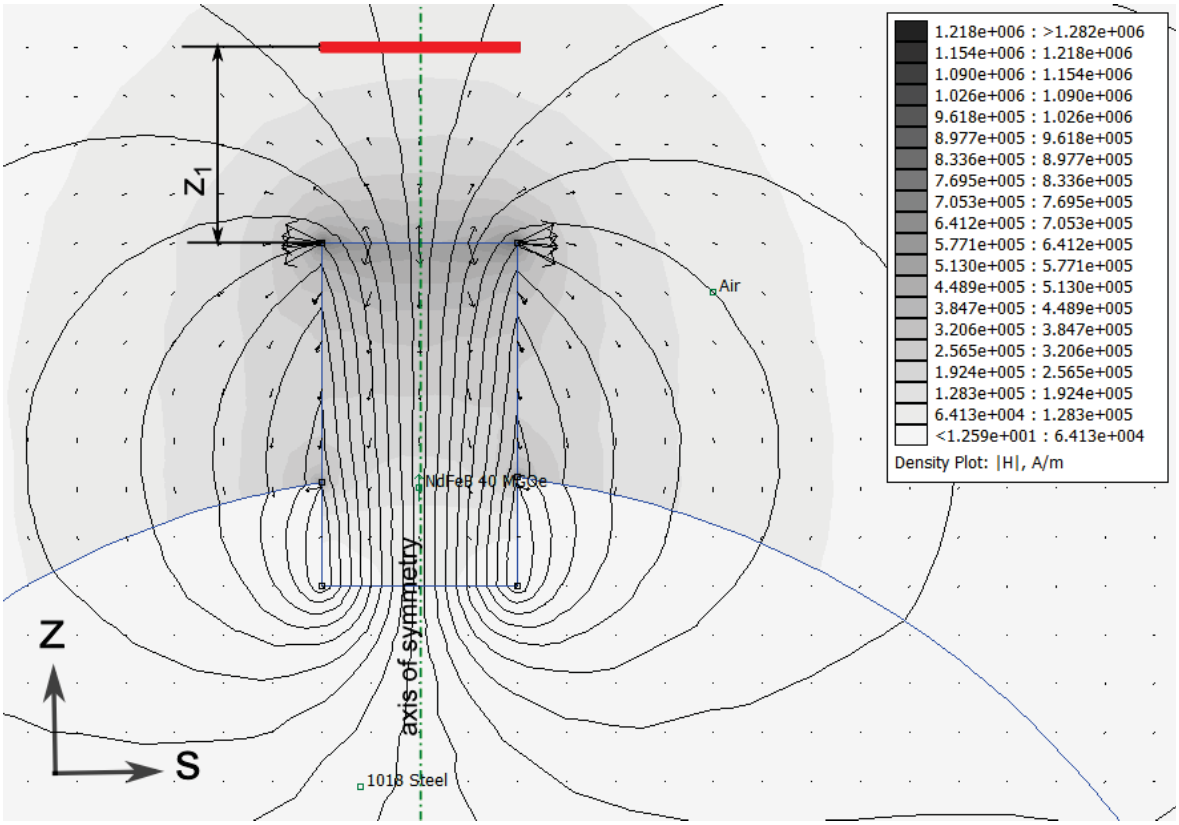


Figure 2.30: FEM simulation of the arrangement with permanent magnet and steel

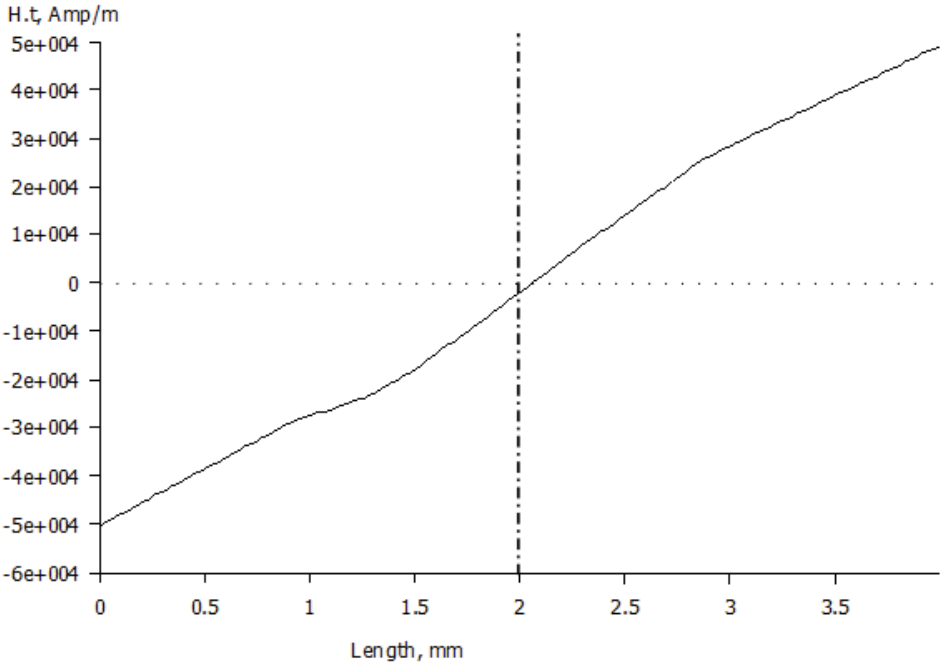


Figure 2.31: Transversal magnetic field H_t along red marked path in Figure 2.30 as it is measured by the GMR sensor

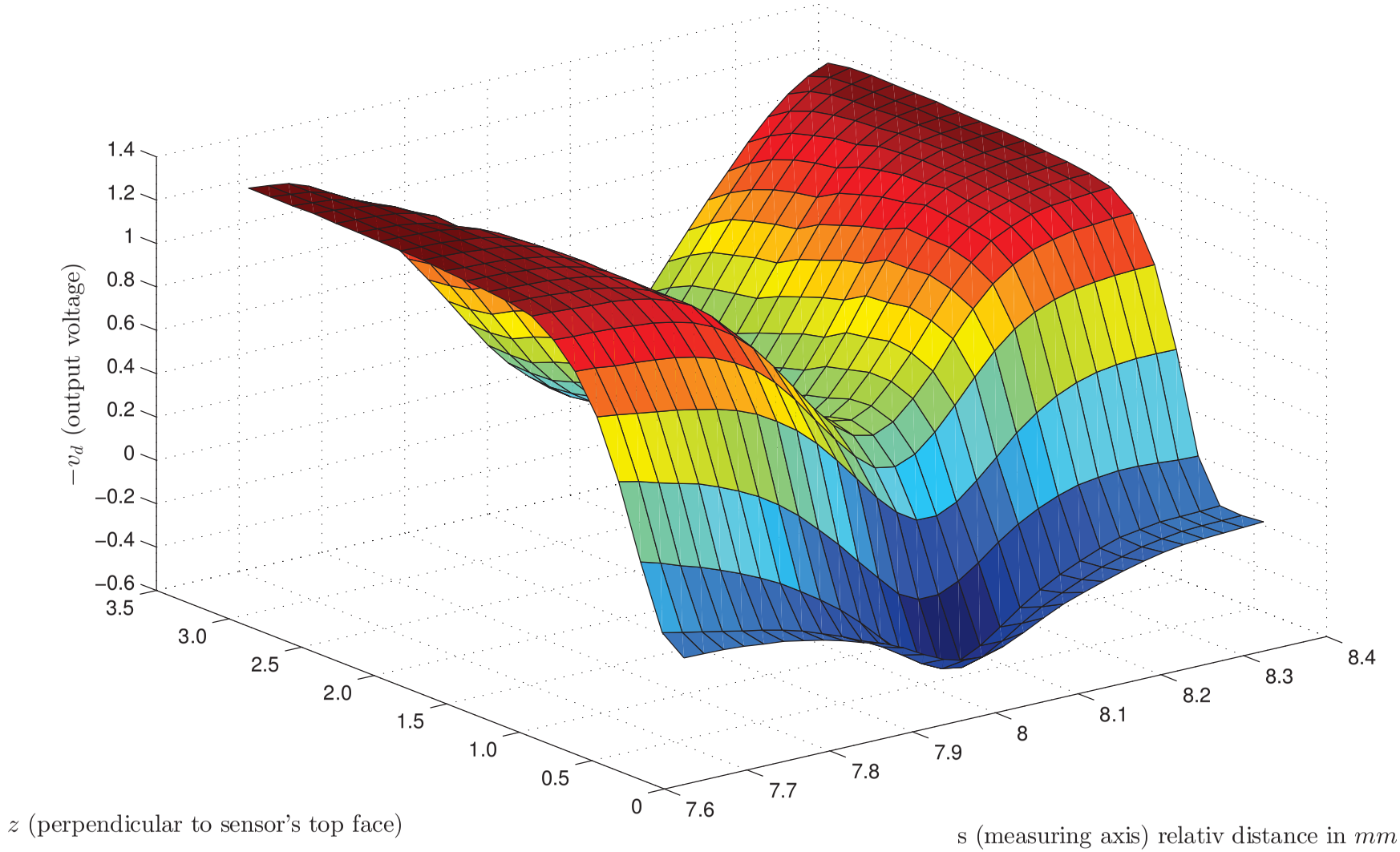


Figure 2.32: Stationary response of the GMR sensor in two axis of actuation.

2.6.2 Temperature drift

Measurements show a high correlation (99%) between output voltage v_q and temperature ϑ. The temperature drift (temperature dependence as introduced in Section 1.4.5) $\mathrm{TD}(v_q) = -230\,\mu\mathrm{V/°C}$ is considered as high. The output voltage is recorded while the GMR sensor is heated up and cooled down in a shielded temperature chamber with very low temperature gradients (rise time $\tau = 20\,\mathrm{min}$ and $\Delta\vartheta = 40\,°\mathrm{C}$) to ensure homogeneous heating. The temperature profile can be seen in Figure 2.33.

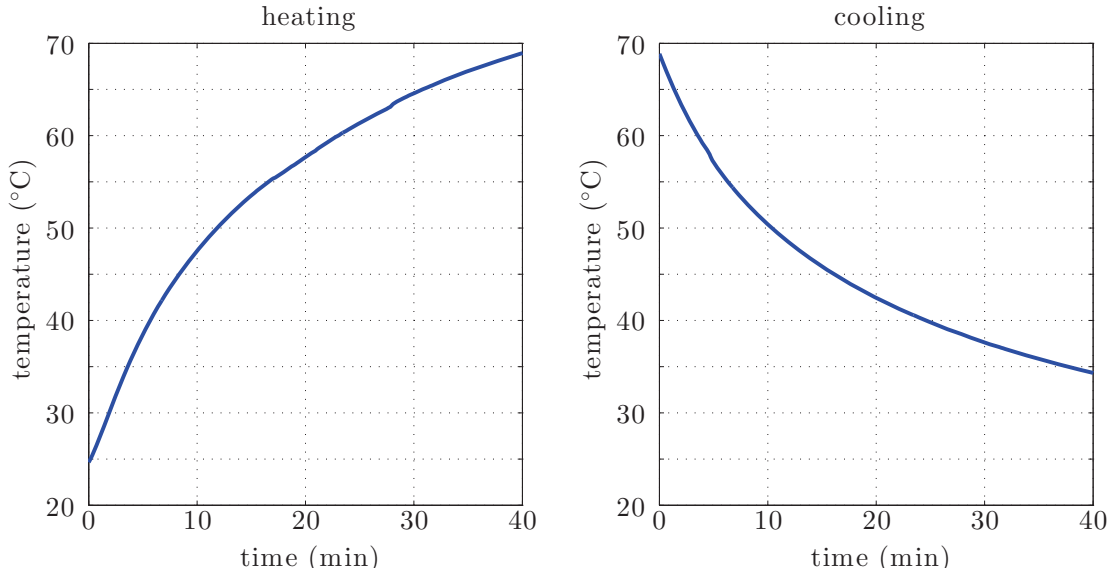


Figure 2.33: Heating and cooling profile of temperature chamber

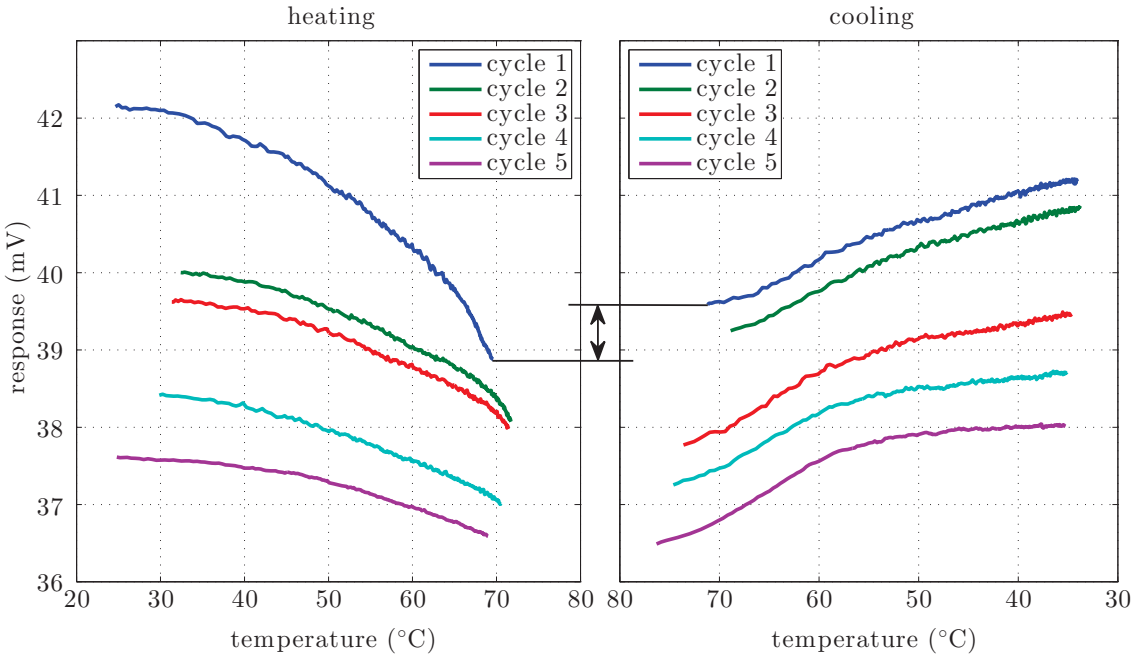


Figure 2.34: Variable temperature drift of the GMR sensor

A long time recording of the sensor signal $v_q(t)$ is performed while the sensor is exposed to 5 consequent cycles of alternating heating and cooling, shown in Figure 2.34. The sensor's output v_q is drawn on the vertical axis towards the temperature on the horizontal axis. The jump in v_q right after stopping the heating, indicated with the arrow in Figure 2.34, arises highly possibly from the magnetic field of the spiral-heater inside the temperature chamber. The reason why it has a varying effect on the measurement is still unclear.
Figure 2.34 shows further that TD(v_q) (the temperature dependence of the bridge voltage) lowers consequently with number of heating and cooling cycles passed (cycle 1...cycle 5). A possible explanation to that might be the lowering bridge voltage v_q and hence a more balanced status of the Wheatstone-bridge. It is widely known that if a Wheatstone-bridge (Figure 2.27) is not balanced (criterion (2.12) is not met) the temperature compensation of its elements do not provide a temperature compensation of the bridge voltage v_q anymore. Thus, the unbalanced status of a Wheatstone-bridge has despite its four identical elements a temperature dependence and hence a drift.

The reason why the bridge gets more balanced just by passing heating and cooling cycles is assumed to arise from the demagnetization of the shielding Mu-metal plates inside the GMR sensor as introduced in Figure 2.27. If this assumption is right TD(v_q) depends on the residual magnetization $\mathbf{B}_r$ producing an operating point dependent offset voltage and hence a non-linearity. To prove (or to disprove) this, the entire sensor (with the shielding Mu-metal plates inside) is demagnetized and TD(v_q) newly evaluated.

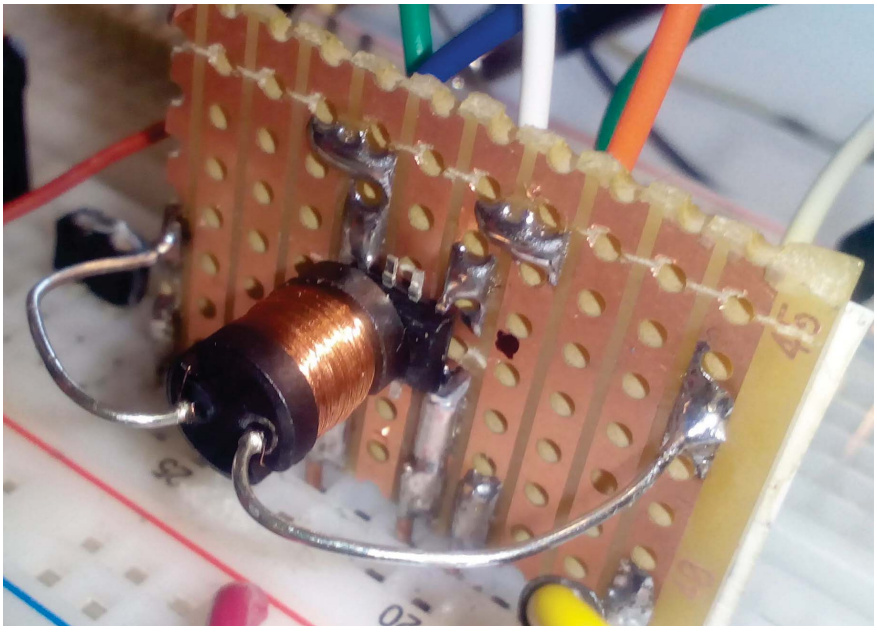


Figure 2.35: GMR sensor with demagnetizing coil

2.6.3 Demagnetization

Due to hysteresis in ferromagnetic materials, they can be demagnetized by cyclical approaching the origo on the $\mathbf{B}/\mathbf{H}$ curve by applying an alternating magnetic field with decaying amplitude produced by a coil with current

$$i_L(t) = \exp\left(-\frac{t}{\tau_L}\right)\sin(\omega t) \tag{2.13}$$

by placing the sensor close to the coil in order use the stray magnetic field of the coil to demagnetise the sensor as the photo in Figure 2.35 shows.

Measurements (Figure 2.36) confirm that demagnetizing the sensor lowered the correlation to temperature by 99% to $|\operatorname{corr}(\vartheta, v_q)| = 0.01$, so $\mathrm{TD}(v_q) \approx 0$, as well as $|v_q| < 0.2\,\mathrm{mV}$ (instead of $v_q \approx 40\,\mathrm{mV}$ before demagnetizing) so that the Wheatstone-bridge is balanced.
On the one hand it could be shown that demagnetizing the sensor eliminates the temperature drift. However, on the other hand it is still not clear whether $\mathrm{TD}(v_q)$ (before demagnetization) arises from a $\mathrm{TD}(\mathbf{B}_r)$ or from a general unbalanced status of the Wheatstone-bridge (as expected during normal operation as displacement sensor).

As the unbalanced status of the Wheatstone-bridge can not be avoided when using the GMR sensor for displacement measurement, it motivates to investigate the AC-measurement bridge (instead of a DC Wheatstone-bridge).

The supply voltage of the AC-measurement bridge is an oscillation with fixed frequency

$$v_s(t) = v_{s0}\cos(\omega_c t), \qquad v_{s0} \in \mathbb{R} \tag{2.14}$$

where $f_c = \omega_c/(2\pi)$ is called the carrier frequency. The response can be formulated as

$$v_q(t) = v_{qr}\cos(\omega_c t) + v_{qi}\sin(\omega_c t) \tag{2.15}$$

with real coefficients, too.

According to [44] an AC field with $f_c = 10\,\mathrm{kHz}$ reduces hysteresis from 2 % to 0.5 % and non-linearity from 3 % to 2 % (in the range of $\pm 300\,\mathrm{A/m}$ field strength). For simplicity, they are using square wave and triggered sampling instead of a sine carrier (2.14) and IQ-detector (magnitude and phase) which might reduce THD and possibly further non-linearities, too.

2.6.4 Resolution

After acquiring the PSD[31] (Figure 2.37) the noise voltage and hence the resolution of the GMR sensor versus the measurement bandwidth can be computed (Figure 2.38). It can be seen that for a 100 kHz bandwidth the $\pm 3\sigma$ resolution is 16 nm or equivalently the RMS-resolution is 2.67 nm.
Hysteresis (inside the sensor) lowers accuracy. It is mostly caused by changes in size and

[31] As for strain gauges already discussed in Section 2.3.4

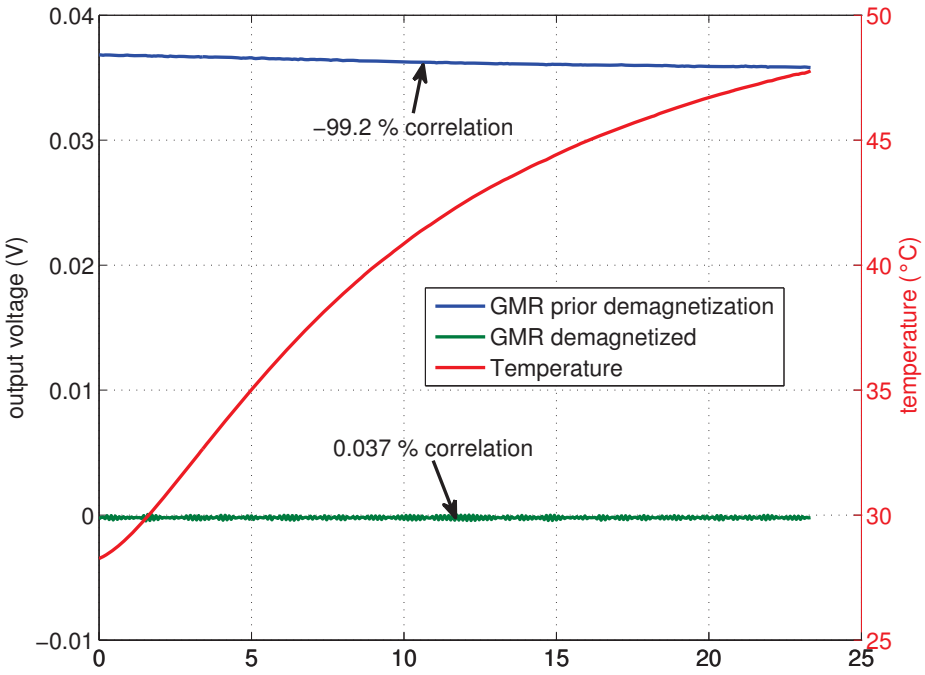


Figure 2.36: Comparison of GMR sensor's output before and after demagnetizing, in relation to temperature

number of magnetic domains in ferromagnetic materials [46]. Another effect is ageing which presumably reduces accuracy and precision. It might be caused by depletion of inner stress.

2.6.5 Bandwidth

By the actuated permanent magnet (acting as source of magnetic field) a bandwidth of $f_{-3\,\mathrm{dB}} = 14.3\,\mathrm{kHz}$ could be achieved. Since the permanent magnet NdBFe40 ($\oslash 4 \times 8$) mm with 2.1 g mass seems to be too heavy to carried by the piezo actuator NAC2025 without excessive resonances, the GMR sensor is characterized by the magnetic field of a solenoid (coil) and current through it

$$i_s(t) = \frac{u_s(t)}{i\omega L} \tag{2.16}$$

with voltage $u_s(t) = u_{s0}\cos(\omega t)$ (with constant amplitude u_{s0}) on the solenoid. It has inductivity L and transfer function of an integrator

$$T_{ui}(s) = \frac{\mathcal{L}(i_s(t))}{\mathcal{L}(u_s(t))} = \frac{1}{sL} \tag{2.17}$$

The frequency response of v_q needs to be corrected for the $-20\,\mathrm{dB}$/decade decay. The result is visible in Figure 2.39, the bandwidth of $f_{-3\,\mathrm{dB}} = 82.4\,\mathrm{kHz}$ can be validated.

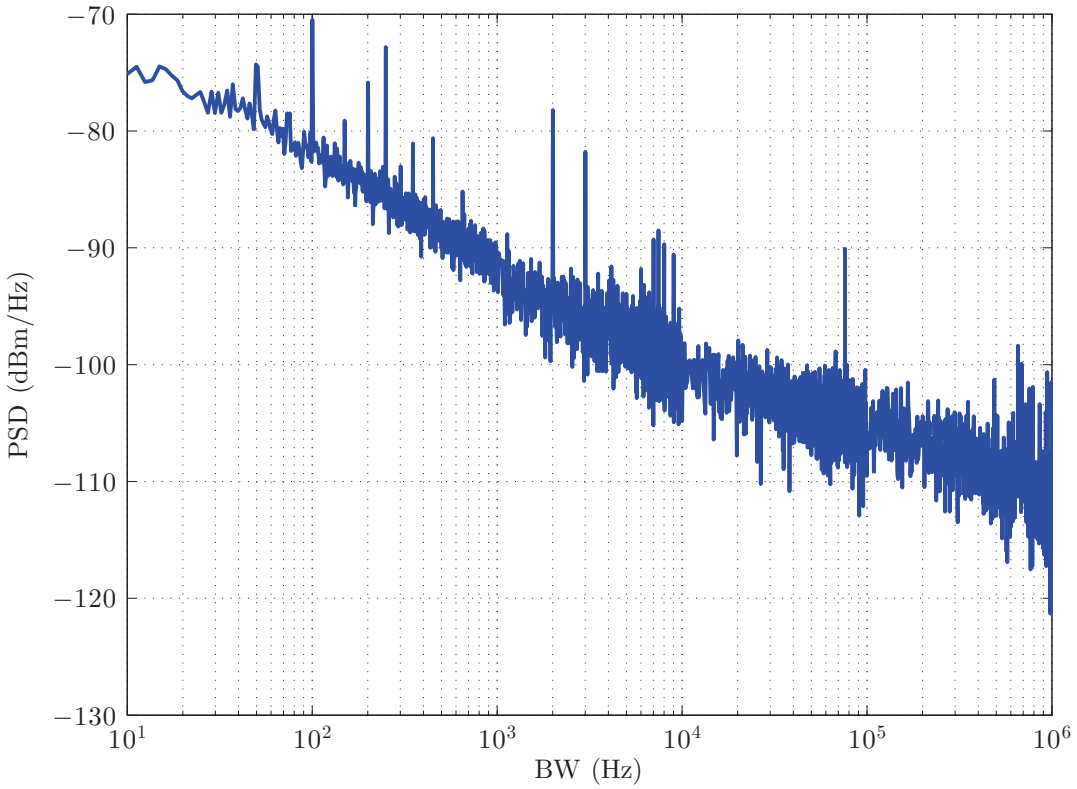


Figure 2.37: Noise PSD of the GMR sensor

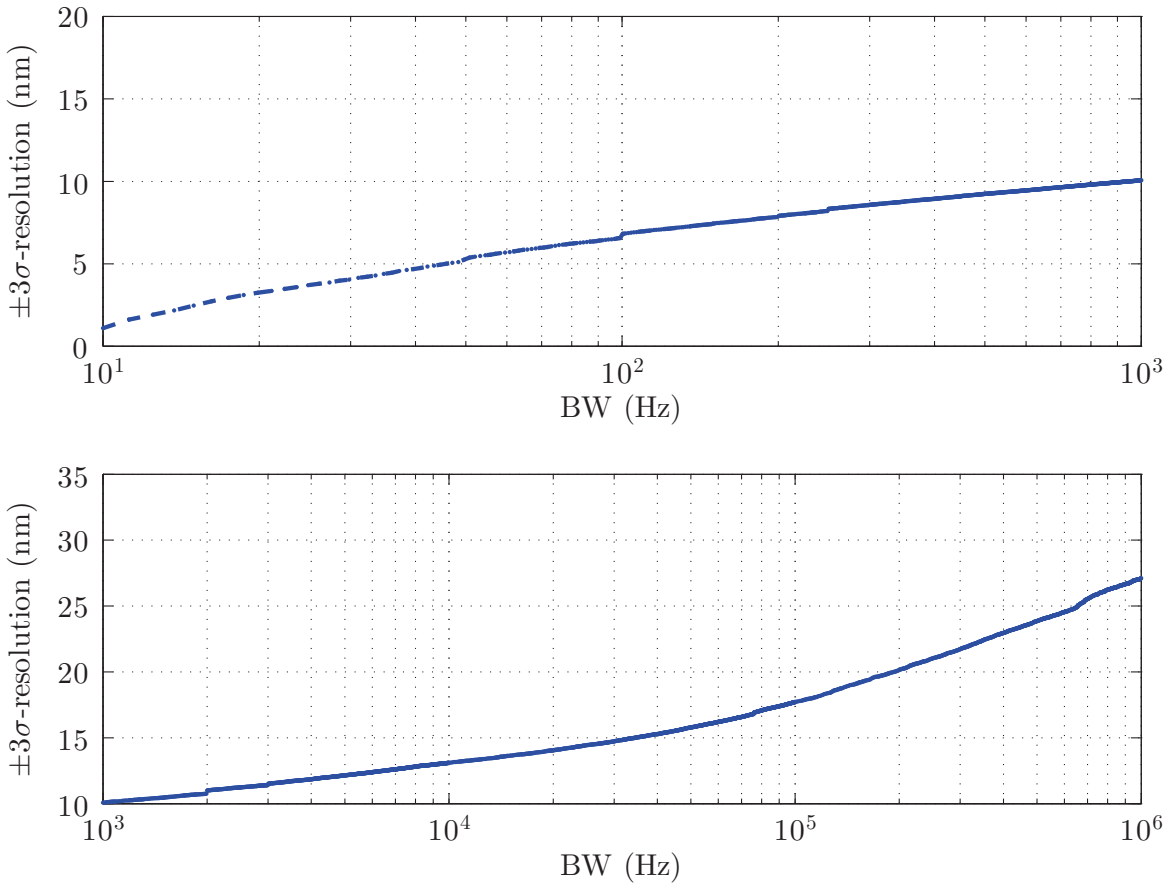


Figure 2.38: $\pm 3\sigma$ resolution of the GMR sensor AA002 on paritioned frequency axis

Out of the datasheet [28] it can be read out that even a bandwidth of 1 MHz is possible. This seems to be confirmed, since similar anisotropic magnetic sensors are used in hard disk drives. For a typical data rates in hard disks of today's technology at 30 MByte/s (1 byte = 8 bits) a frequency of 240 MHz should be possible, however sensors are delivering only binary information (the resolution is not comparable) and there is a huge *knowledge* on signal conditioning built in. However, a much higher bandwidth than 82.4 kHz is assumed to be possible.
The limits of the verification are mostly

- that there are no available actuators which can move by few hundred nanometers at least (SNR) at frequencies above 100 kHz by carrying either the GMR sensor or a small permanent magnet,
- or by the characterization with magnetic fields of coils the impact of high-frequency effects such as the Skin-effect is not known. It would require a numerical simulation of the setup of coil and sensor to determine the HF-magnetic field which the sensor might measure. Alternatively a precise HF magnetic field meter could be used as a reference.

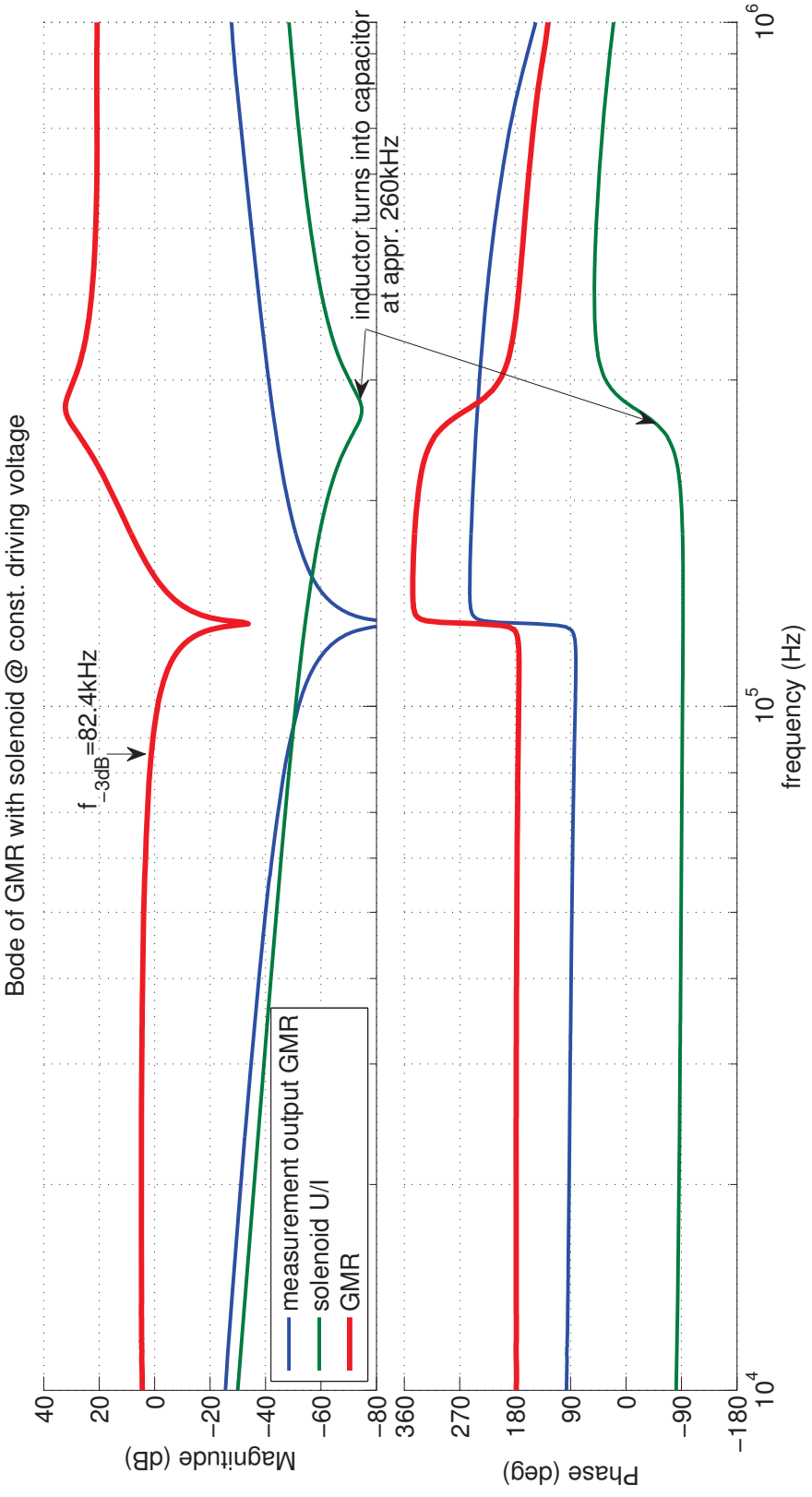


Figure 2.39: Transfer function of GMR sensor, corrected for the low-pass of the coil current.

2.7 Piezoelectric force sensor

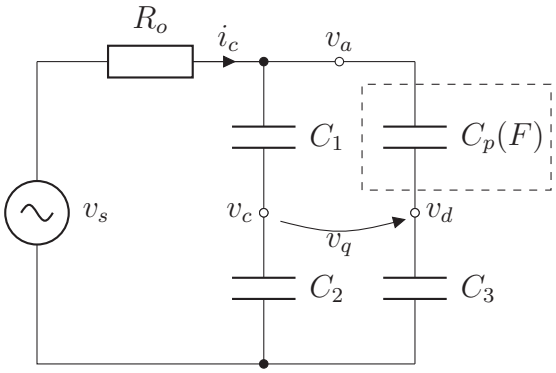


Figure 2.40: Driving piezoelectric actuator $C_p(F)$ in bridge, (self-sensing [41])

The piezoelectric actuator is often modeled as a capacitor C_p. The electric charge Q between conductive layers within a piezoelectric actuator is in good approximation proportional to its elongation s [26]. By considering the following global relationship

$$i = \frac{\mathrm{d}Q}{\mathrm{d}t} \implies Q = \int i(t)\,\mathrm{d}t \propto \Delta s \tag{2.18}$$

it can be easily seen that even small currents $i(t)$ (leakage currents) causes a change in the charge and hence causes a drift of the actuator's elongation over time $s = s_0 + \Delta s$.

As reported in [21] the impedance change of the piezo actuator $C_p(F)$ during actuation with a fixed frequency of sine can be used to monitor forces $F(t)$ (the sum of acceleration forces and external forces) acting on the piezo itself. Therefore, a driving circuit according to Figure 2.40 is necessary.
By chosing $C_1 = C_2 = C_3 = C = 400\,\mathrm{nF}$ as fixed capacitances in the range of the piezo's capacity and by assigning $R_o = 200\,\Omega$ as the concentrated resistance of the amplifier, the AC-bridge has a frequency independent sensitivity given by

$$\frac{v_q}{v_a} = \frac{C - C_p(F)}{2(C + C_p(F))} \tag{2.19}$$

$$\frac{v_q}{v_s} = \frac{C(C - C_p(F))(C + 3C_p(F))}{(2C + 2C_p(F))(C^2 + 3CC_p(F) + 2CR_o + 2R_oC_p(F))} \tag{2.20}$$

which are shown for $C_p = C \pm 20\%$ in Figure 2.41. It can be seen that a high R_o considerably lowers the sensitivity by 5×10^7 and partly therefore, an instrumentation amplifier needs to be used to amplify v_q to a practical range of voltage. It is also necessary because of the high common mode voltage.

The identified transfer function $\mathcal{L}(v_q(t))/\mathcal{L}(v_s(t))$ can be seen in Figure 2.42. In the case if an analytical model of the piezo actuator would exist, the sensor principle might be used for calculating the displacement by integrating the acceleration twice. Since the exact relationship between the forces F ($F \propto v_q$) and the acceleration is not known, this type of sensor is not investigated furthermore. However, in [32] it is shown that

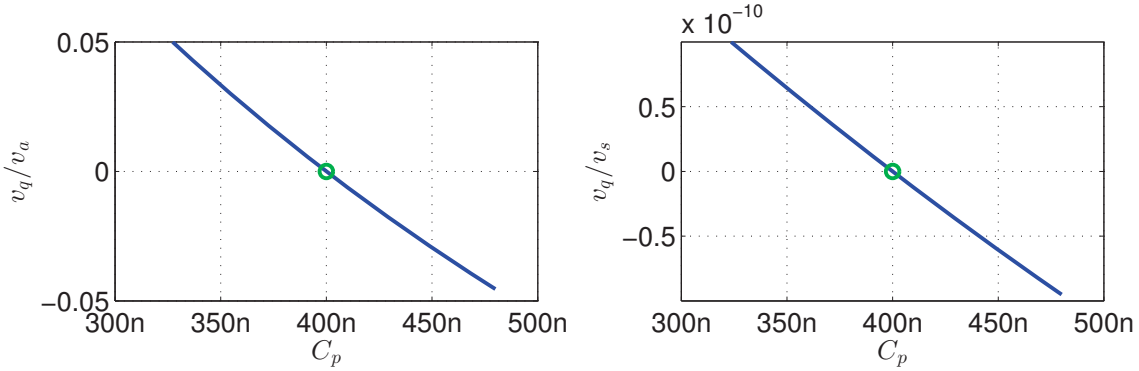


Figure 2.41: Sensitivity of capacitive divider of Figure 2.40

presented technique of self-sensing can be used to damp resonance of piezo actuators by a feedback control.

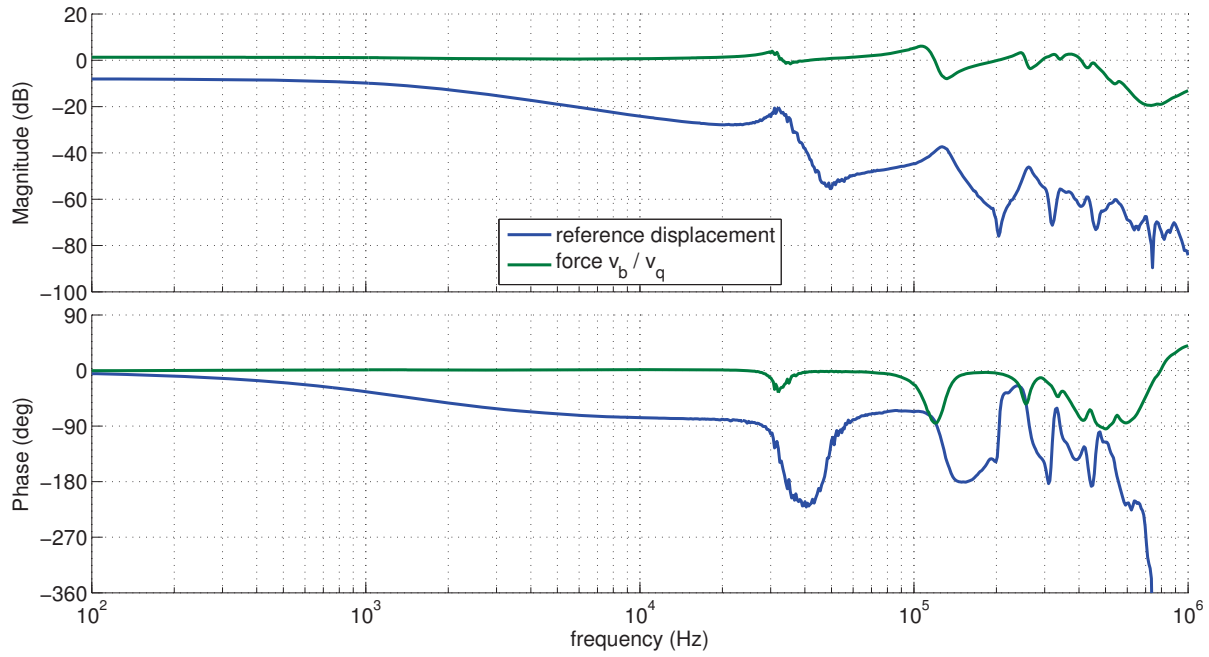


Figure 2.42: Transfer functions displacement and force acquired by impedance change of actuator

2.8 Summary

In followings a short summary of discussed sensors and other displacement measuring principles out of literature are given.

1. **magnetic** field sensors e.g. GMR sensor AA002 from NVE
   - ⊕ bandwidth at least of $f_{-3\,\mathrm{dB}} = 82.4\,\mathrm{kHz}$ (Figure 2.39)
   - ⊕ high resolution $r(10\,\mathrm{kHz}) = 2.0\,\mathrm{nm_{rms}}$ (Figure 2.38)
   - ⊕ sensor is very lightweight (87 mg)
   - ⊕ inexpensive
   - ⊖ temperature drift if not balanced (Figure 2.36)
   - ⊖ interference with any source of magnetic field (can be lowered using AC-bridge)
2. **capacitive** sensors e.g. 5501-250 from MicroEpsilon
   - ⊕ high bandwidth of 100 kHz
   - ⊕ high resolution $r(100\,\mathrm{kHz}) = 1.09\,\mathrm{nm_{rms}}$
   - ⊖ conductive surface of actuator (actuated stage) is required (if already present, the most lightweight sensor)
   - ⊖ usually rather expensive
3. **optical** proximity sensor e.g. HDSL9100 from Agilent
   - ⊕ bandwidth at least of $f_{-3\,\mathrm{dB}} = 70\,\mathrm{kHz}$ (Figure 2.26)
   - ⊖ crosstalk to piezoelectric actuators in their closeness (within 15 mm lateral)
   - ⊕ only reflective surface is needed (lightweight)
   - ⊕ low costs
   - ⊖ moderate resolution $r(10\,\mathrm{kHz}) = 82\,\mathrm{nm_{rms}}$
   - ⊖ influence of ambient light
4. **mechanical** strain gauges e.g. FAET-A6347Q from Micro-Measurements
   - ⊕ high resolution $r(10\,\mathrm{kHz}) = 3.1\,\mathrm{nm_{rms}}$ (Figure 2.19)
   - ⊕ low costs
   - ⊖ temperature drift despite full Wheatstone bridge
   - ⊖ interference with the actuator especially at high frequencies
   - ⊖ low bandwidth of $f_{-3\,\mathrm{dB}} = 20\,\mathrm{kHz}$ (Figure 2.16)
5. **optical** astigmatic method
   - ⊕ high resolution $r(10\,\mathrm{kHz}) = 5\,\mathrm{nm_{rms}}$
   - ⊕ high bandwidth (45 MHz) [DVD pickup unit]
   - ⊕ no significant influence of external light
   - ⊖ *inconvenient* setup due to several degrees of freedom for adjustment
   - ⊖ not as small as others because of 4QPD and optical path

CHAPTER 3

Fusion

Data fusion is the process of combining information from a number of different sources to provide a robust and complete description of an environment or process of interest [8]. In focus to the real time implementation of a position estimation for closed loop control, only causal methods of filtering are presented.

For performance improvement and disturbance rejection of position estimation, sensors based on different physical principles have to be combined to minimize effect of external disturbances in one single domain.

It is also possible to combine not only displacement sensors. It is conceivable to merge data of an acceleration sensor by integrating it twice with a displacement sensor. However, due to practical limitations, especially for nano-positioning in AFMs additional moving mass (acceleration sensor) deteriorates dynamic behavior of the actuator by possibly adding additional resonances, so they are not as preferred as contact-less measurements.

Real sensors suffer next to limited bandwidth also from enhanced sensitivity for some frequencies or insensitivity for others. To compensate for these *linear imperfections*, e.g. the method of filtering with the inverse of the sensor's model is used.

3.1 Model inversion

The aim of model inversion (consensual used term compensation filtering) is to compensate for the uneven sensitivity over the frequency for a given sensor. Sensor's dynamics are assumed to not change with time, thus LTI models are used to describe their behavior. The procedure to obtain filters can be grouped mainly in four tasks.

1. Acquisition: The linear sensor dynamics is acquired by recording a Bode plot containing the transfer function from displacement to the sensor's output. The vibrometer is used as a reference sensor measuring the displacement.

2. Identification: Continuous-time transfer function models are fitted on the measurement, representing an approximation of the sensor's linear dynamics up to a certain degree of complexity (model order). $\rightarrow G(s)$

3. Model inversion: A realizable filter is designed to compensate for the modeled linear imperfections of the sensor over a wide range of frequencies (at least up to a high upper frequency limit). $\rightarrow F(s)$

4. Verification: The downstream connected filter $F(s)$ together with the sensor is validated through measurements, in order to verify modeling simplifications (2.) and the performance of the compensation (3.).

The aim of model inversion can be expressed by

$$G(i\omega)F(i\omega) \approx 1 \tag{3.1}$$

for a wide range of frequencies ω with sensor's transfer function $G(s)$ and compensation filter's transfer function $F(s)$ with the remark to correct the phase to 0°, too. Therefore, it can be summarized as

$$\Delta\big(G(i\omega)F(i\omega)\big) \ll 1\,\mathrm{dB} \tag{3.2}$$

with the variability of a transfer function $\Delta(\cdot)$ according to definition (1.24).

Demonstrative Example

For example, a sensor is modeled with a second order minimum-phase model, i.e. all zeros of the corresponding transfer function (according to (1.21)) are in the LHP of the Laplace-domain,

$$G_1(s) = \frac{C}{1 + 2\xi\frac{s}{\omega_0} + (\frac{s}{\omega_0})^2} \tag{3.3}$$

with gain $C = 10^4$, damping factor $\xi = 0.1$ indicating a decent peaking and resonance frequency $\omega_0 = 1$. The two poles are located at $p_1 = -0.1 \pm 1i \in \mathbb{C}$. Theoretically the inverse transfer function

$$G_2(s) = \frac{1}{G_1(s)} \tag{3.4}$$

with no poles and two zeros at $z_2 = p_1$ would cancel out the poles of $G_1(s)$ resulting in a uniform transmission $G_1(s)G_2(s) = 1$. Unfortunately $G_2(s)$ is not proper and therefore not realizable. Adding at least two LHP poles (or one pair of complex poles) at high frequencies makes it realizable although the amplification needs to be adjusted. Therefore, calculating the inverse by

$$G_3(s) = \frac{1}{1 + G_1(s)} \overset{(3.3)}{=} \frac{1 + 2\xi\frac{s}{\omega_0} + (\frac{s}{\omega_0})^2}{1 + 2\xi\frac{s}{\omega_0} + (\frac{s}{\omega_0})^2 + C} \tag{3.5}$$

with poles in the LHP at $p_3 = -0.1 \pm 100i$ results in a compensator shown on the left in Figure 3.1. It provides a compensation to 0 dB as far as $G_1(s)$ firstly crosses the 0 dB line, although it has also a peak caused by insufficient damping. By moving the poles more to the left on the s-plane to $p_4 = -100 \pm 1i$ an adequate compensation (shown by

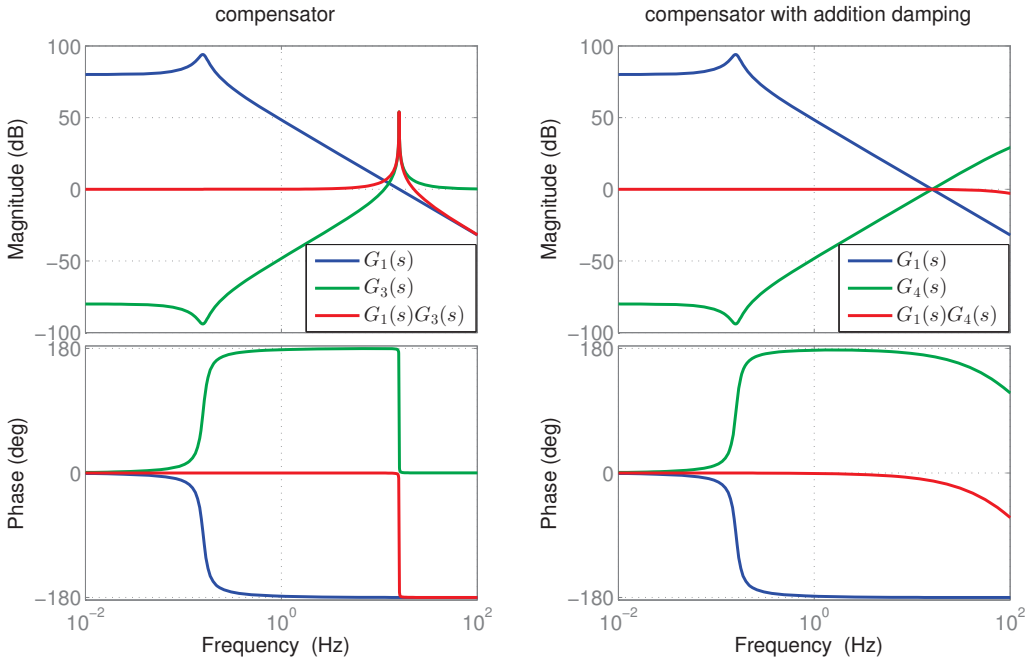


Figure 3.1: Compensator design: Bode plots for sensor $G_1(s)$, inverse dynamics with additional poles $G_3(s)$ or $G_4(s)$ and transfer function of the serial connection of them $G_1(s)G_3(s)$ and $G_1(s)G_4(s)$, respectively.

transfer function $G_4(s)$) can be achieved without peaking, as indicated on the right in Figure 3.1.

There are several practical limitations of zero-pole cancellation.

- A compensation for non-minimum-phase sensors (RHP zeros) can only be done by an unstable compensator with RHP poles, which is avoided due to practical reasons.
- Each measurement is superimposed with noise. Amplification of sensor's signal will also amplify noise (and possibly add some additional noise). Therefore, amplification (and also damping) should be limited to a certain level, de facto until SNR drops below a critical level (e.g. 40 dB).
- Parameter uncertainties and change of transfer function (e.g. temporal or thermal) can evoke that $G_1(s)G_4(s) \neq 1$ even not in bandwidth of interest.

Therefore, compensation of sensor dynamics should be limited to a frequency range where above criteria can be fulfilled. It is required to characterize the sensor dynamics in advance and also the sensor's sensitivity to environmental parameters, such as the orientation of magnetic sensors due to the Earth's magnetic field, tilting of optical sensors due to sudden change of reflectivity, etc. A more advanced method to compensate for parameter uncertainties are model based parameter estimates (e.g. Kalman filter, or robust control).

Generally, the method for compensator design according to (3.5) can be applied on strict proper rational transfer function models of any order, however not for linear models with direct feed-through or for linear models in infinite dimensional normed vector spaces, such as time delays like $G(s) = \exp(-T_d s)$ or even non-linear models. H_∞-optimal filtering (which is discussed in Section 3.4) makes it obsolete to calculate inverse dynamics as presented in this section. Before doing that, a very simple method to fuse sensors is presented in next.

3.2 Complementary filters

A simple method to fuse data from different sensors can take place in the frequency domain by low-pass and high-pass filters. Assuming there are two displacement sensors with overlapping bandwidth (both have *sufficient* signal level in the same frequency range). For example

1. a *fast sensor* with high-pass characteristic $G_H(s)$, e.g. AC-coupled due to high temperature drift of the measurement) and

2. a *slow sensor* with low-pass characteristic $G_L(s)$ with preferably no drift.

For a given crossover frequency f_{co} the filter arrangement as presented in Figure 3.2 can extend the bandwidth of the first sensor by the non-overlapping bandwidth of the second one.

In order to ensure the same sensitivity for both sensors, inverse filters for each according to Section 3.1 are computed, and the transfer functions $G_L(s)$ and $G_H(s)$ in Figure 3.2 already include the terms for the compensation.

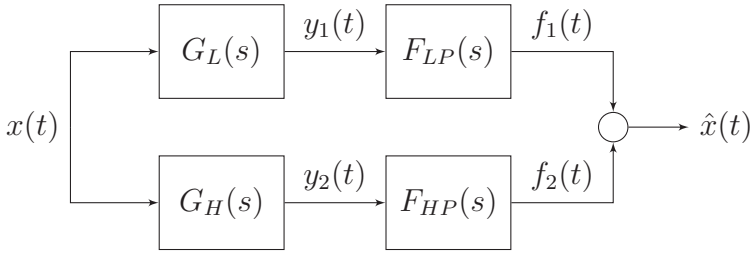


Figure 3.2: Sensor fusion by high-pass and low-pass filtering

3.2.1 Implementation

For the implementation of low-pass and high-pass filters an analogue circuitry may perform well, however as it is discussed in Section 2.2 an ADC is used to sample the sensor's voltage and filters are implemented in digital. This enables advanced methods of filtering e.g. to adjust f_{co} adaptively, according to a time varying variance $\sigma^2(t)$ or a location dependent variance $\sigma^2(x)$ of the one or even both sensors. The digital implementation also enables the usage of high order filters with fast development time.

For the implementation of simultaneous high-pass and low-pass filtering of different sensors at the same crossover frequency f_{co} the method illustrated in Figure 3.3 can be used. The output of the fast sensor $y_2(t)$ may be biased[1] the estimation $\hat{x}(t)$ does not contain it.

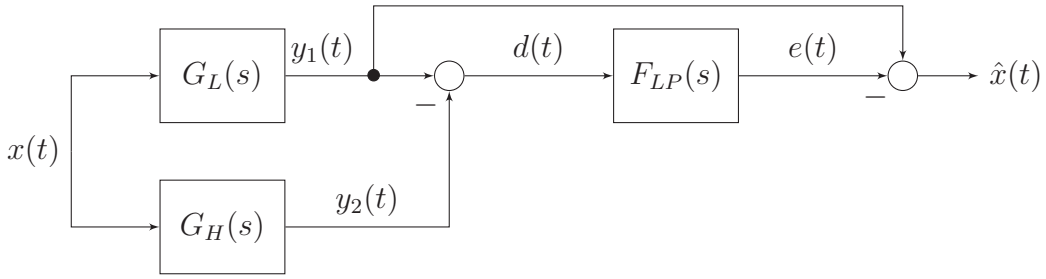


Figure 3.3: Implementation of complementary filtering

For verification, two different filter designs are tested.

a) First order low-pass filter with $f_{co} = 1\,\text{kHz}$ is implemented on a PS (discussed in Section 2.2) with a sampling rate of $f_s = 1$ MSPS,

b) Equiripple linear-phase Finite Impulse Response (FIR) filter is designed in MATLAB by command `firpm()` (based on [24]) and implemented on the same PS at sampling rate up to $f_s = 0.6$ MSPS.

The bandwidth of the *slow* optical sensor HDSL9100 (presented in Section 2.4) can be extended from 7 kHz to 30 kHz by combining with the AC-coupled *fast* magnetic sensor NVE AA001 (presented in Section 2.6). Sensor noise is not influenced, thus the noise of the estimation $\hat{x}(t)$ is approximately equivalent to the sensors before filtering exclusively quantization noise of the ADC. Model uncertainties (and unequal sensitivity of the two sensors despite to filtering by model's inverse) causes a small change in amplitude when crossing f_{co} while sweeping. Some additional phase lag ($|\varphi| > 45°$) can also be observed around f_{co} and at high frequencies ($> 70\,\text{kHz}$). Both phenomena are more dominant with increasing order of the used filter.

[1] It can have any drift with gradient lower than f_{co}.

3.3 Kalman filter

3.3.1 Theory of operation

The Kalman Filter[2] is a linear state estimator for unbiased noise of normal distribution, which recursively calculates a minimum error variance estimation for unknown states of a linear dynamic system, based on periodic measurements. For Linear Time Invariant (LTI) systems the procedure to obtain a steady state Kalman Filter (ssKF) can be divided into

1. problem formulation,
2. offline design (the Kalman gain as solution of the stationary Riccati equation),
3. implementation of filtering (state-space, transfer function) and
4. validation of the performance with parameter uncertainties on a real time PS.

The Multiple Input Multiple Output (MIMO) discrete-time process with sampling time T_s is described by the linear difference equation system

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{G}\mathbf{w}_k \tag{3.6}$$

with the measurement equation

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k \tag{3.7}$$

with the time index k, state vector $\mathbf{x} \in \mathbb{R}^n$, input $\mathbf{u} \in \mathbb{R}^p$, measurement $\mathbf{y} \in \mathbb{R}^q$ and random variables $\mathbf{w} \in \mathbb{R}^r$, $\mathbf{v} \in \mathbb{R}^q$ representing process and measurement noise, respectively. Matrices have appropriate dimensions: $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times p}$, $\mathbf{G} \in \mathbb{R}^{n\times r}$, $\mathbf{C} \in \mathbb{R}^{q\times n}$. $\mathbf{w}$ and $\mathbf{v}$ are assumed to be white noise with normal distribution (Gaussian) and zero mean (unbiased). Additionally, they have to be uncorrelated, i.e. they are of independent source, which is no major limitation since with the assumption $\mathbf{G} = \mathbf{B} \implies r = p$, $\mathbf{w}$ can be considered as the noise of the reference signal $\mathbf{u}$ (input noise), while $\mathbf{v}$ represents the noise of the measurement $\mathbf{y}$ (measurement noise $\equiv$ sensor noise). For LTI systems constant

$$\text{process noise covariance } \operatorname{cov}(\mathbf{w}) = E[\mathbf{w}_k\mathbf{w}_j^T] = \begin{cases}\mathbf{Q} \text{ for } k=j\\ \mathbf{0} \text{ else}\end{cases} \tag{3.8a}$$

$$\text{measurement noise covariance } \operatorname{cov}(\mathbf{v}) = E[\mathbf{v}_k\mathbf{v}_j^T] = \begin{cases}\mathbf{R} \text{ for } k=j\\ \mathbf{0} \text{ else}\end{cases} \tag{3.8b}$$

matrices are assumed, with E as the expectation operator. By using the notation[3]

[2] Named after Rudolf E. Kalman, who introduced in 1960 recursive solution to discrete data filtering problem [36].

[3] *a priori* ... *from the earlier*, *a posteriori* ... *from the later*. The notion of this distinction (like *empirical* and *non-empirical*) comes from Immanuel Kant: Critique of Pure Reason.

$\hat{\mathbf{x}}_k^-$ ..... a priori state estimation, based on measurements $0, ..., k-1$,
$\hat{\mathbf{x}}_k$ ..... a posteriori state estimation, based on measurements $0, ..., k$,
$\mathbf{e}_k^- = \mathbf{x}_k - \hat{\mathbf{x}}_k^-$ ..... a priori estimation error vector,
$\mathbf{e}_k = \mathbf{x}_k - \hat{\mathbf{x}}_k \in \mathbb{R}^n$ ..... a posteriori estimation error vector,
$\mathbf{P}_k^- = E[\mathbf{e}_k^- \mathbf{e}_k^{-T}]$ ..... a priori estimation error covariance matrix,
$\mathbf{P}_k = E[\mathbf{e}_k \mathbf{e}_k^T] \in \mathbb{R}^{n \times n}$ ... a posteriori estimation error covariance matrix.

as used by [35]. The idea of the Kalman filter is to minimize the squared estimation error

$$\min_{\mathbf{K}} E\left(\|\mathbf{e}_k\|_2^2\right) = \min_{\mathbf{K}} \|\mathbf{P}_k\|_2 \tag{3.9}$$

by an appropriate vector[4] $\mathbf{K}$. With equation

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}(\mathbf{y}_k - \mathbf{C}\hat{\mathbf{x}}_k^-) \quad \text{... of measurement update} \tag{3.10}$$

a linear combination between state estimation $\hat{\mathbf{x}}_k^-$ and a weighted difference of new measurement $\mathbf{y}_k$ and measurement prediction $\mathbf{C}\hat{\mathbf{x}}_k^-$ is given. The equation

$$\hat{\mathbf{x}}_{k+1} = \mathbf{A}\hat{\mathbf{x}}_k^- + \mathbf{B}\mathbf{u}_k \quad \text{... represents state prediction.} \tag{3.11}$$

The Kalman filtering process can be considered as a prediction-update formulation, which uses a predefined linear model of the system to predict the state at the next time step [15]. Model errors are corrected using actual measurements of the system. The prediction (3.11) and update (3.10) are combined using the Kalman gain $\mathbf{K} \in \mathbb{R}^{n \times q}$ which is determined to minimize the 2-norm of $\mathbf{P}_k$ (3.9), the mean-square error of the state estimates. Stationary Kalman gain $\mathbf{K}_\infty$ is determined by solving the discrete Riccati-equation backwards in time, e.g. in MATLAB by `kalman()` or `kalmd()` for continuous-time or discrete-time implementation of the plant, respectively. Since, of the assumption of an LTI system, $\mathbf{K}_\infty$ as the solution of the stationary Riccati-equation is the best gain to weight measurements in order to minimize the variance of estimation. Therefore, $\mathbf{K} = \mathbf{K}_\infty$ and this special Kalman filter is called the steady-state Kalman filter (ssKF).

In followings, MISO ssKF (inputs: control input and noisy sensor signal, output: position estimation) for white sensor noise and later on for colored noise (bandwidth limited sensor noise) is being discussed and implementation results are shown.

3.3.2 Kalman filter for white noise

ssKF is designed for the third order ($n = 3$) model of actuation, including an amplifier, a piezoelectric actuator and one ideal sensor (the vibrometer is considered in bandwidth of interest as reference measurement → ideal sensor) in series, denoted by the transfer function $G_{act}(s)$, as plotted in Figure 3.4. It's realization in state-space is presented

[4] The term vector should not be a constrain. For one-dimensional output systems $\mathbf{K}$ is a row vector with n rows, however for multiple output systems it is a matrix with as many columns as outputs.

with differential equations (1.18) in continuous-time with matrices

$$\mathbf{A} = \begin{bmatrix} -7.64 \times 10^4 & -6.25 \times 10^{10} & -1.64 \times 10^{15} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \mathbf{c} = \begin{bmatrix} 1.03 \times 10^4 \\ 1.178 \times 10^9 \\ 1.086 \times 10^{15} \end{bmatrix}$$

and $d = 0$. This is the companion realization (by default in MATLAB) which is numerically ill conditioned (10^{15}). By converting it to the modal form, matrices are

$$\mathbf{A} = \begin{bmatrix} -2.48 \times 10^4 & 2.46 \times 10^5 & 0 \\ -2.46 \times 10^5 & -2.48 \times 10^4 & 0 \\ 0 & 0 & -2.68 \times 10^4 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 226.7 \\ 125.3 \\ 78.5 \end{bmatrix}, \mathbf{c} = \begin{bmatrix} -19.21 \\ -22.83 \\ 223.3 \end{bmatrix} \quad (3.12)$$

and the condition number is reduced to 10 (factor of 10^{14}) and eigenvalues of $\mathbf{A}$ can be seen directly. All poles are in the LHP located at

$p_{sys} = \{-2.48 \times 10^4 \pm 2.46 \times 10^5 i, -2.68 \times 10^4\}$ and zeros also in the LHP at

$z_{sys} = -5.71 \times 10^4 \pm 3.19 \times 10^5 i$.

Conversion to discrete-time system (discretizing)

Due to the finite sampling frequency $f_s = 1/T_s$ (and due to some additional delay because of computation time when implementing Kalman filter) on real systems, performance of filtering by the underlying discretized model compared to that with continuous-time model may degrade.
A zero order Sample-and-Hold with sampling frequency f_s and the transfer function

$$G(s) = \frac{1}{s}(1 - \exp(-T_s s)) \quad (3.13)$$

modifies the continuous-time transfer function $G_{act}(s)$ by producing $\pm$ damping and $\pm$ phase delay as can be seen in Figure 3.4 for two different sampling rates. According Nyquist the frequency of the input has to be limited to $f_s/2$ to avoid aliasing.
Plant (3.12) is discretized with $T_s = 10\mu$s ($f_s = 100\,$kHz) using a Zero-Order-Hold (ZOH) sampling e.g. by MATLAB command `c2d()`. All poles are inside the unity circle as can be seen in Figure 3.5. Plant (3.12) is fully state-observable, since the observability matrix

$$\mathbf{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \quad (3.14)$$

which needs to have full rank in order that the system is fully state-observable, has for the presented system rank($\mathbf{O}$)=3[5]. The Kalman Filter is guaranteed to create a stable nominal dynamics with poles at the eigenvalues of ($\mathbf{A} - \mathbf{KC}$), as long as the plant is fully state-observable [36].

[5] If it would not have full rank, minimal realization needs to be computed before the next step.

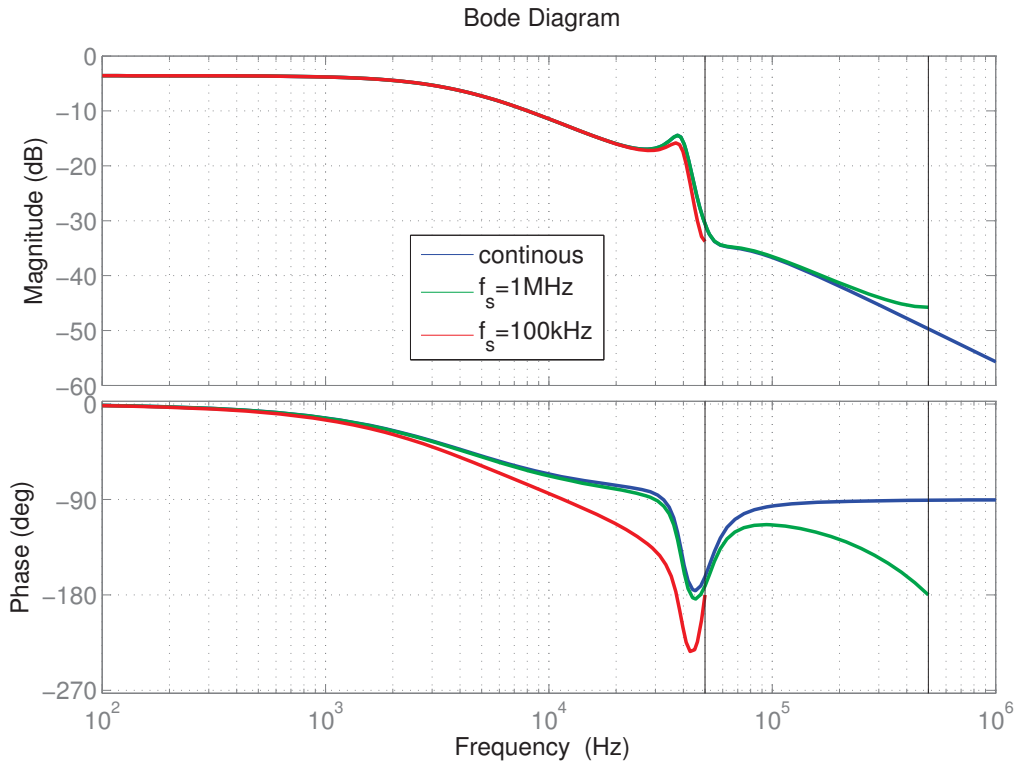


Figure 3.4: Effect of sampling with Zero-Order-Hold at sampling frequency 100 kHz and 1 MHz in comparison to the continuous-time third order model of the actuation system.

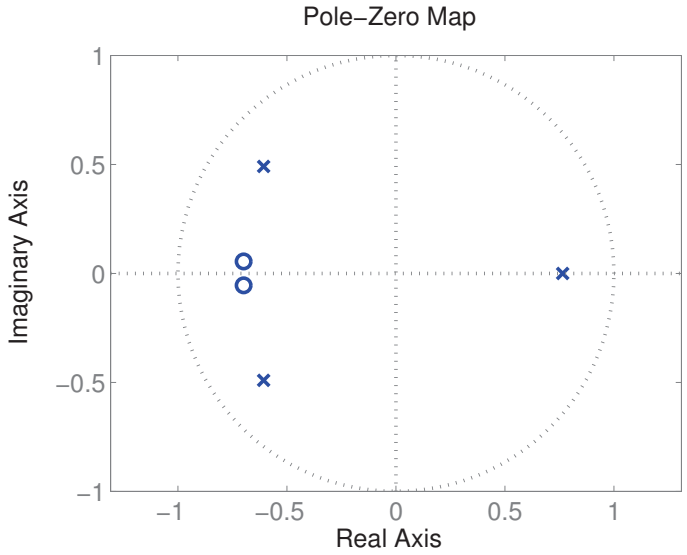


Figure 3.5: Poles(x) and zeros(o) of the discretized actuation system

The Kalman filter is designed for the one-dimensional ($p = 1$) input trajectory u_k, one-dimensional ($q = 1$) measurement y_k, both superimposed with noise, as one-dimensional ($r = 1$) process noise w_k which is assumed to effect system as input u_k by setting $\mathbf{G} = \mathbf{B}$ in (3.6). The stationary Kalman gain for the present configuration is determined to

$$\mathbf{K} = \begin{bmatrix} -1.197 \times 10^{-4} \\ 1.643 \times 10^{-4} \\ 1.752 \times 10^{-4} \end{bmatrix}$$

The filter is implemented as a state-space model in the simulation as in Figure 3.6 and as a set of periodic state prediction (3.11) and update (3.10) equations, in programming language *C* for realization on a PS.

For validation purpose a reference trajectory u_k is chosen to be a step from -1 to 1 at $t = 1\,\text{ms}$, (by exciting multiple frequencies at one time), so step response with both sensor noise and measurement noise can be seen in the time domain in Figure 3.7 labelled by *measurement*. The position estimation and histograms of both measurement and estimation are also shown. The histograms exclude the yellow marked time slots where the initial estimation error settles (due to the unknown initial state vector as this is realistic for real systems) and the level transition slot, which occurs due to the finite rise time of actuation plant.

The presented filtering successfully lowers variance σ^2 (second central moment of normal distribution) by 95.3 % while standard deviation σ is lowered by 78.3 %. This yields an enhancement of sensor's resolution by 78.3 %. To influence performance and minimize estimation error (3.9) $\mathbf{Q}$, $\mathbf{R}$ can be tuned manually or estimated using the Autocovariance Least-Squares (ALS) method. For the presented case they are set to $\mathbf{Q} = \sigma_w^2 = 0.1$, $\mathbf{R} = \sigma_v^2 = 0.1$ where σ^2 represents the assumed noise variance.

Most of sensors in nanopositioning applications operate at a narrow set of frequencies $\mathscr{F} = \{f_1, f_2 \dots f_p\}$ of the trajectory, where the frequencies are well known (such trajectories are e.g. triangular scan signals). Despite the linear model, the response onto input trajectories containing multiple frequencies can be simulated (principle of superimposition) to improve position estimation compared to only measurements.

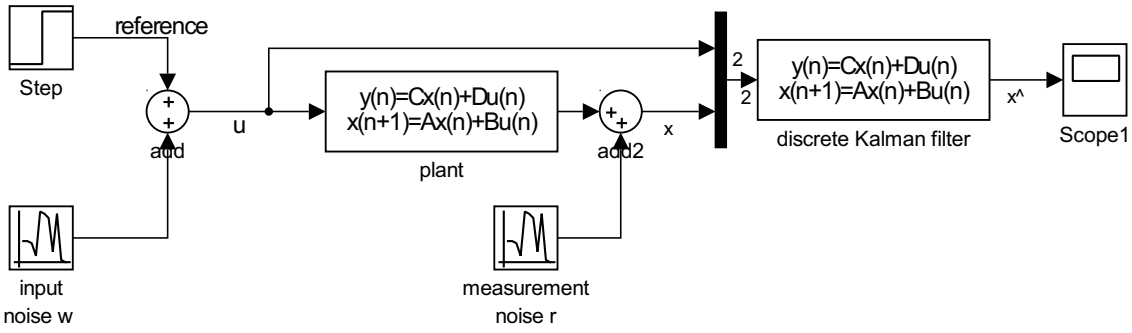


Figure 3.6: Steady state Kalman filter in SIMULINK together with actuation system

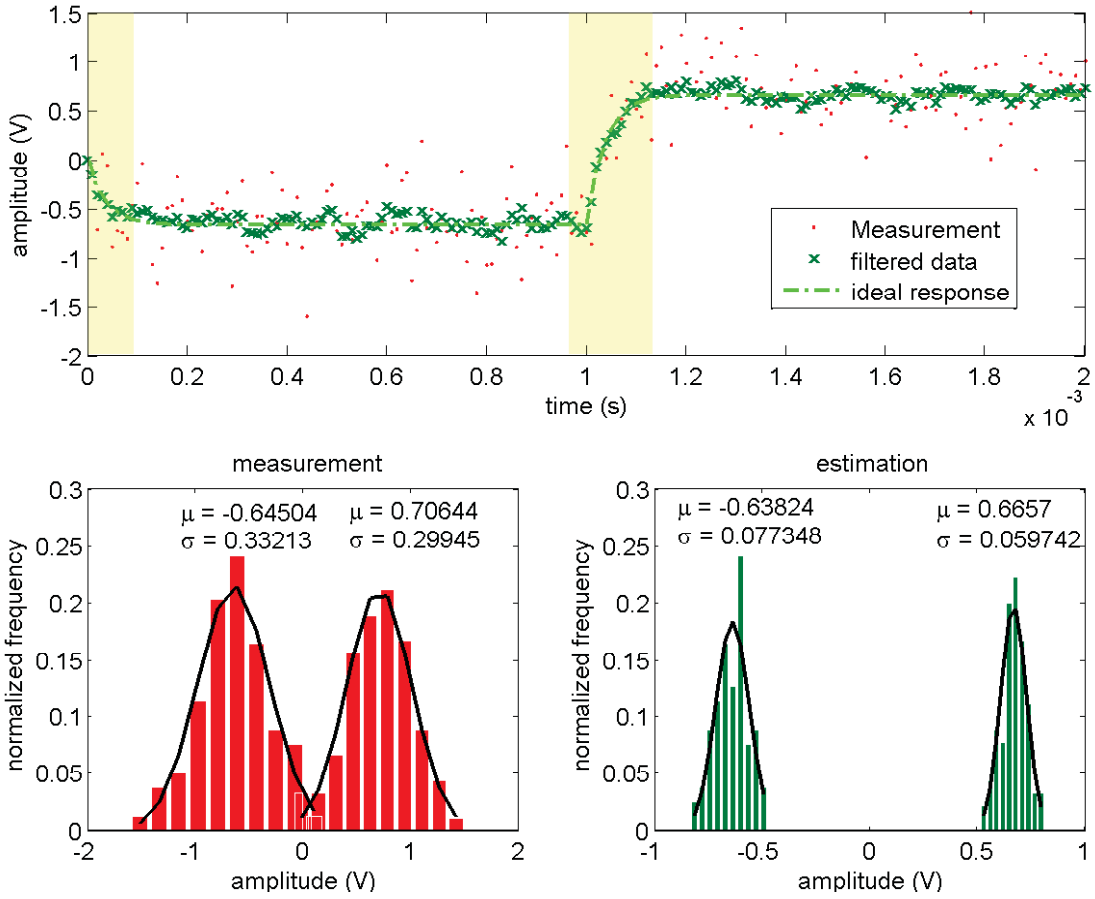
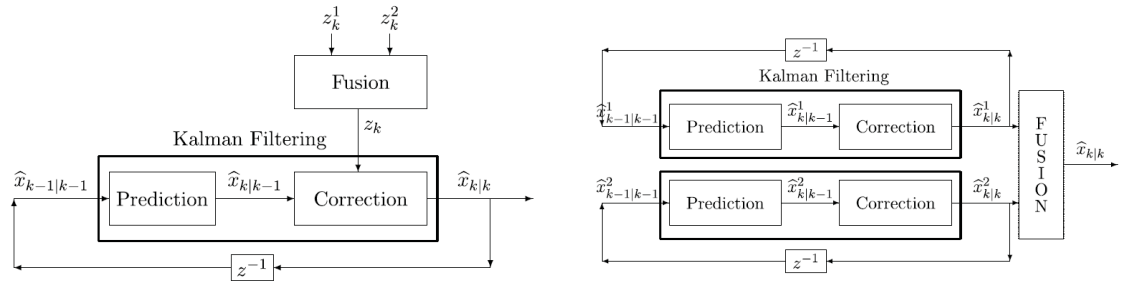


Figure 3.7: Step response of actuation system and Kalman filtered signals (estimation)



(a) measurement fusion (b) estimation fusion

Figure 3.8: Fusion levels for Kalman filtering (taken from [15])

3.3.3 MISO Kalman filter

Assuming there are two sensors with time invariant observation matrices $\mathbf{C}_1, \mathbf{C}_2$ each of dimension $(1 \times n)$ representing an n-th order SISO system for each. Measurement noise v_1 and v_2 is assumed to be not correlated, i.e. $E(v_{1_j} v_{2_k}^T) = 0$ for $\forall j, k$, although in practice some correlated noise might be caused by the common actuation system. As it turns out by targeted measurements, this is one of the limiting factors (next to non-linearities) limiting the performance of filtering.

The sensors have a certain variance[6] expressed by $\text{var}(v_1) = \sigma_1^2$ and $\text{var}(v_2) = \sigma_2^2$, respectively. Fusion takes place at measurement vector level, as illustrated in Figure 3.8a. The two sensors are considered as one group of sensors with augmented observation matrix $\bar{\mathbf{C}}$ so that the Kalman filter filters them according to their variance, independently of each other.

$$\underbrace{\begin{bmatrix} y_{1k} \\ y_{2k} \end{bmatrix}}_{\bar{\mathbf{y}}_k \in \mathbb{R}^2} = \underbrace{\begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix}}_{\bar{\mathbf{C}} \in \mathbb{R}^{2 \times n}} \mathbf{x}_k + \underbrace{\begin{bmatrix} v_{1k} \\ v_{2k} \end{bmatrix}}_{\bar{\mathbf{v}}_k} \tag{3.15}$$

The Kalman filter is designed for augmented MISO system with measurement noise covariance matrix

$$\bar{\mathbf{R}} = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} \tag{3.16}$$

expressing that cross-covariances are zero. The Kalman gain is of dimension $\mathbf{K} \in \mathbb{R}^{n \times 2}$ where each column of it represent the weighting of individual measurements made by the two sensors towards predictions. Implementation results are presented in Section 3.3.7.

3.3.4 Weighted input Kalman filter

Another approach to fuse sensors is to consider each measurement made by each sensor as a separate and independent realization of a Kalman filter, as illustrated in Figure 3.8b. The displacement s is estimated by

$$s_k = y_{1k} + \underbrace{\frac{\sigma_{1_k}^2}{\sigma_{1_k}^2 + \sigma_{2_k}^2}}_{h(\cdot)} (y_{2k} - y_{1k}) \tag{3.17}$$

by weighted sum of filtered sensor signals y_1 and y_2 according to their variance. One major benefit of this method is that ssKF can be designed in advance, only (3.10), (3.11) and (3.17) need to be computed online. However, it is possible to weight independent sensor estimates individually by $h(\vartheta)$ or $h(s)$ according to a temperature varying or position[7] varying variance of the one or other sensor. It is an alternative way to acquire the non-linear system and filter with the so called Extended Kalman Filter (EKF).

[6] Variance $\text{var}(\cdot) \in \mathbb{R}$ is the one-dimensional equivalent to covariance $\text{cov}(\cdot) \in \mathbb{R}^{r \times r}$ of an r-dimensional vector.

[7] $h(s)$ is typical for e.g. magnetic sensors, since the unbalanced bridge has higher variance than a balanced. Optical sensors tend to have a lower sensitivity and higher variance when the saturation of the photodiode is initiated (causing non-linearities).

3.3.5 Extended Kalman filter

Many practical systems can be approximated with time varying linear dynamics. For such, extended Kalman filter is the common tool to estimate for states. For the system described by

$$\mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k + \mathbf{G}_k\mathbf{w}_k \tag{3.18a}$$
$$\mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{v}_k \tag{3.18b}$$

with the same assumptions about the noise as before (3.8)

$$E[\mathbf{w}_k\mathbf{w}_j^T] = \begin{cases}\mathbf{Q}_k \text{ for } k=j\\ \mathbf{0} \text{ else}\end{cases} \tag{3.19a}$$
$$E[\mathbf{v}_k\mathbf{v}_j^T] = \begin{cases}\mathbf{R}_k \text{ for } k=j\\ \mathbf{0} \text{ else}\end{cases} \tag{3.19b}$$
$$E[\mathbf{w}_k\mathbf{v}_j^T] = \mathbf{0}\,\forall\, k,j \tag{3.19c}$$

and initial state estimation $E[\mathbf{x}_0] = \hat{\mathbf{x}}_0$ and initial covariance matrix $E\left[[\mathbf{x}_0 - \hat{\mathbf{x}}_0]\,[\mathbf{x}_0 - \hat{\mathbf{x}}_0]^T\right] = \mathbf{P}_0$, the following set of matrix equations provides the best minimum variance estimation. The Kalman gain is computed by

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{C}_k^T\left(\mathbf{C}_k\mathbf{P}_k^-\mathbf{C}_k^T + \mathbf{R}_k\right)^{-1} \tag{3.20}$$

with update of states and covariance

$$\mathbf{x}_k = \mathbf{x}_k^- + \mathbf{K}_k\left(\mathbf{y}_k - \mathbf{C}_k\mathbf{x}_k^-\right) \tag{3.21a}$$
$$\mathbf{P}_k = \left(\mathbf{I} - \mathbf{K}_k\mathbf{C}_k\right)\mathbf{P}_k^- \tag{3.21b}$$

At last prediction of those is computed

$$\mathbf{x}_{k+1}^- = \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k, \tag{3.22a}$$
$$\mathbf{P}_{k+1}^- = \mathbf{A}_k\mathbf{P}_k\mathbf{A}_k^T + \mathbf{G}_k\mathbf{Q}_k\mathbf{G}_k^T \tag{3.22b}$$

The effort to invert a matrix increases in proportion to the square of its dimension. There are some workarounds to avoid matrix inversion as needed for calculating the Kalman gain (3.20), additionally to preserve symmetry of the error covariance matrix $\mathbf{P}_k$. These are summarized in [8].

Limitation of usage

Usually, a dynamic system is derived out of differential equations of motion and physical relations, so that the elements of the state vector express physical dimensions e.g. position, velocity and acceleration. Often these differential equations are non-linear, and therefore, it is necessary to compute a linearization at each time step k. If the

non-linear model exists, there are several ways for linearization, some of them are based on the expansion of the function to Taylor-series and making an approximation.

However, even by knowing basic concepts of sensors, it is in general very hard to model analytically the one or other peak, drop, or insensitivity as can be seen on many transfer functions of different sensors acquired by measurements.

If a linear sensor model is acquired by identification, a time snapshot of the system $\mathscr{S}$[8] is recorded at a certain time t_1, denoted by $\mathscr{S}_{t1}$. Even if the acquisition is repeated a second time causing $\mathscr{S}_{t2}$ the transition T: $\mathscr{S}_{t1} \xrightarrow{T} \mathscr{S}_{t2}$ is not known. There is no known method of my known to build a non-linear model out of multiple linear acquisitions. At highest, the temporal development of transfer functions can be presented on waterfall plots.

However, even if only $\mathscr{S}_{t1}$ is known, the EKF brings some benefits with time invariant model, compared to a ssKF. Covariance matrices $\mathbf{R}_k, (\mathbf{Q}_k)$ can be adopted to e.g. position dependent noise levels of sensors $\mathbf{R}_k = g(\mathbf{x}_k, \sigma^2_{v_k})$, which have to be recorded in advance (or possibly even estimated in real time).

3.3.6 Kalman filter with noise model

In most of the cases, some more information about sensor noise can be unveiled than only variance. For the LTI system

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{G}\mathbf{w}_k \tag{3.23a}$$
$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k \tag{3.23b}$$

as (3.6) and (3.7), the sensor noise $\mathbf{v}_k$ is modeled as

$$\boldsymbol{\xi}_{k+1} = \mathbf{A}_v\xi_k + \mathbf{n}_k \tag{3.24a}$$
$$\mathbf{v}_k = \mathbf{C}_v\boldsymbol{\xi}_k \tag{3.24b}$$

where $\mathbf{n}_k$ is white noise of dimension l with appropriate matrices $\mathbf{A}_v \in \mathbb{R}^{l\times l}$, $\mathbf{C}_v \in \mathbb{R}^{q\times l}$. Combining (3.24) with (3.23) results in the augmented system [35]

$$\underbrace{\begin{bmatrix}\mathbf{x}_{k+1}\\ \boldsymbol{\xi}_{k+1}\end{bmatrix}}_{\bar{\mathbf{x}}_{k+1}} = \underbrace{\begin{bmatrix}\mathbf{A} & \mathbf{0}\\ \mathbf{0} & \mathbf{A_v}\end{bmatrix}}_{\bar{\mathbf{A}}}\underbrace{\begin{bmatrix}\mathbf{x}_k\\ \boldsymbol{\xi}_k\end{bmatrix}}_{\bar{\mathbf{x}}_k} + \underbrace{\begin{bmatrix}\mathbf{B}\\ \mathbf{0}\end{bmatrix}}_{\bar{\mathbf{B}}}\mathbf{u}_k + \underbrace{\begin{bmatrix}\mathbf{G} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}}_{\bar{\mathbf{G}}}\underbrace{\begin{bmatrix}\mathbf{w}_k\\ \mathbf{n}_k\end{bmatrix}}_{\bar{\mathbf{w}}_k} \tag{3.25a}$$

$$\mathbf{y}_k = \underbrace{\begin{bmatrix}\mathbf{C} & \mathbf{C_v}\end{bmatrix}}_{\bar{\mathbf{C}}}\underbrace{\begin{bmatrix}\mathbf{x}_k\\ \boldsymbol{\xi}_k\end{bmatrix}}_{\bar{\mathbf{x}}_k} \tag{3.25b}$$

for which ssKF can be designed according to Section 3.3 with stationary Kalman gain $\mathbf{K} = \begin{bmatrix}\mathbf{K}_x & \mathbf{K}_\xi\end{bmatrix}^T$, and which can be applied and implemented with (3.10), (3.11). However, due to practical considerations (unaccurate estimation of the noise' amplitude

[8] The system $\mathscr{S} = \{\mathbf{A}_k, \mathbf{B}_k, \mathbf{G}_k, \mathbf{C}_k\}$ of (3.18) denotes the informal collection of matrices.

and thus $\mathbf{K}_\xi$ is very sensitive to the environment) the position is estimated by

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k \tag{3.26}$$

instead of (3.25b).

In order to profit from minimum variance estimation at time varying covariances

$$\bar{\mathbf{Q}}_k = \begin{bmatrix} \mathrm{cov}(\mathbf{w}_k) & \mathbf{0} \\ \mathbf{0} & \mathrm{cov}(\mathbf{n}_k) \end{bmatrix} \in \mathbb{R}^{(n+l)\times(n+l)} \tag{3.27a}$$

$$\bar{\mathbf{R}}_k = \mathrm{cov}(\mathbf{v}_k) \in \mathbb{R}^{q\times q} \tag{3.27b}$$

EKF is implemented with (3.20), (3.21) and (3.22) for the third order actuator system (3.12) with satisfying results.

Compared to the Kalman filter for white noise (Section 3.3.2) the augmented Kalman filter with first order noise model ($l = 1$) of a low-pass[9] the standard deviation of estimation for a constant control input can be reduced additionally by 12%.

3.3.7 Verification

For perfect performance, the Kalman filter requires that the plant dynamics is known exactly. Since the actuator and sensor model for what it was designed is the result of an identification procedure with approximations, the best way to test for model uncertainties is to implement the Kalman filter on a target (e.g. on a PS such as STM32F407 as introduced in Section 2.2) with sensors connected over ADCs.

On the presented PS, SISO ssKF (as introduced in Section 3.3.2) for third order model of actuation system can run up to a sampling rate of 500 kHz, so each 2 μs the position is estimated. This includes simultaneous sampling of one sensor and one reference input with 16 bits resolution, scaling of ADC values, and computing equations of measurement update (3.10) and time update (3.11).

The $\pm 3\sigma$-resolution of the GMR sensor for a 10 kHz measurement bandwidth can be successfully improved from 20 nm to 4.1 nm (increase of resolution by a factor of 5) by the presented Kalman filter. For a more detailed, sixth order model of the actuation, the sampling rate lowers to 103 kHz under the above criteria, however, the resolution is only *slightly* better since it is 4 nm for the same bandwidth of measurement.

For two sensors, by fusioning them either at measurement level (Section 3.3.3 MISO Kalman filter) or at state vector level (Section 3.3.4 Weighted inputs Kalman filter) sampling rate reduces to 368 kHz (by 27% compared to that with a third order model of the actuator). By the simultaneous recording of the actuator's movement with two sensors (GMR and strain gauges (SG) as can be seen in Figure 3.9) and weighted inputs fused Kalman filtering, the positioning accuracy could be significantly improved. Table 3.1 summarizes the statistical parameters.

[9] One single pole at $p_l = -2\pi 10^5$ Hz, motivated from PSD in Figure 2.18.

The process noise covariance is set to $\mathbf{Q} = \text{cov}(\mathbf{w}) = 5 \times 10^{-4}$ while measurement noise variances are estimated out of the measurement and set according to $\sigma^2_{unfiltered}$ in Table 3.1. The resulting Kalman gains

$$\mathbf{K}_{GMR} = \begin{bmatrix} -2.11 \times 10^{-4} \\ 1.74 \times 10^{-4} \\ 3.12 \times 10^{-4} \end{bmatrix}, \mathbf{K}_{SG} = \begin{bmatrix} -1.49 \times 10^{-5} \\ 1.28 \times 10^{-5} \\ 2.58 \times 10^{-5} \end{bmatrix} \tag{3.28}$$

yield a lower *reliability* in case of strain gauge sensor since $\|\mathbf{K}_{GMR}\| \approx 13\|\mathbf{K}_{SG}\|$ due to non-linear effects as possibly originated by over-stretching the strain gauge's glue layer as it can be seen in Figure 3.9.
The Kalman filter can be made more robust to unmodeled noise and unmodeled dynamics by increasing $\mathbf{Q}_k$ in the design. This results in a larger covariance matrix $\mathbf{P}_k$, which involves a smaller gain $\mathbf{K}$ [36].

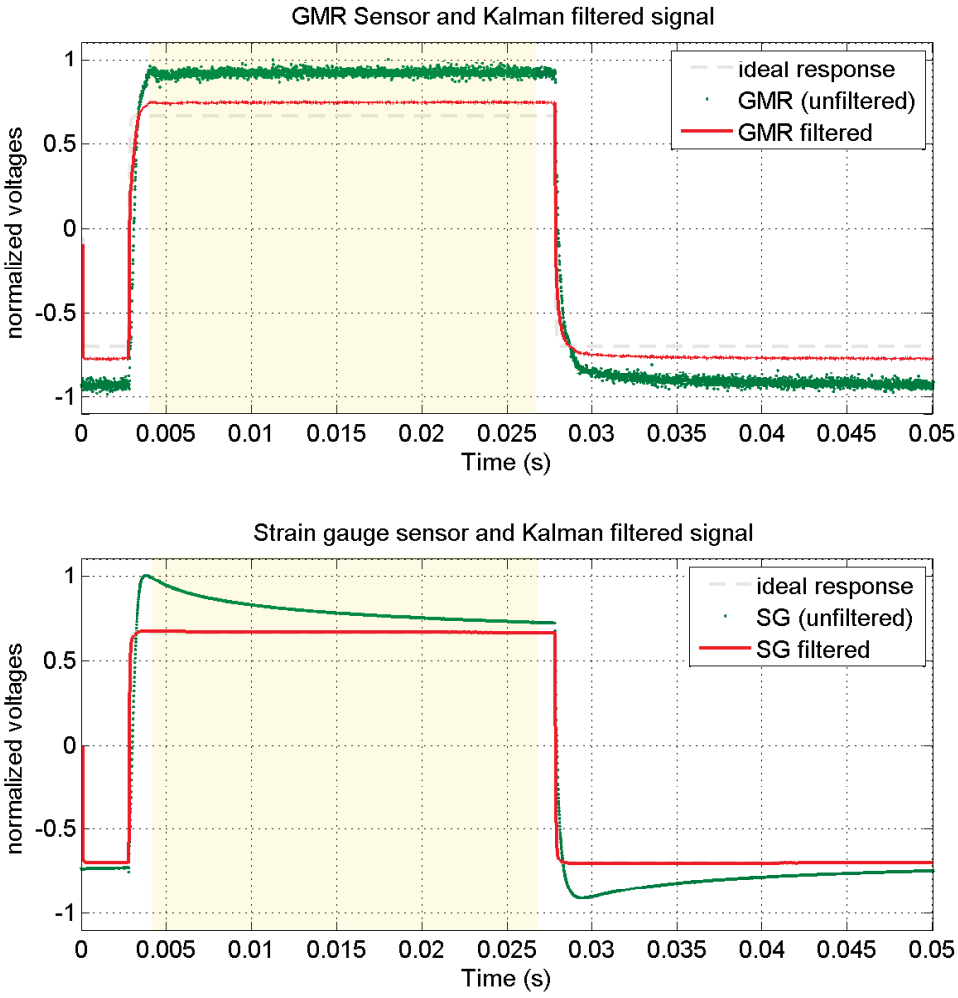


Figure 3.9: Kalman filtered sensors

	$\sigma^2_{unfiltered}$	$\sigma^2_{filtered}$	$\sigma_{unfiltered}$	$\sigma_{filtered}$	factor of resolution improvement
GMR	2.55×10^{-4}	6.08×10^{-6}	1.59×10^{-2}	2.46×10^{-3}	6.47
SG	3.79×10^{-3}	3.42×10^{-6}	6.15×10^{-2}	1.85×10^{-3}	33.2

Table 3.1: Statistics of implemented Kalman filtering

3.4 Robust filters

Unlike the Kalman filter, the technique presented in this section, can only be applied if multiple sensors are measuring the target position simultaneously[10]. The aim of robust filtering can be summarized as follows [1].

For each sensor a filter is designed, by complying the criterion that the frequency response of the sum of all the filtered sensor signals is similar[11] to a predefined transfer function $T_W(s)$, called *weight function*. (*First criterion on robust filters*)

Since the PSD of noise is acquired in advance for each sensor[12] and transfer function models are fitted onto them, the resulting noise models can be included in the filter design extending the performance of the Kalman filter. Usually, H_2 and H_∞-optimal filters make use of the dissimilar frequency distribution of the sensor noise for different types of sensors. The idea to weight the sensor signals according to their noise levels particularly at the given frequency, (i.e. amplifying the sensor with lower noise more than that with higher noise) is realized by robust filters such as the H_2 and H_∞-optimal filters. (*Second criterion on robust filters*)

Among tools of filter design especially the methods of H_2 and H_∞ filter synthesis provide a wide variability of opportunities i.a. satisfying the First and Second criterion on robust filters.
In case of H_2-optimal filtering the method to determine the filters is equivalent to solve a quadratic cost function as the Linear Quadratic Gaussian (LQG) controller does. However the design if performed in the frequency domain in contrast to the LQG where the problem formulation is required to be performed the time domain.
One main advantage of H_∞-filters is that they do not need any constraint about the noise, unlike Kalman filters. They provide an optimal position estimation in terms of a minimum sensor noise over all frequencies, unlike the Kalman filter which is intended to minimize the mean of the squared position error of all frequencies.
One disadvantage of H_∞-filters is that the quantity being minimized does not express any physical dimension, not as the traditional Kalman filter where the quantity being minimized is of the dimension of the position.
In followings definitions of norms are introduced, as well as the strategy to obtain H_2 or H_∞ filters is introduced divided into the *problem formulation* and *filter synthesis* and at last filters are applied on practical problems and measurement results on a test setup with multiple sensors are reported.

[10] Although, this is not a general limitation, only in the extent of this thesis it can not be handled in a more general way.
[11] for methods of filter synthesis with analytical solution even equal...
[12] By measurements as e.g. in Figure 2.17 for strain gauges or as in Figure 2.37 for the GMR sensor.

3.4.1 Definitions

For a strict proper linear dynamic system with the transfer matrix

$$\mathbf{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} \tag{3.29}$$

the 2-norm denoted by $\|\cdot\|_2$ is defined as [37]

$$\|\mathbf{G}(i\omega)\|_2 = \sqrt{\frac{1}{2\pi}\int_{-\infty}^{\infty} \operatorname{trace}(\mathbf{G}(i\omega)^H \mathbf{G}(i\omega))\,\mathrm{d}\omega} \tag{3.30}$$

where superscript H denotes the conjugate transpose. (The argument of trace is the squared Frobenius norm of $\mathbf{G}(s)$.) Usually, the 2-norm is calculated by

$$\|\mathbf{G}(i\omega)\|_2 = \sqrt{\operatorname{trace}(\mathbf{B}^T\mathbf{Q}\mathbf{B})} \tag{3.31}$$

where $\mathbf{Q}$ is the observability Gramian, obtained by solving the Lyapunov equation [37]

$$\mathbf{A}^T\mathbf{Q} + \mathbf{Q}\mathbf{A} = -\mathbf{C}^T\mathbf{C} \tag{3.32}$$

The 2-norm represents the *noise energy*, since it is the mean of the root mean square of the plant's impulse response [36].

The infinity-norm denoted by $\|\cdot\|_\infty$ can be interpreted as the maximum singular value (peak gain) of the plant across all frequencies (*noise power*) due to it's definition [37] extending (1.23) to

$$\|\mathbf{G}(i\omega)\|_\infty = \max_\omega |\mathbf{G}(i\omega)| = \lim_{p\to\infty}\left(\int_{-\infty}^{\infty} |\mathbf{G}(i\omega)|^p\,\mathrm{d}\omega\right)^{\frac{1}{p}} \tag{3.33}$$

The powerfulness of H_∞ filtering is to minimize the worst-case estimation error over all frequencies [36].

3.4.2 Problem formulation

As illustrated in Figure 3.10, the displacement x is measured simultaneously by two sensors modeled as transfer functions $S_1(s)$ and $S_2(s)$. Due to a dissimilar gain and phase of real sensor's in general, inverse filtering of each needs to be done. It is realized by filters $F_1(s)$ and $F_2(s)$ obtained as discussed in Section 3.1. From here on it can be assumed that

$$S_1(s)F_1(s) \approx 1, \quad S_2(s)F_2(s) \approx 1 \tag{3.34}$$

is ensured over the entire frequency range of interest.

According to the model structure in Figure 3.10 it is assumed that the additive noise enters the measurement as a position noise. The band limited noise is modeled by filtering white noise with filters $N_1(s)$ and $N_2(s)$. The noise models are determined by fitting transfer function models to the acquired noise PSDs. The PSD measurement of the GMR sensor is shown in Figure 2.37 while that of strain gauges in Figure 2.17.

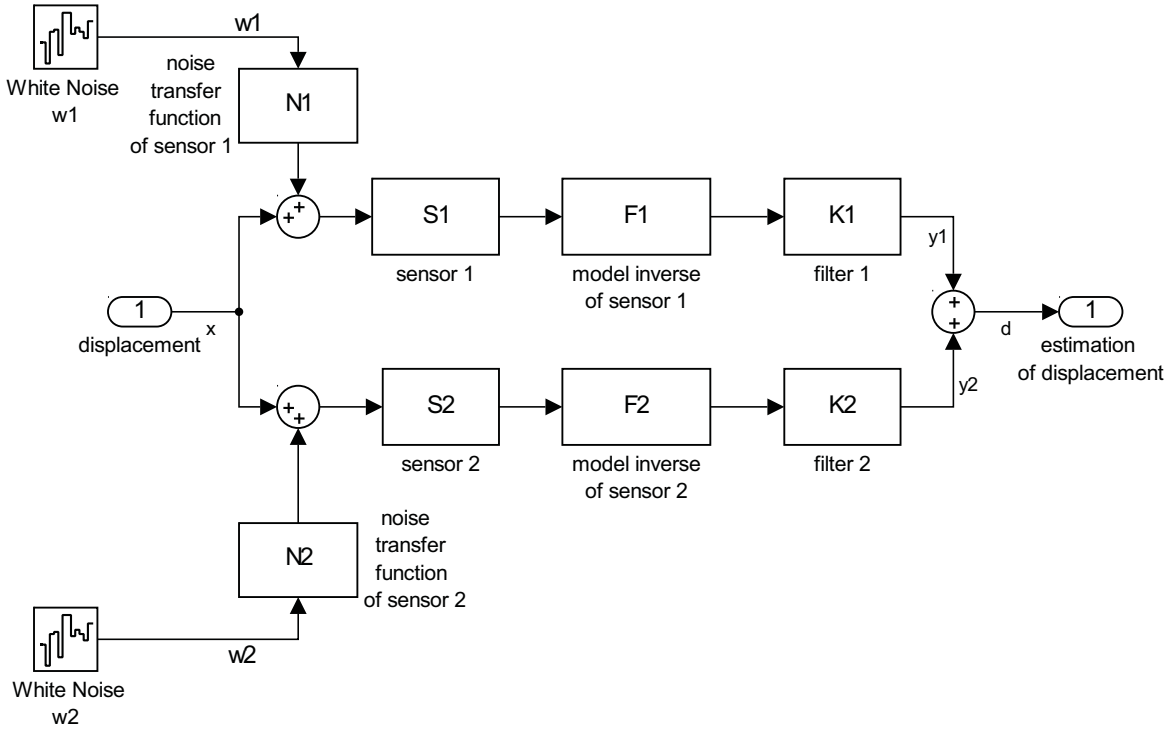


Figure 3.10: Concept of robust filtering: white noise $w_1(t)$, $w_2(t)$ is filtered by noise models $N_1(s)$ and $N_2(s)$, respectively and superimposed to the position measured by the sensors with transfer function models $S_1(s)$ and $S_2(s)$. Inverse model filters $F_1(s), F_2(s)$ and H_2/H_∞-optimal filters $K_1(s), K_2(s)$ are filtering the sensor signals in order make transfer function $G_{xd}(s) \approx 1$

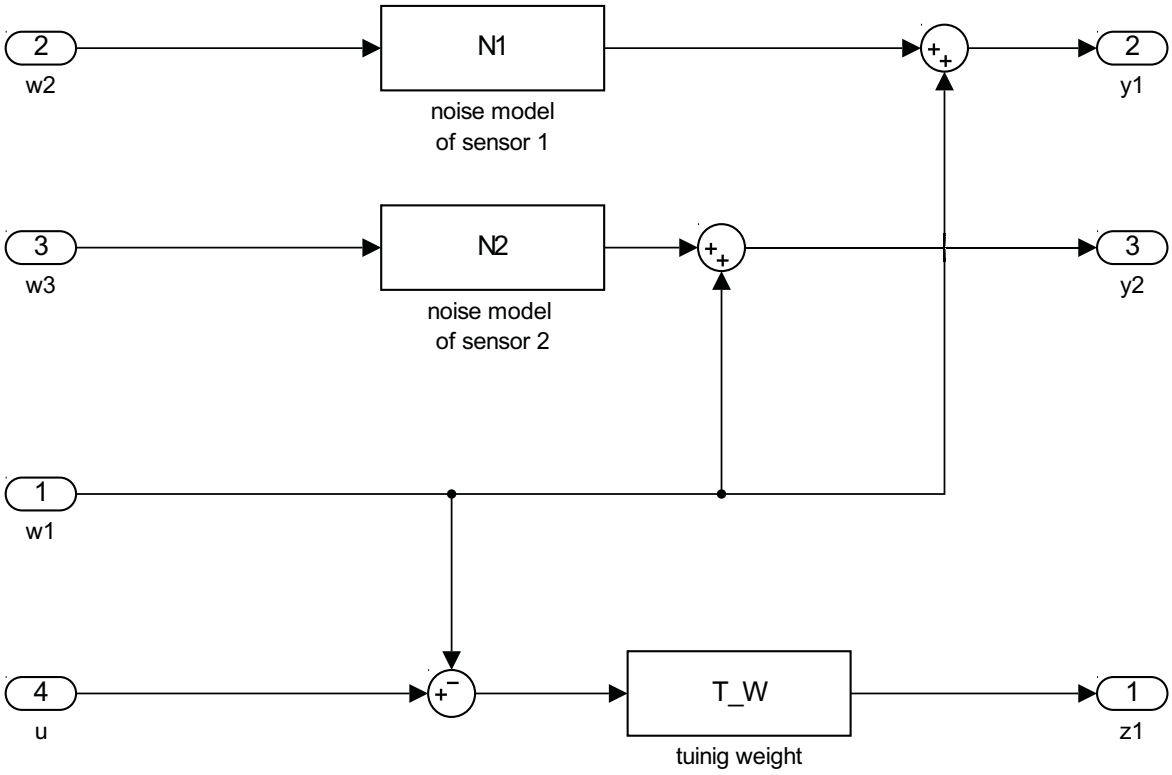


Figure 3.11: Model structure in SIMULINK for obtaining the generalized plant.

The first order fitted models with one pole and one zero per each are determined to

$$N_1(s) = \frac{5.033 \times 10^{-15} s + 3.162 \times 10^{-5}}{1.592 \times 10^{-6} s + 1} \tag{3.35a}$$

$$N_2(s) = \frac{10^{-6} s + 0.06283}{s + 62.83} \tag{3.35b}$$

for the GMR sensor and for strain gauges, respectively. The frequency response can be seen in Figure 3.13.

The strategy to make H_2/H_∞ filter synthesis possible is as follows. A MIMO system $\mathbf{P}$ is augmented out of the noise models $N_1(s)$ and $N_2(s)$ as well as out of a tuning function $T_W(s)$. The input consists of a vector $\mathbf{w}$ and a scalar u. The output consists of the vector $\mathbf{y}$ and a scalar z in accordance to Figure 3.12.

- $\mathbf{w} = [w_1, w_2, w_3]^T$ is the vector of weighted exogenous inputs [37].
- u is the control signal.
- z is the weighted exogenous output.
- $\mathbf{y} = [y_1, y_2]^T$ is the vector of measurements.

They can also be found in the model structure in Figure 3.11.

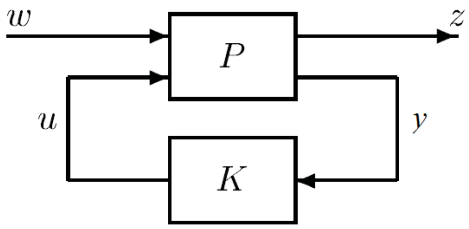


Figure 3.12: Generalized Plant [37]

To covert the equations obtained from the model structure in Figure 3.11 written as a matrix equation[13]

$$\underbrace{\begin{bmatrix} \mathbf{y}_k \\ z_k \end{bmatrix}}_{\boldsymbol{\mu}_k} \overset{s}{=} \begin{bmatrix} 1 & N_1 & 0 & 0 \\ 1 & 0 & N_2 & 0 \\ -T_W & 0 & 0 & T_W \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{w}_k \\ u_k \end{bmatrix}}_{\boldsymbol{\lambda}_k} \tag{3.36}$$

to a state-space representation with vectors of new input $\boldsymbol{\mu}$ and of output $\boldsymbol{\lambda}$, the Linear Fractional Transformation (LFT) is used. The LTI system is determined to

$$\boldsymbol{\xi}_{k+1} = \mathbf{A}_g \boldsymbol{\xi}_k + \mathbf{B}_g \boldsymbol{\mu}_k \tag{3.37a}$$

$$\boldsymbol{\lambda}_k = \mathbf{C}_g \boldsymbol{\xi}_k + \mathbf{D}_g \boldsymbol{\mu}_k \tag{3.37b}$$

[13] with renounced argument s denoted by $\overset{s}{=}$ as in [37]

with matrices $\mathbf{A}_g \in \mathbb{R}^{3\times3}$, $\mathbf{B}_g \in \mathbb{R}^{3\times4}$, $\mathbf{C}_g \in \mathbb{R}^{3\times3}$, $\mathbf{D}_g \in \mathbb{R}^{4\times3}$ for the first order noise models $N_1(s)$ and $N_2(s)$ according to (3.35a) and (3.35b) and the first order tuning function $T_W(s)$ according to (3.41). Alternatively the MATLAB command `linmod()` can be used to obtain 3.37 from the model structure in Figure 3.11 which performs the transformation, too.

As the next step, the matrices $\mathbf{A}_g, \mathbf{B}_g, \mathbf{C}_g, \mathbf{D}_g$ are arranged in following structure to obtain matrix $\mathbf{P}$ to

$$\mathbf{P} = \left[\begin{array}{cc|c} \mathbf{A}_g & \mathbf{B}_g & \mathbf{0} \\ \mathbf{C}_g & \mathbf{D}_g & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & -\infty \end{array}\right] \in \mathbb{R}^{7\times8} \tag{3.38}$$

For a good conditioned numerical computation it is necessary to find a truncated balanced realization of the matrix $\mathbf{P}$ prior to filter synthesis [18], which can be computed e.g. by the MATLAB command `sysbal()`.

3.4.3 Filter synthesis

The MISO filter $\mathbf{K}$ (commonly used term controller [37]) has as inputs the measurements $\mathbf{y}$ and can influence $\mathbf{P}$ by the control signal u and the control equation $u = \mathbf{K}\mathbf{y}$ according to Figure 3.12. The filter $\mathbf{K}$ is determined (synthesized) in the way, that some norm (either the 2-norm or the ∞-norm) of the transfer matrix $\mathbf{G}_{\mathbf{wz}}(s)$ is being minimized.

$$\min_{\mathbf{K}} \|\mathbf{G}_{\mathbf{wz}}(s)\|_2 = \gamma \tag{3.39}$$

(E.g. here the 2-norm particularly.) It can be interpreted as that the filters $K_1(s)$ and $K_2(s)$ in Figure 3.10 are determined by considering the First criterion on robust filters. For the presented model structure the transfer function from displacement to its estimation fulfills the criterion

$$G_{xd}(s) = G_1(s)F_1(s)K_1(s) + G_2(s)F_2(s)K_2(s) \approx 1 \implies \Delta G_{xd}(s) \ll 1\,\mathrm{dB} \tag{3.40}$$

(variability Δ according to (1.24)) up to a certain upper frequency limit which is influenced by the weight function $T_W(s)$.
For the presented case the weight function $T_W(s)$ is found empirically by placing one LHP-pole to 100 kHz and one LHP-zero to 10 MHz, expressed by

$$T_W(s) = \frac{0.01\,(s + 2\pi \times 10^7)}{s + 2\pi \times 10^5} \tag{3.41}$$

The H_2/H_∞-optimal filters $K_1(s)$ and $K_2(s)$ are derived from $\mathbf{K}$. To compute $\mathbf{K}$ there are two approaches.

(a) Solving the two corresponding Riccati equations [37] analytically. (*exact-method*)

(b) The Riccati equations can be reformulated to an optimization problem, and solved iteratively with the method of Linear Matrix Inequalities (LMI). (*LMI-approach*)

In case of minimizing the H_2-norm (a) is mostly possible for practical systems [37]. However when minimizing the H_∞-norm (a) is only possible in some cases under

restrictive prerequisites and by some simplifications. The iterative optimization by the LMI-method (b) may deliver a (sub)optimal solution, however it does not require some restrictive prerequisites [37].

The commands `h2syn()`, `hinfsyn()` or `hinflmi()` of the Robust Control Toolbox can be used to compute the (sub)optimal filters.
If a filter $\mathbf{K}$ is found so that $\gamma < \infty$ the filters can be evaluated. For the presented inputs, the H_2-optimal filters $K_1(s)$ and $K_2(s)$ (realizable, strict proper) are determined, given in continuous-time state-space representation by

$$K_1 \overset{s}{=} \begin{cases} \dot{\mathbf{x}}(t) = \underbrace{\begin{bmatrix} -6.28 \times 10^7 & 52.5 & 175 \\ 0 & -6.09 \times 10^4 & 1.08 \times 10^6 \\ 0 & 1.08 \times 10^6 & -1.98 \times 10^7 \end{bmatrix}}_{\mathbf{A}_F} \mathbf{x}(t) + \underbrace{\begin{bmatrix} -39.2 \\ -2.43 \times 10^5 \\ 4.31 \times 10^6 \end{bmatrix}}_{\mathbf{b}_{F1}} y_{n1}(t), \\ y_{f1}(t) = \underbrace{\begin{bmatrix} -2.5 \times 10^5 & 0 & 0 \end{bmatrix}}_{\mathbf{c}_F^T} \mathbf{x}(t) \end{cases} \tag{3.42}$$

$$K_2 \overset{s}{=} \begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}_F \mathbf{x}(t) + \underbrace{\begin{bmatrix} -210 \\ 2.43 \times 10^5 \\ -4.31 \times 10^6 \end{bmatrix}}_{\mathbf{b}_{F2}} y_{n2}(t), \\ y_{f2}(t) = \mathbf{c}_F^T \mathbf{x}(t) \end{cases} \tag{3.43}$$

where $y_{n1}(t)$ and $y_{n2}(t)$ are the noisy sensor signals as inputs to the filters and $y_{f1}(t)$ and $y_{f2}(t)$ are the filtered signals as outputs of the filters. By the assumption of perfect compensation (3.34) the transfer function $G_{xd}(s)$ of (3.40) reduces to $G_{xd}(s) = K_1(s) + K_2(s)$, which can be evaluated shown in Figure 3.14.

All three poles of the filters are in the LHP located at

$$p_{H2} = \{-6.28 \times 10^7, -1.99 \times 10^7, -1.98 \times 10^3\}.$$

The spectral radii of the filters determine the minimal sampling frequency of the discretized filters in order to avoid distorsion of the dynamics. The spectral radius ρ of a matrix $\mathbf{A}_F$ is defined as the maximum among the magnitudes of its eigenvalues,

$$\rho = \max(|\lambda_i|) \tag{3.44}$$

where λ_i are the eigenvalues of the matrix $\mathbf{A}_F$. For the filters (3.42) and (3.43) $\rho_{H2} \approx 6 \times 10^7$ implies fast modes.

To implement the filters without distorting the dynamics significantly, they need to be discretized and following criterion has to be met

$$f_s > 2\rho \tag{3.45}$$

with sampling frequency f_s according to Nyquist. This would require a sampling frequency for the presented filters of at least 19 MHz, which is considered in connection with the ADC presented in Section 2.2 having a maximum of sampling rate of 2.5 MHz as high. Techniques of model order reduction and elimination of fast modes are summarized in Section 3.4.5.

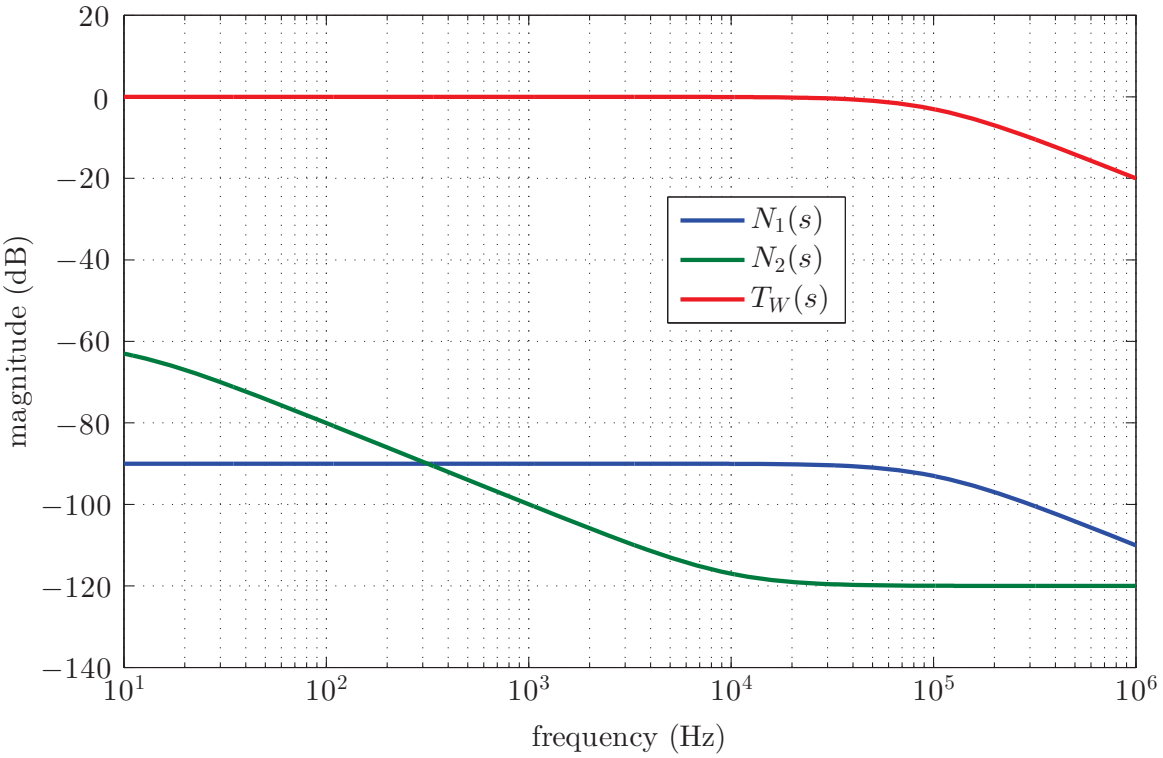


Figure 3.13: Noise and weight transfer functions

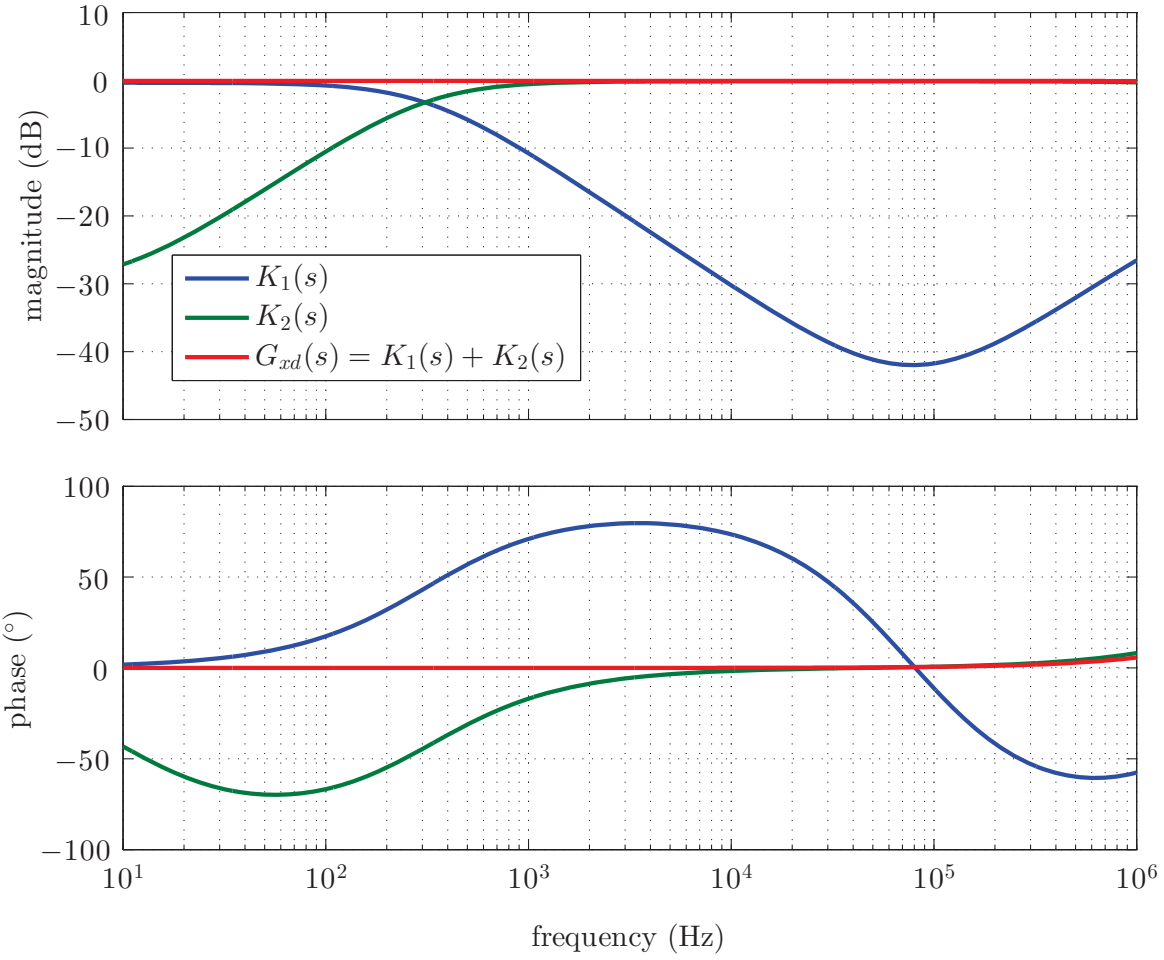


Figure 3.14: Bode diagram of H_2-optimal filters

3.4.4 H_2/H_∞-optimal filtering with compensation for sensor dynamics

Until now, the unequal sensitivities and dissimilar transfer functions of the used sensors claimed for inverse filters as discussed in Section 3.1. It is very convenient if H_∞ or H_2 designed optimal filters can also compensate for sensor dynamics modeled by the transfer functions $S_1(s)$ and $S_2(s)$ besides satisfying the First and Second criterion on robust filters.

The aim of this section is to find filters $K_1(s)$ and $K_2(s)$ so that the interconnection as shown in Figure 3.15 makes transfer function

$$G_{xd}(s) = S_1(s)K_1(s) + S_2(s)K_2(s) \approx 1 \tag{3.46}$$

over a wide range of frequencies with a phase close to 0° similar to (3.40) as before.

The two sensor models arises from first order approximation of the GMR and strain gauge sensors to

$$S_1(s) = \frac{1.592 \times 10^{-7}s + 0.1}{1.592 \times 10^{-4}s + 1} \tag{3.47a}$$

$$S_2(s) = \frac{8.754 \times 10^{-9}s + 0.55}{1.592 \times 10^{-5}s + 1} \tag{3.47b}$$

as shown in Figure 3.17. The make the compensation of sensor dynamics possible, $S_1(s)$ and $S_2(s)$ are included in the design, extending the model structure from as presented in Figure 3.11 to the model as can be seen in Figure 3.16. The noise models $N_1(s)$ and $N_2(s)$ are still given by (3.35a) and (3.35b) and the weight function $T_W(s)$ by (3.41), respectively.

The resulting H_2-optimal filters $K_1(s)$ and $K_2(s)$ are of fourth order in comparison to the third order filters as in the section before. They include a term for compensation of the sensor dynamics and further one which is a noise dependent filtering, however they can not be separated as before (Figure 3.10). The filters are given in continuous-time state-space representation as in (3.42) with matrices

$$\begin{aligned}
\mathbf{A}_F &= \begin{bmatrix} -1.18 \times 10^8 & -1.78 \times 10^8 & -6.58 \times 10^6 & 1.97 \times 10^5 \\ 2.54 \times 10^6 & -5.38 \times 10^7 & -6.43 \times 10^6 & 5.69 \times 10^4 \\ -8.85 \times 10^7 & -2.9 \times 10^8 & -3.47 \times 10^7 & 3.18 \times 10^5 \\ -1.74 \times 10 & -5.55 \times 10 & -6.65 \times 10^9 & 6.08 \times 10^7 \end{bmatrix} \\
\mathbf{c}_F^T &= \begin{bmatrix} -9.29 \times 10^4 & 1.01 \times 10^4 & 188 & -2.73 \times 10^{-5} \end{bmatrix} \\
\mathbf{b}_{FS1} &= \begin{bmatrix} -8.44 \times 10^4 \\ 1.76 \times 10^5 \\ 9.36 \times 10^5 \\ 1.8 \times 10^8 \end{bmatrix}, \mathbf{b}_{FS2} = \begin{bmatrix} -1.06 \times 10^6 \\ -3.06 \times 10^5 \\ -1.71 \times 10^6 \\ -3.27 \times 10^8 \end{bmatrix}
\end{aligned} \tag{3.48}$$

It can be seen, that all four poles are in the LHP located at

$$p_{H2F} = eig(\mathbf{A_F}) = \{-6.28 \times 10^7, -6.28 \times 10^7, -1.99 \times 10^7, -1.99 \times 10^3\}$$

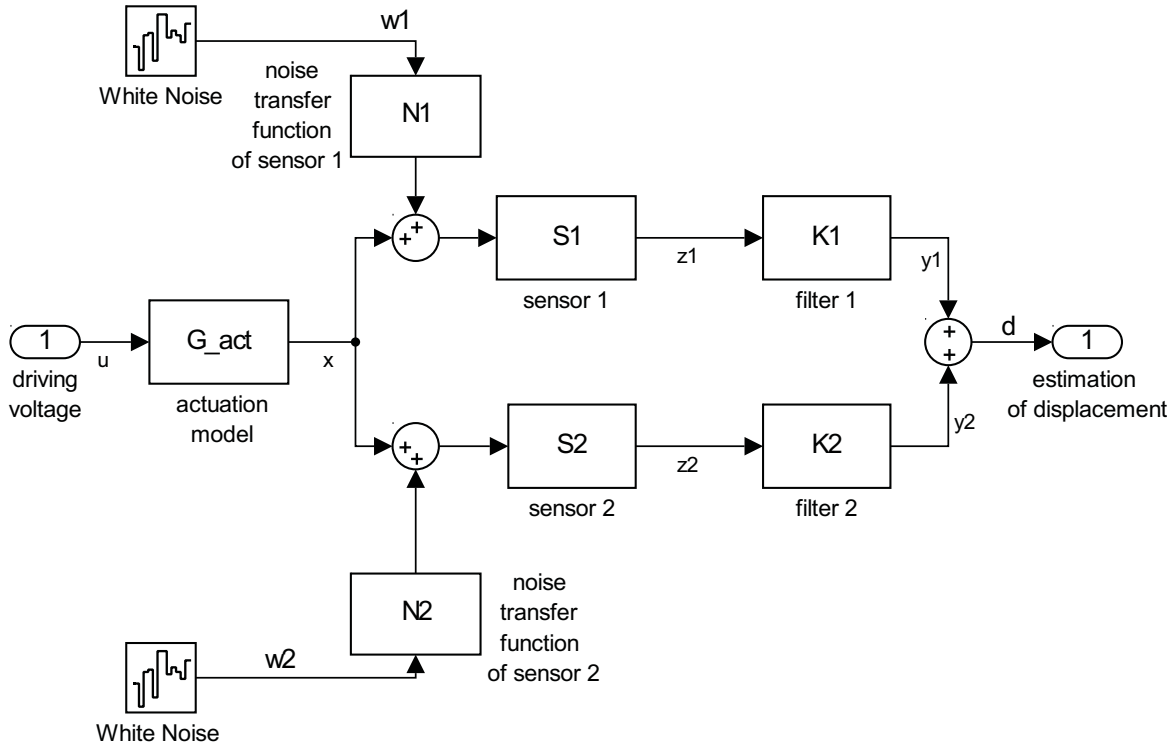


Figure 3.15: Concept of robust filtering with compensation for sensor dynamics. H_2/H_∞-optimal filters $K_1(s)$ and $K_2(s)$ are filtering the sensor signals in order to make transfer function $G_{xd}(s) \approx 1$.

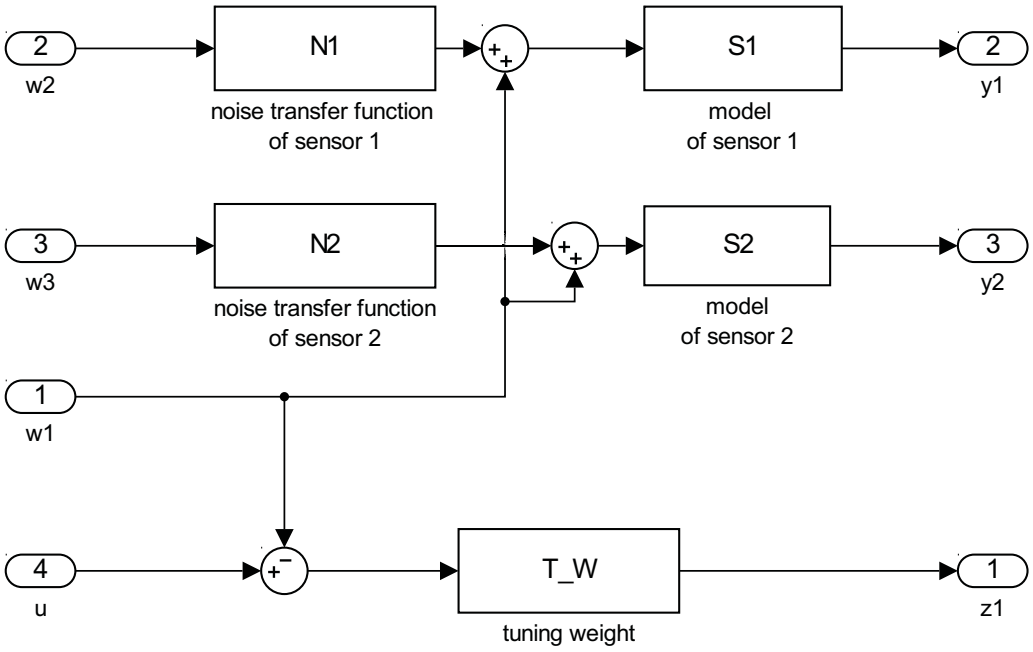


Figure 3.16: Model structure in SIMULINK for obtaining the generalized plant, augmented system.

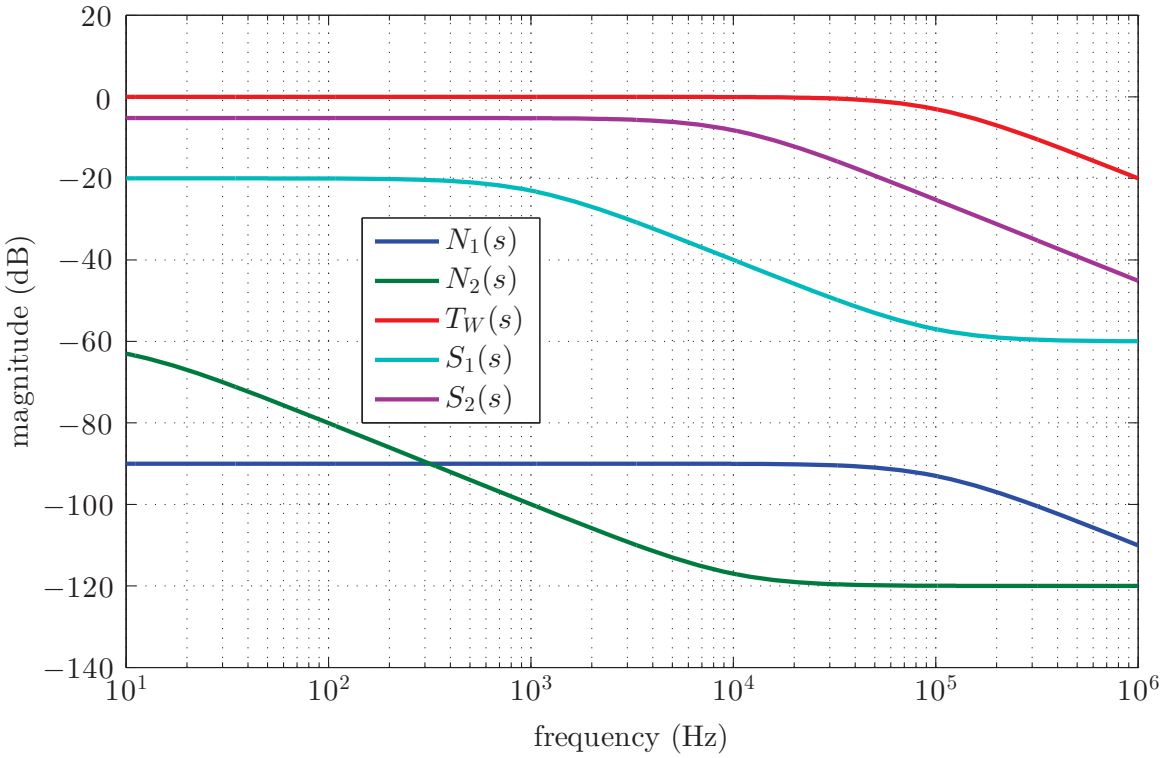


Figure 3.17: Noise, sensor and weight transfer functions

and in consequence the filters have a spectral radius of $\rho_{H2F} \approx 6 \times 10^7$. The filters are BIBO stable, as it can be also seen out of the step response in Figure 3.18. The Bode diagram of the filters $K_1(s)$ and $K_2(s)$ (3.48) are shown in Figure 3.19 as well as of the filtered sensors in Figure 3.20. In the last one the transfer function of the *summed filtered signals* $G_{xd}(s)$ is also shown which satisfies the criterion $\Delta G_{xd} \ll 1\,\mathrm{dB}$ with $G_{xd}(s)$ of (3.46) in the frequency range $[10\,\mathrm{Hz}, 1\,\mathrm{MHz}]$, at least in the numerical simulation. The difficulties of fast modes are discussed in Section 3.4.5.

To minimize H_∞-norm of the transfer matrix $\mathbf{G_{wz}}(s)$ of the generalized plant in Figure 3.12 there are two appoaches as already introduced. With identical assumptions as for the H_2 synthesis the exact method (a) solving the Riccati equations analytically yields a $\gamma = 7.8 \times 10^{-3}$, while the LMI-method (b) shows even a bit higher performance in terms of $\gamma = 2 \times 10^{-3}$. The exact-method (a) places the poles of the filters identical to H_2-optimal filters (3.48), i.e. to

$$p_{H\infty Fa} = p_{H2F} = \{-6.28 \times 10^7, -6.28 \times 10^7, -1.99 \times 10^7, -1.99 \times 10^3\}$$

while the LMI-approach solved H_∞-filters have poles at

$$p_{H\infty Fb} = \{-6.28 \times 10^7, -9.153 \times 10^6, -3.46 \times 10^5, -733\}$$

The largest poles are identical, however all the others are not even close to each other. The filters for (a) and (b) are remarkably different as can be seen by comparing the Bode diagrams in Figure 3.21a and 3.21b. The poles obtained by the LMI method

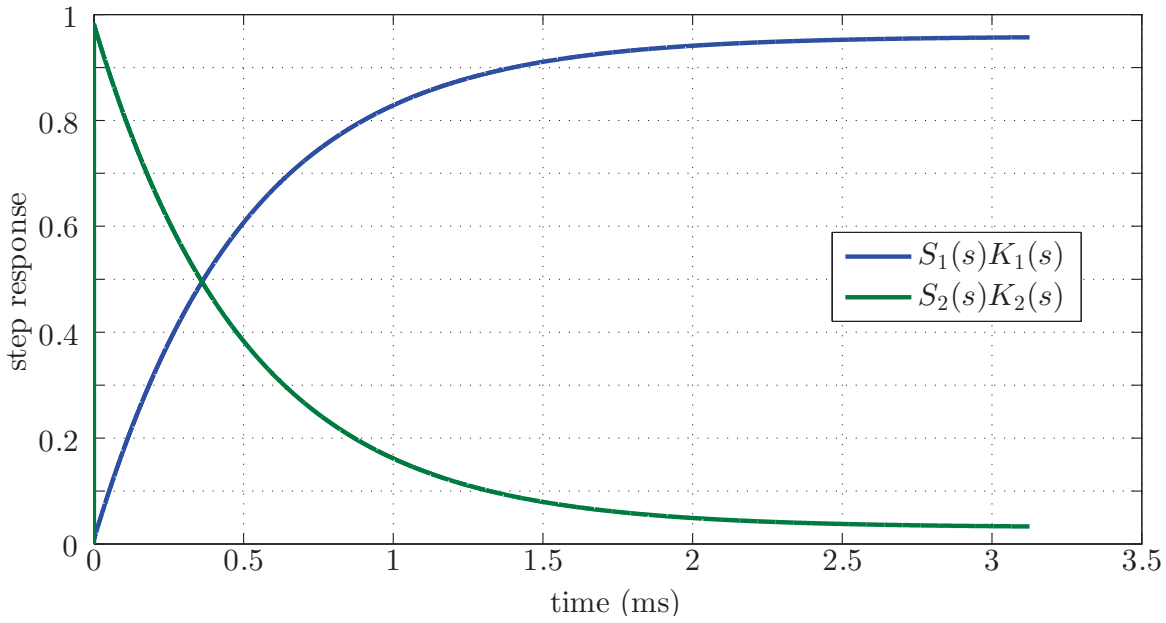


Figure 3.18: Step response of H_2-optimal filtered sensors.

$p_{H\infty Fb}$ do not even strive the poles $p_{H\infty Fa}$ even when the abortion criterion for the iterative optimization is made more strict.

Furthermore, it can be observed, that both H_2-optimal and H_∞-optimal filters obtained by solving the Riccati equations analytically (a), shows at 300 Hz (where the noise level of the sensors are equal) also an intersection of the filters' transfer function. The LMI-solved H_∞-filters (b) do not show this kind of behavior. To investigate the effectiveness of filtering, the H_∞ obtained filters are implemented on the multisensor system of Section 2.1 in followings.

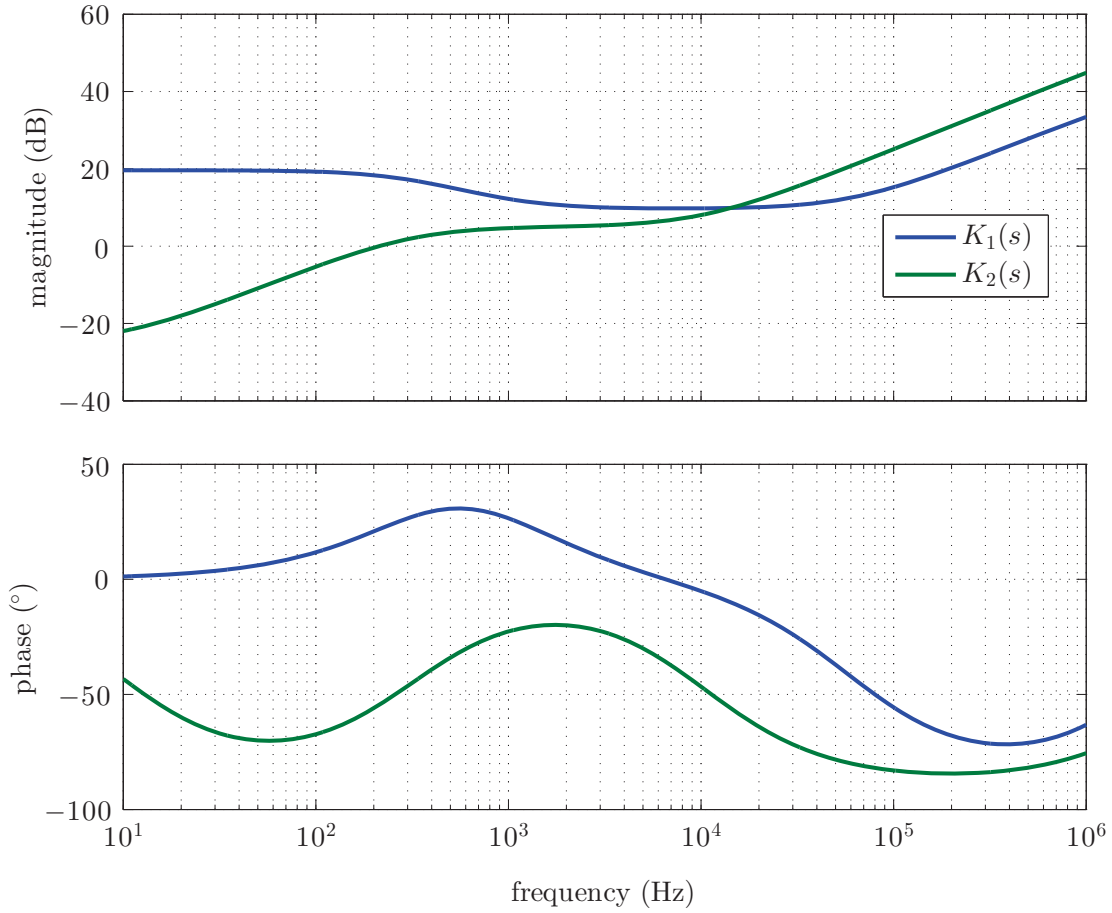


Figure 3.19: Bode diagram of H_2-optimal filters

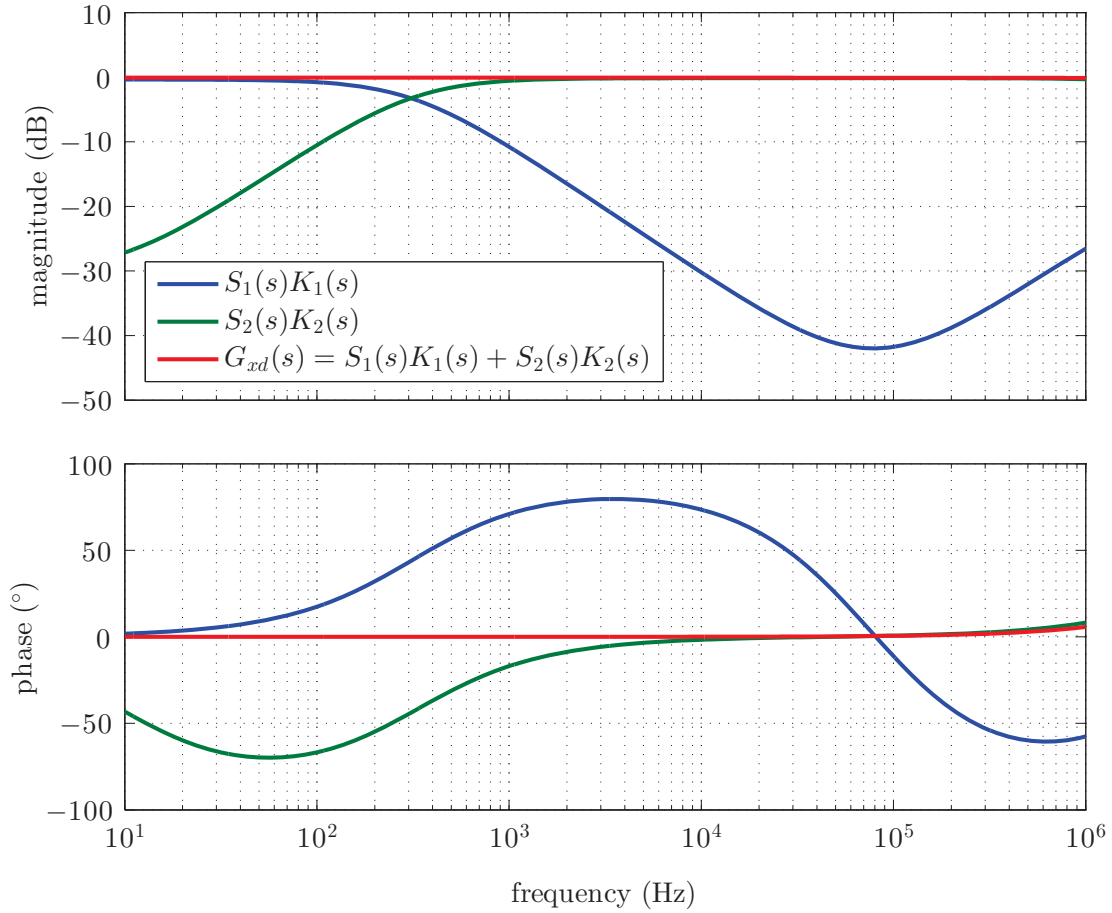


Figure 3.20: Bode diagram of H_2-optimal filtered sensors

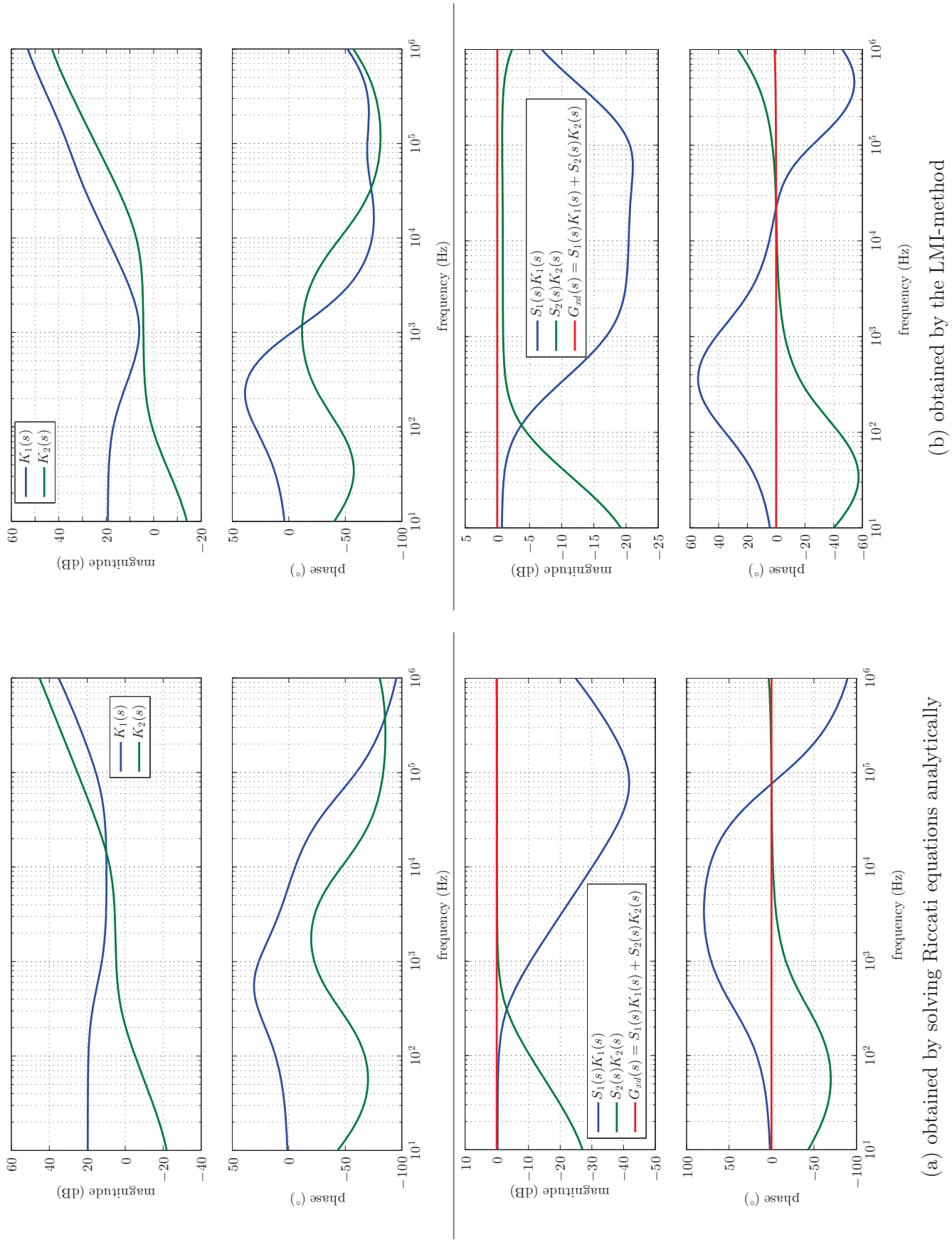


Figure 3.21: H_∞-optimal filters and filtered sensors

3.4.5 Verification

To acquire dynamic properties of the sensors, an actuator and an amplifier (*actuation*) are indispensable. The components of the test setup (as introduced partly in Figure 2.1) can be seen on a photo in Figure 3.22 which shows the arrangement of the components used for the verification.

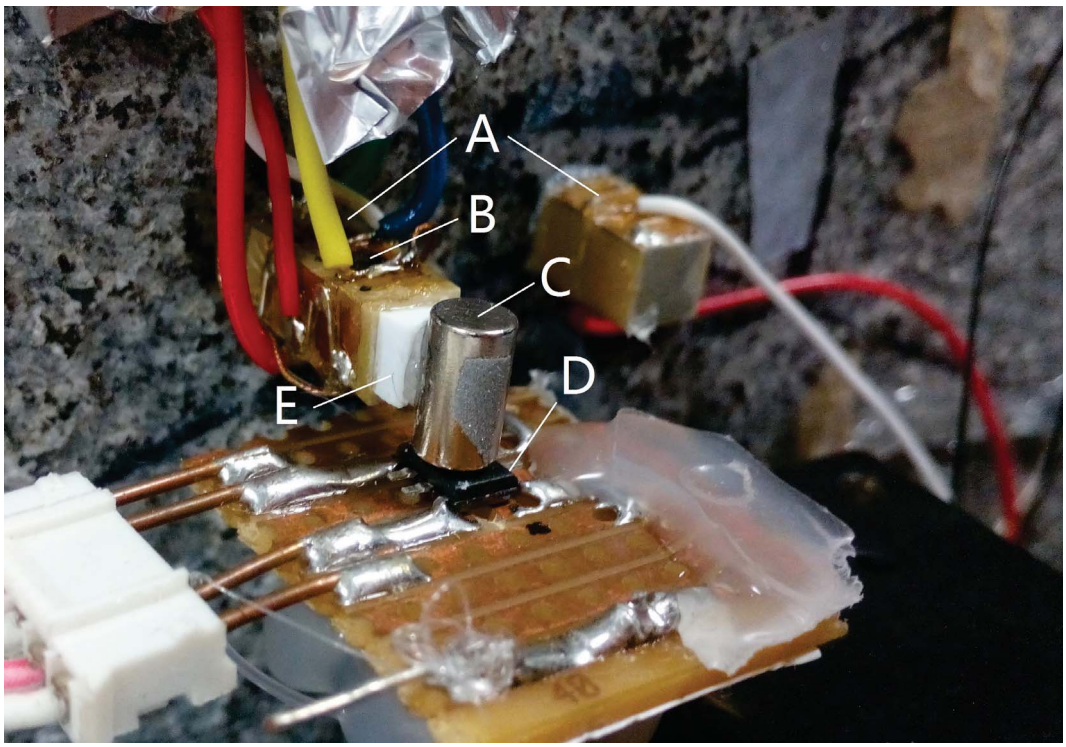


Figure 3.22: Photo of the setup: piezoelectric actuators (A), spacer (E), actuated permanent magnet (C), strain gauge sensors (B) and GMR sensor (D) are fixed to a granite stone.

Stacked piezoelectric actuators (A) are glued to a balance made of granite with a mass of 71 kg. It is chosen to be as large and heavy to provide sufficient damping of excitation forces initiated by the piezoelectric actuator especially at high frequencies. The permanent magnet (C) with a weight of 0.47 g is attached through the polyethylene spacer (E) with cross section of ca. 16 mm^2, length of ca. 2 mm, Young's modulus 0.7 GPa [25] and therefore a stiffness of ca. 5.6 kN/m on top of the actuator (A). Despite the compact design, the finite stiffness of the spacer and the not negligible mass of the permanent magnet creates additional resonance at about 550 Hz so that elements A+E+C can not be modeled as one stiff body[14]. The vibrometer measures the elongation of the piezo actuator in the middle of the concave front of (C).

Figure 3.23 shows the measured frequency response of the system including the actuation and the sensors. Since the transfer function of the actuation has a −20 dB/decade

[14] In consequence, the strain gauge sensors (B) and the GMR sensor (D) are out of phase and they are measuring some projection of the piezo's true elongation. Transversal resonance and other unmodeled effects cause remarkable non-linearities and higher modes

slope and also some parasitic dynamics as shown in Figure 2.6, it can not be neglected if the sensors' frequency response have to be acquired. Therefore, a relative measurement using the network analyzer HP4395 with the vibrometer as a reference sensor, makes it possible to acquire only the transfer function of the sensors. It is as if the actuation would be corrected for a constant amplitude of excitation over a wide frequency range. It is theoretically equivalent to the method, if the frequency response of the actuation would be recorded separately and then substracted from the acquired system's response (actuation and sensors) as well in magnitude as in phase. This is verified and provides approximately the same results until a certain SNR is violated.

The sensors' transfer functions and the fitted models can be seen in Figure 3.24. The model fitting is supported by the System Identification Toolbox of Matlab and state-space models of 8^{th} order are chosen to fit to the measurement properly (in terms of mean square estimates (MSE) of 0.88 and 0.91 for the GMR and for strain gauges, respectively). They are given by the differential equations

$$S_1 \stackrel{s}{=} \begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}_1\mathbf{x}(t) + \mathbf{b}_1 u(t) \\ y(t) = \mathbf{c}_1^T\mathbf{x}(t) \end{cases} \tag{3.49a}$$

$$S_2 \stackrel{s}{=} \begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}_2\mathbf{x}(t) + \mathbf{b}_2 u(t) \\ y(t) = \mathbf{c}_2^T\mathbf{x}(t) \end{cases} \tag{3.49b}$$

where $S_1(s)$ is the model of the GMR sensor and $S_2(s)$ that of the strain gauge sensor, respectively. The matrices are

$$\mathbf{A}_1 = \begin{bmatrix} -3.51\times10^4 & 1.76\times10^5 & -4.36\times10^4 & 1.19\times10^4 & -8.92\times10^4 & 2.32\times10^4 & 7.87\times10^4 & -1.32\times10^5 \\ -1.15\times10^5 & -2.3\times10^4 & 1.71\times10^5 & -1.29\times10^4 & 7.38\times10^4 & -1.22\times10^4 & -9.27\times10^4 & 9.42\times10^4 \\ -1.62\times10^4 & -1.13\times10^5 & -2.11\times10^4 & 4.82\times10^4 & -7.64\times10^4 & 2.7\times10^4 & 4.06\times10^4 & -1.4\times10^5 \\ 621 & 7.05\times10^3 & -3.16\times10^4 & 897 & 3.7\times10^4 & -3.09\times10^3 & -3.48\times10^4 & 4.78\times10^3 \\ 1.48\times10^4 & 6.43\times10^3 & 3.25\times10^4 & -1.06\times10^4 & -1.83\times10^4 & 2.61\times10^4 & 3.41\times10^4 & -2.21\times10^5 \\ -3.03\times10^3 & -2.95\times10^3 & -6.94\times10^3 & 1.56\times10^3 & 2.75\times10^3 & -2.6\times10^3 & -1.4\times10^5 & 3.89\times10^4 \\ -456 & 3.07\times10^3 & -1.27\times10^3 & 2.38\times10^3 & -1.2\times10^4 & 1.08\times10^5 & -761 & 2.83\times10^5 \\ -1.3\times10^3 & -1.82\times10^3 & -895 & -301 & 1.27\times10^4 & -7.99\times10^3 & -4.51\times10^4 & -9.34\times10^4 \end{bmatrix}$$

$$\mathbf{b}_1 = \begin{bmatrix} -262 & -619 & -516 & -69.4 & 247 & -44.3 & 71.8 & -28.1 \end{bmatrix}^T$$

$$\mathbf{c}_1^T = \begin{bmatrix} -267 & 239 & -215 & 55.5 & -257 & 81.5 & 260 & -548 \end{bmatrix}$$

$$\mathbf{A}_2 = \begin{bmatrix} -1.04\times10^4 & 9.86\times10^4 & 2.37\times10^4 & 2.16\times10^4 & 3.15\times10^4 & -2.4\times10^4 & -2.3\times10^4 & 2.95\times10^4 \\ -7.04\times10^4 & -1.83\times10^4 & -4.93\times10^4 & -3.88\times10^4 & -5.36\times10^4 & 2.5\times10^4 & 3.07\times10^4 & -4.28\times10^4 \\ 2.54\times10^3 & 8.14\times10^3 & -7.53\times10^3 & -1.62\times10^5 & -8.13\times10^4 & 5.75\times10^4 & 4.82\times10^4 & -3.8\times10^4 \\ 2.97\times10^3 & 8.39\times10^3 & 1.22\times10^5 & -1.23\times10^4 & -1.8\times10^4 & 6.15\times10^4 & 1.52\times10^4 & -2.72\times10^4 \\ 3.77\times10^3 & 9.44\times10^3 & 2.55\times10^4 & -1.82\times10^4 & -2.65\times10^4 & 1.27\times10^5 & 2.2\times10^4 & -3.88\times10^4 \\ 2.48\times10^3 & 3.48\times10^3 & -2.77\times10^4 & -3.78\times10^4 & -9.26\times10^4 & -1.12\times10^4 & -1.14\times10^5 & 4.79\times10^4 \\ -1.19\times10^3 & -2.84\times10^3 & -1.97\times10^3 & 5.8\times10^3 & 9.66\times10^3 & 9.18\times10^4 & -7.53\times10^3 & 2.25\times10^4 \\ 1.14\times10^3 & 2.03\times10^3 & -3.65\times10^3 & -7.23\times10^3 & -1.1\times10^4 & -1.61\times10^4 & 1\times10^4 & -2.27\times10^4 \end{bmatrix}$$

$$\mathbf{b}_2 = \begin{bmatrix} -3.1 & -469 & 215 & -2.96 & 70.9 & 90.8 & -88.1 & 60 \end{bmatrix}^T$$

$$\mathbf{c}_2^T = \begin{bmatrix} -144 & 194 & 188 & 167 & 241 & -149 & -163 & 212 \end{bmatrix}$$

The noise PSDs of the sensors are recorded in the analogue domain using the network analyzer HP4395 resulting in the measurement shown in Figure 3.25. They show that strain gauges have in average a lower noise level for the setup with parameters as in Table 2.3, amplifier gain 33.5 dB of strain gauges and 40 dB for the GMR sensor which are chosen to ensure approximately the same sensitivity for displacement.

The signals are also recorded using the simultaneous sampling data acquisition device U2542A[15] including synchronized sampling 16 bit ADCs. The histograms of the detrended (mean free) voltages for a 10^4-point sample, sampled at 200 kHz (in total 50 ms) are visible in Figure 3.26. It is expected that the sensor noise is normally distributed. The reason why in case of stain gauges the least squares fit of the distribution differs from the Gaussian bell is not fully investigated. It is assumed to arise from disturbance voltage, because the uniformly distributed quantization noise of the 16 bit ADC with ± 1.25 V range is determined by the LSB whic are ca. 38 μV.

To estimate the PSDs, the spectrum is averaged out of 40 spectra computed from 10^3 samples each, sampled uniformly at 200 kHz. The averaged PSDs are shown in Figure 3.26. They let us recognize roughly the same, strain gauges have less noise over all frequencies. Furthermore, it can be seen that some dominant characteristics of the transfer functions in Figure 3.24 can also be found in the PSDs. Especially, for strain gauges around 13 kHz some increase of the intrinsic background noise as well as a peak in the transfer function at that frequency can be observed. It is assumed that the *resonance* is excited even by white noise.

The noise power $p(\mathscr{F})$, the noise voltage $v(\mathscr{F})$, and hence, the resolution of the sensors $r(\mathscr{F})$ (RMS) in the corresponding bandwidth $\mathscr{F} = [1\,\text{kHz}, 100\,\text{kHz}]$ are determined to

sensor	S	$p(\mathscr{F})$	$v(\mathscr{F})$	$r(\mathscr{F})$
GMR	59.5 V/m	6.20 μW	2.49 mV	41.85 μm
SG	86.9 V/m	3.87 μW	1.97 mV	22.67 μm

Table 3.2: Noise and RMS-resolution of sensors before filtering.

The PSDs of the sensors are approximated with first and second order linear models, found by pole placement, and thus, given by

$$N_1(s) = \frac{100(s+10^3)}{(s+100)(s+10^6)} \tag{3.50a}$$

$$N_2(s) = \frac{300}{(s+10^7)} \tag{3.50b}$$

[15] 16 bits, 500 kSPS at maximum, Agilent Technologies, Santa Clara, CA, USA.

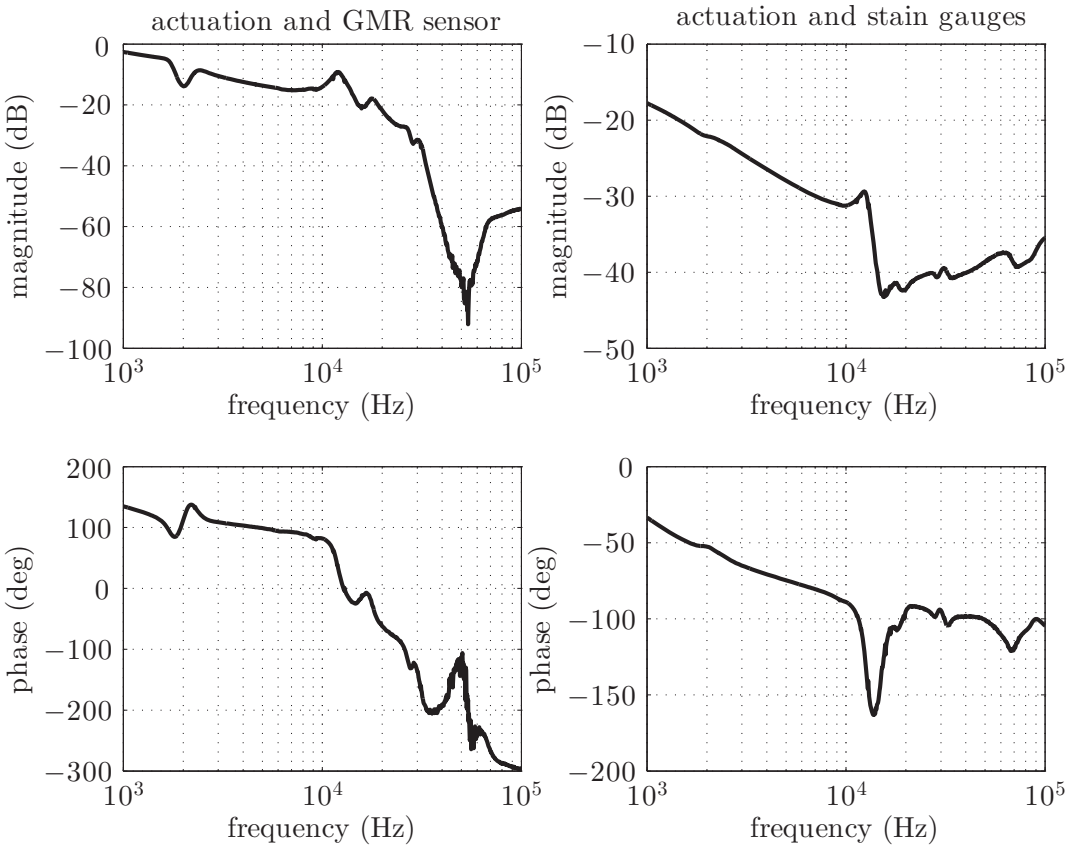


Figure 3.23: Bode plot of sensors including actuation dynamics

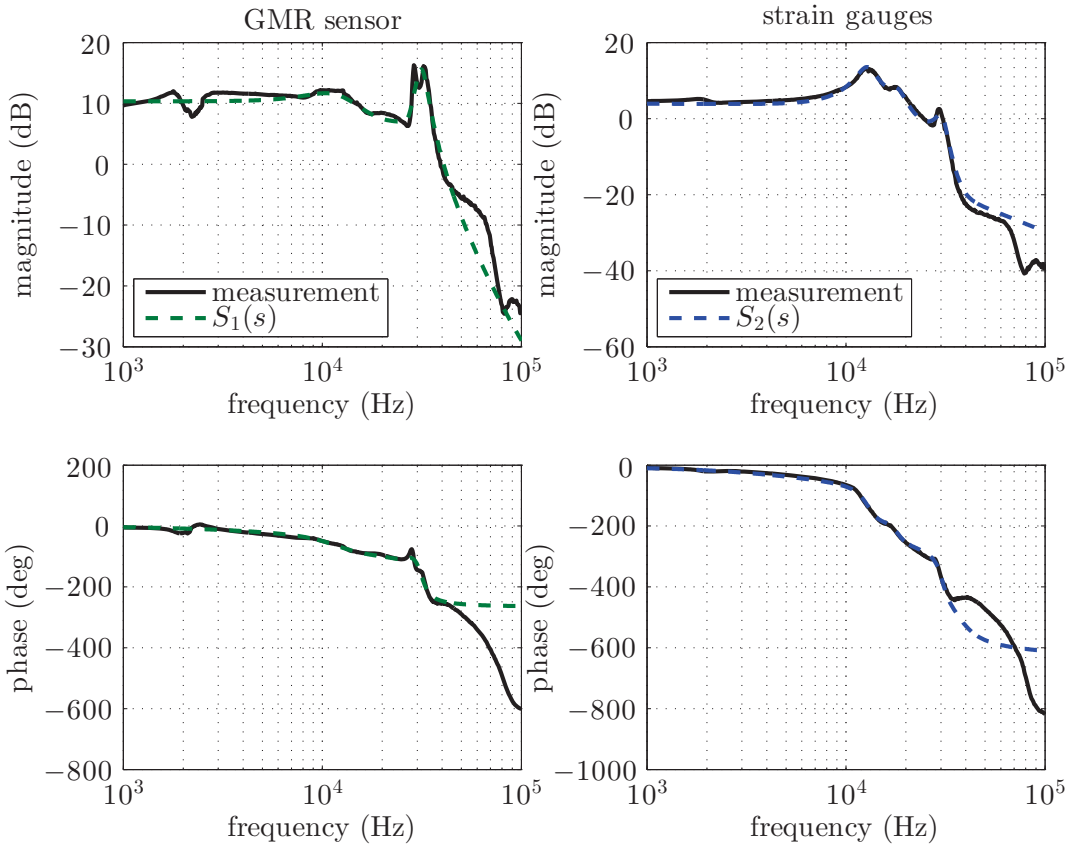


Figure 3.24: Bode plot of sensors with continuous-time 8^{th} order state-space models $S_1(s)$ and $S_2(s)$.

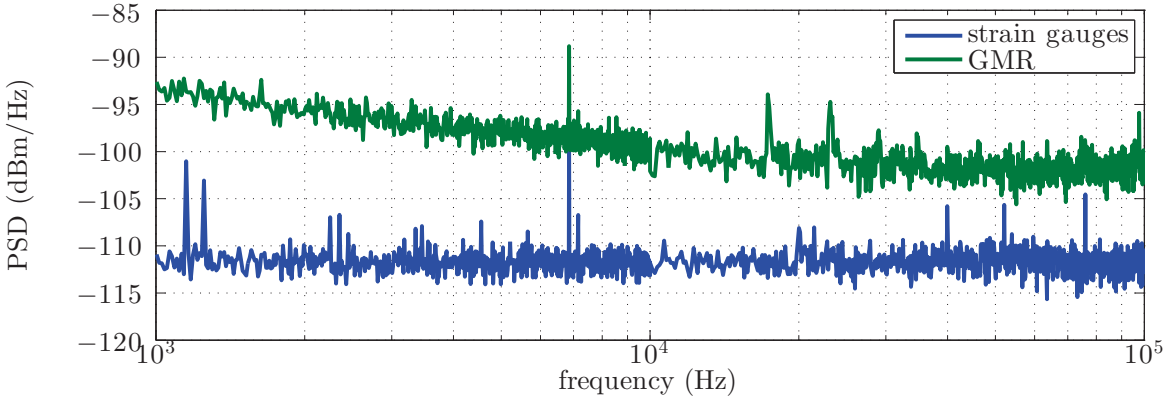


Figure 3.25: PSDs of sensors recorded with spectrum analyzer HP4395 at RBW=1 Hz, VBW=100 mHz.

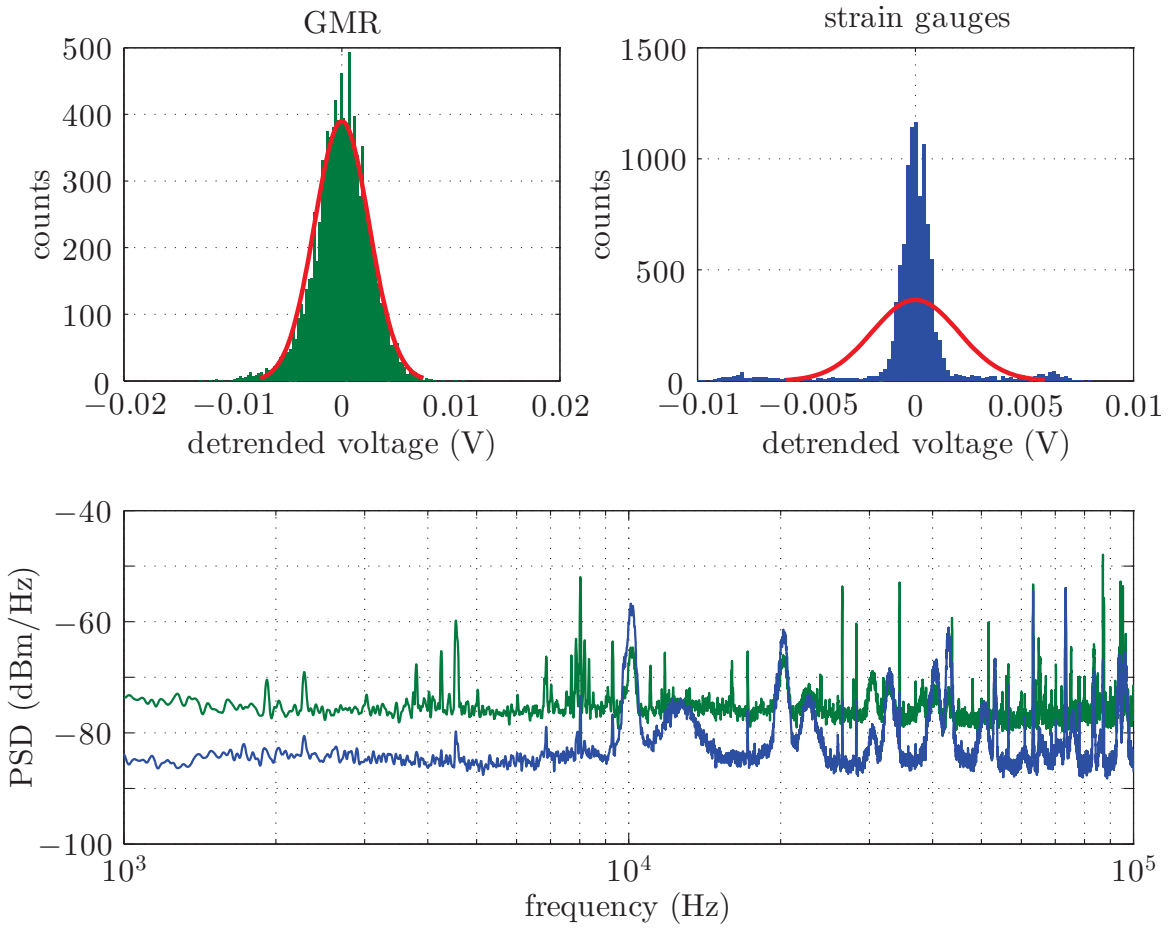


Figure 3.26: Histograms of sensor signals, and PSD estimates computed of the hanning windowed sensor signals of length 10^3 points, sampled at 200 kHz, spectrum averaged out of 40 estimates.

H_∞-optimal filter design

A balanced, minimal-phase, stable realization of all input models (sensor models and noise models) and thus, of the augmented system described by the matrix $\mathbf{P}$ according to (3.38), is a necessary, but not a sufficient condition for the Riccati-solved H_∞ synthesis (exact-method) [18]. For the LMI-solved H_∞ problem it is already a sufficient condition for practical systems [37].

H_∞ filter design yields filters of 20^{th} order for the presented input models. The distribution of the state energy is expressed by the Hankel values, plotted in Figure 3.27. Due to the logarithmic display of the energy it can be assumed that at least 7 states (labelled with 14 to 20) can be neglected, the performance of filtering will not degrade remarkably.

The model order reduction is preferred over slow and fast mode decomposition. After eliminating 10 states of $K_1(s)$ and 9 states of $K_2(s)$ no major difference on the Bode diagram of the filters can be observed. The resulting filters $K_{R1}(s)$ and $K_{R2}(s)$ are shown in Figure 3.28.

In order to implement the filtering on a processing system (PS) with finite sampling time, in the next step the discretization of the filters is performed. The bilinear transform by Tustin is one of the most faithful approximations of the continuous-time filters in the frequency domain, much better than a Zero-Order-Hold. Therefore, filters are discretized using the Tustin transform, computed by MATLAB command `c2d(·,·,'tustin')` and sampling frequency $f_s = 500\,\text{kHz}$, since this is the maximum of the DAQ unit. The resulting discrete-time filters $K_{1d}(z)$ and $K_{2d}(z)$ are also shown in Figure 3.28.

Lastly, the discretized filters $K_{1d}(z)$ and $K_{2d}(z)$ are applied on the sensor signals and the performance of filtering is verified by measurements. The filtering takes place in the time domain, computed by MATLAB command `lsim()`. However to compute a Bode diagram as well the sensor signals as the filtered sensor signals have to be transformed to the frequency domain. The discrete Fourier transform (DFT) (1.29) is used and the Bode diagram is computed sequentially (offline) given by

$$T(i\omega) = \frac{\mathcal{F}(\text{hann}(y_k))}{\mathcal{F}(\text{hann}(x_k))} \tag{3.51}$$

where $\mathcal{F}$ denotes the m-point discrete Fourier transform of the hanning windowed time signals y_k of output (filters) and x_k of input (sensors) at discrete time steps $k = 1..m$ implemented by the $2^{(\log_2(m)+2)}$-point fast Fourier transform (FFT) algorithm for the zero padded windowed time signals. The phase $\angle T(i\omega)$ is defined by (1.27) and it is computed by the four-quadrant inverse tangent realized by the MATLAB commands `atan2()` or `angle()`.

The measurements, presented in Figure 3.29, show the sequentially acquired Bode diagrams of the filtered sensors as well as the total transfer function

$$T_{sum}(z) = S_1(z)K_{1d}(z) + S_2(z)K_{2d}(z) \tag{3.52}$$

acquired in the frequency range $\mathscr{F}_1 = [500\,\text{Hz}, 250\,\text{kHz}]$ out of 150 logarithmically spaced discrete steps of frequencies by recording each time 10^5 samples, uniformly

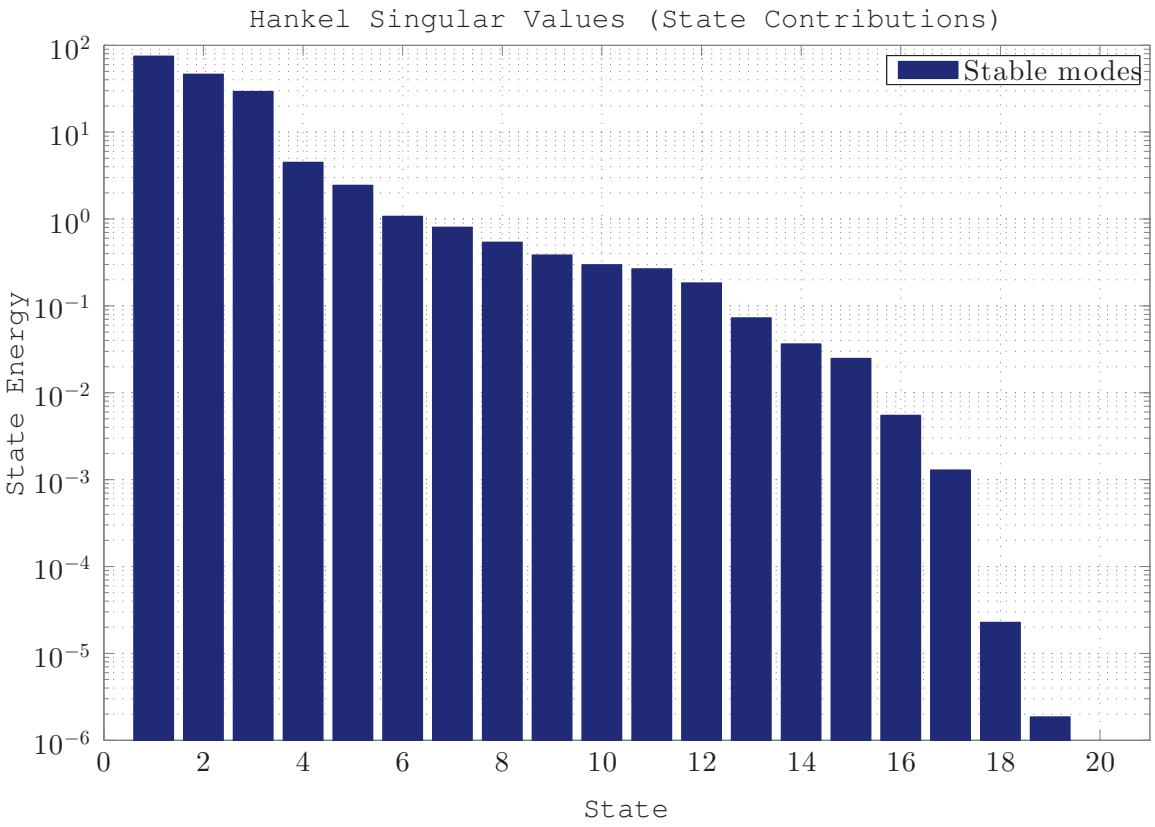


Figure 3.27: Hankel values of the analytically solved H_∞-optimal filters.

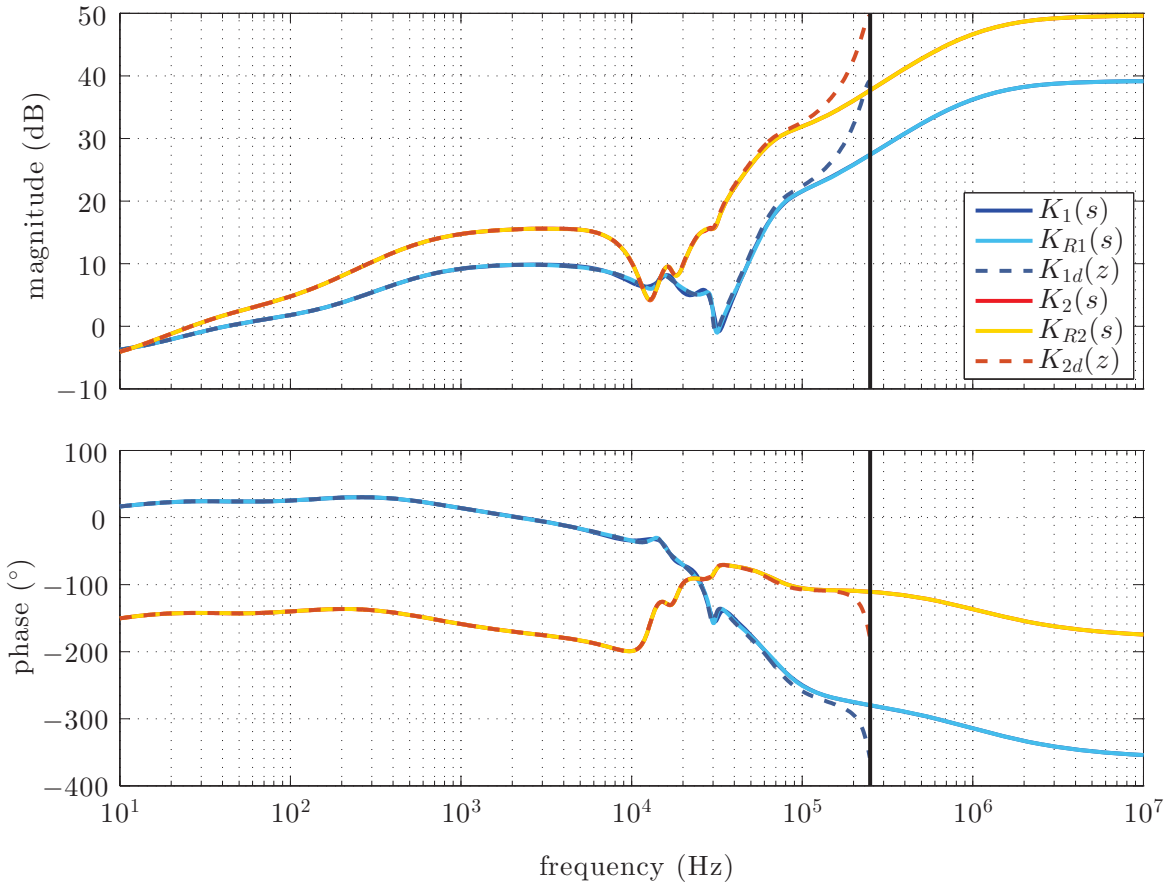


Figure 3.28: Time-continuous filters, reduced order filters and discretized filters at sampling frequency of 500 kHz.

sampled at 500 kHz. It can be seen, that the variability of the summed transfer function is $\Delta T_{sum}(\mathscr{F}_2) < 14\,\text{dB}$ for $\mathscr{F}_2 = [500\,\text{Hz}, 100\,\text{kHz}]$ compared to the variability of the sensors with at least 50 dB in $\mathscr{F}_2$ as in Figure 3.24. It should be noted that the *effectiveness* is not as perfect as in the simulation in Figure 3.20. The deviation, despite to the good fitted sensor models, is assumed to arise from the temporal or thermal change of the test setup, as a position dependent sensitivity of the magnetic sensor.

Evaluation

It can be shown that the H_∞-optimal filtering of the two sensors each having excessive peaks in the transfer functions up to 30 dB are mostly compensated (up to 6 dB remains) and, thus the bandwidth of control is also enlarged.

According to the computed PSDs in Figure 3.30 the resolution of the fused sensor signal seems to be even worse than that of a single sensor. This is only a pretense, as the filters $K_{1d}(z)$ and $K_{2d}(z)$ have an amplification $> 0\,\text{dB}$ for all frequencies above $\approx 100\,\text{Hz}$, as the filters are designed for[16]. Hence, the noise gets amplified, too.

Instead of evaluating filter performance out of noise densities the Signal-to-Noise ratio (SNR) is computed. It is defined by the relationship between power quantities

$$SNR = 10 \log \left(\frac{p_{signal}}{p_{noise}} \right) \tag{3.53}$$

in units of 1 dB where p_{signal} is the power of the sine wave delivered by the sensor (squared RMS voltage) and p_{noise} is the average power of noise in the specified band. Figure 3.31 clearly shows that the SNR of the filtered sensor signals is at almost all frequencies below 30 kHz greater, and hence better than that of a single sensor. It allows to state that the resolution of the fused signal can be successfully improved by a factor of 2.51 (by 8 dB in terms of power) in average to 9.03 μm, compared to that of a single sensor having a resolution of 22.67 μm for the presented setup in the bandwidth of 30 kHz.

The presented method of sensor fusion, *prefers* always the sensor with lower noise (up to a certain complexity depending of the filter's order). For the presented combination of sensors (strain gauges and GMR) with no intersection of the PSDs the true power of H_∞ filtering is not as remarkable in terms of noise reduction as for the presented example in Section 3.4.4.

By evaluating the poles and zeros of the filters $K_{R1}(s)$ and $K_{R2}(s)$, it can be concluded that the thermal drift of the GMR sensor (sensor 1) is damped at least by 40 dB at 1 Hz, so almost no offset error of the GMR sensor distorts the position estimation.

A further example illustrates the case, if the actuation dynamics is also included in the modeling, and H_∞-filters compensate for that, too. The frequency response is

[16] It should be noted, that the frequency response in Figure 3.24 is recorded with the factor of 500 nm/V and the measurement in Figure 3.29 is done with the scale factor of 5 μm/V because of practical reasons, causing a shift of $-20\,\text{dB}$ of the transfer functions compared to the plot in Figure 3.29.

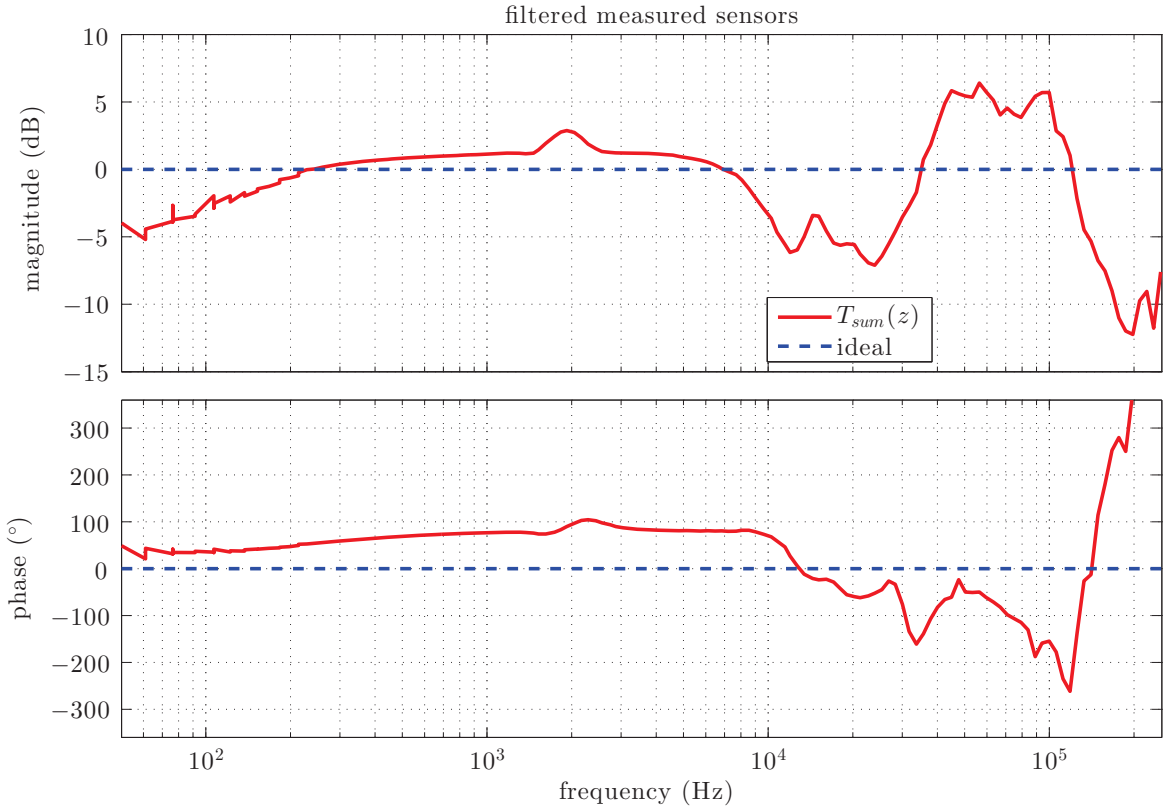


Figure 3.29: Result of H_∞-optimal filtering: Bode diagram of $T_{sum}(z)$, $\Delta T_{sum}([500\,\text{Hz}, 100\,\text{kHz}]) < 12\,\text{dB}$

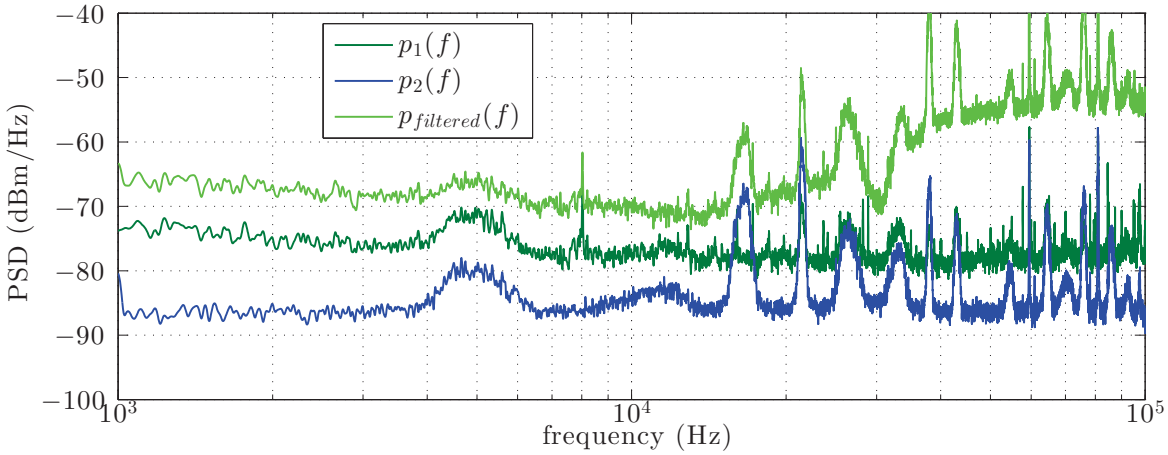


Figure 3.30: PSD of sensors 1, 2 and of the fusion.

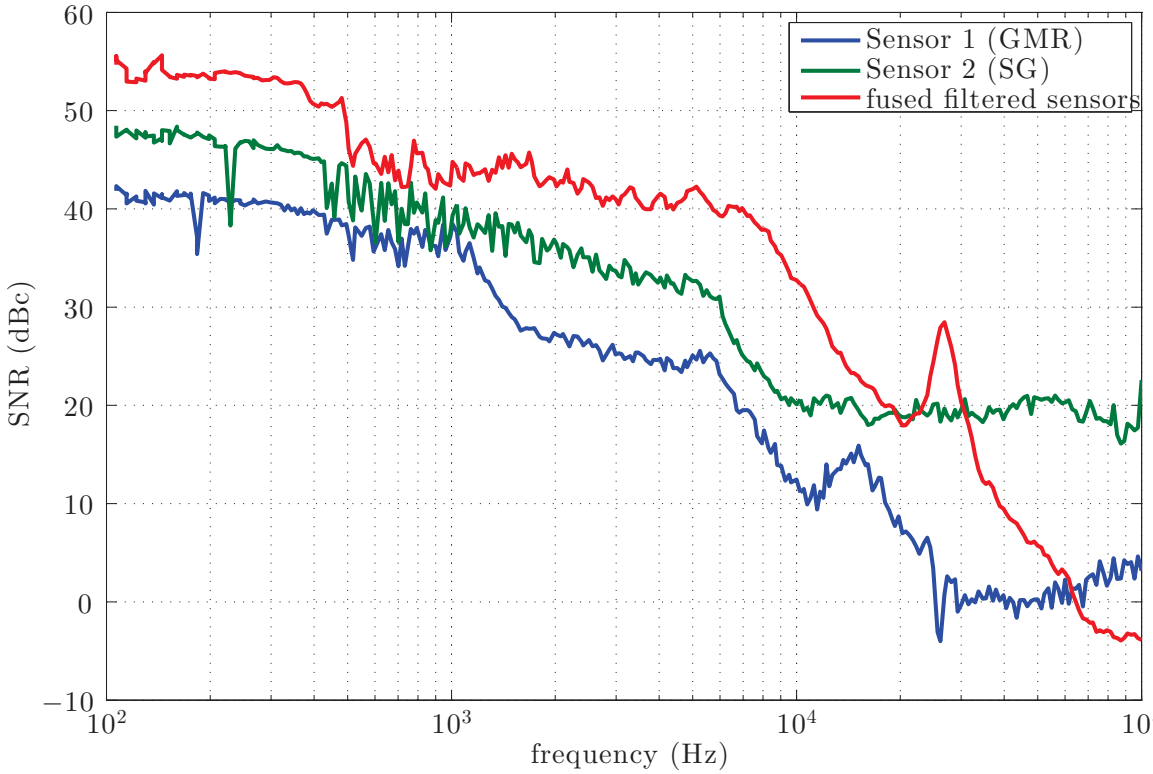


Figure 3.31: Signal-to-Noise ratio of sensors before and after filtering with H_∞-optimal filters (LMI).

presented in Figure 3.32. The resulting transfer function $sum(s)$ has in the frequency range of $\mathscr{F}_2 = [10\,\text{Hz}, 10\,\text{kHz}]$ a variability of $\Delta sum(\mathscr{F}_2) = 7\,\text{dB}$ and an unintended large phase lag possibly originated from modeling uncertainties.

The presented method of optimal filtering as discussed in Section 3.4 can be extended to more than two sensors. Preferably taking a capacitive, an optical and a magnetic sensor can provide a more reliable estimation of the true position as disturbances in the magnetic field (e.g. caused by mains, switched transformers, etc.) usually do not influence equally e.g. magnetic and capacitive sensors. It is useful to extend the presented method of optimal filtering to as much sensors as possible, as even an additional sensor with the same SNR as all the others increases theoretically the SNR of the fused signal by 3 dB. The theoretical enhancement of the SNR concerning fusion is summarized in Table 3.3. It assumes perfectly correlated signals and uncorrelated noises[17].

[17] The fourth column can be interpreted as the addition of two sensor signals with the properties owning that of Sensor A.

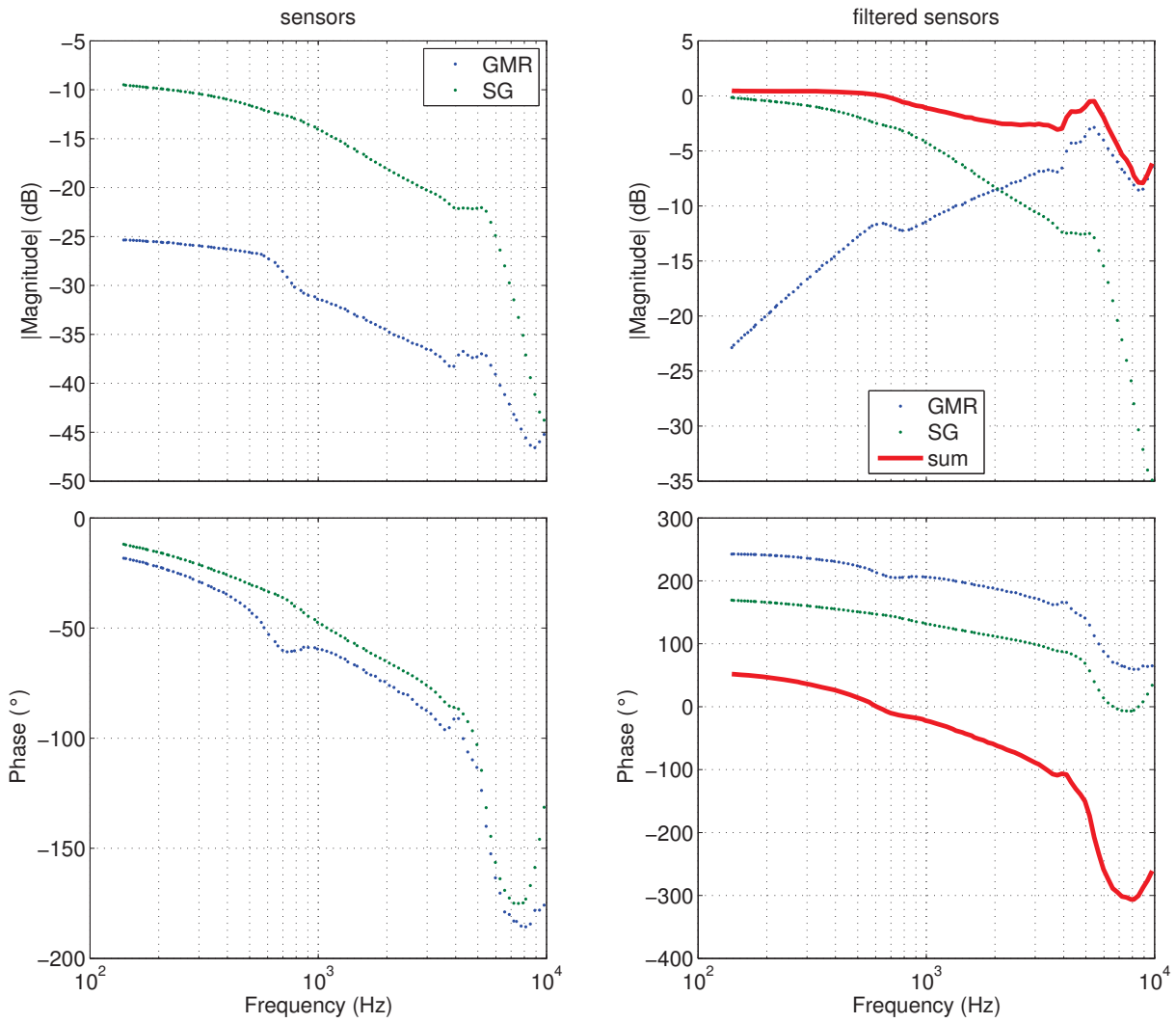


Figure 3.32: Bode plot of sensors together with actuation (left) and effect of noise-optimal H_∞-filters and resulting fusion (right)

	sensor A	sensor B	A+A (additive fusion)	A+B (additive fusion)
signal power p_s	v_A^2	v_B^2	$(2v_A)^2$	$(v_A+v_B)^2$
noise power p_n	p_A	p_B	$2p_A$	p_A+p_B
$\frac{p_s}{p_n}$	$\frac{v_A^2}{p_A}$	$\frac{v_B^2}{p_B}$	$\frac{2v_A^2}{p_A}$	$\frac{(v_A+v_B)^2}{p_A+p_B} := r$
$\mathrm{SNR} = 10\log\frac{p_s}{p_n}$	SNR_A	SNR_B	$\mathrm{SNR}_A + 3\,\mathrm{dB}$	$10\log(r)$

Table 3.3: SNR enhancement for two sensors with assumed perfectly correlated signals and uncorrelated noise.

CHAPTER 4

Summary and outlook

An attempt has been made to treat the problems and challenges in sensor filtering and sensor fusion throughout the entire circumference of the present thesis by a methodological approach. I was guided by the the principle of G. H. Hardy, namely

> *Sometimes one has to say difficult things, but one ought to say them as simply as one knows how.*

by preserving the scientific language of argumentation.

4.1 Summary

It is shown that by the fusion of two sensor signal's (GMR magnetic sensor AA002 and strain gauges FAET-A6347Q) with H_∞-optimal filters the thermal drift of the GMR sensor is practically eliminated. In addition the sensor's bandwidth is extended from 10 kHz (of the fast sensor) to 30 kHz. Furthermore, the resolution is improved by 151% from 22.67 μm to 9.03 μm for the 30 kHz measurement bandwidth.

Also the Kalman filter augmented with a noise model of a sensor provides an enhancement of resolution up to a factor of 6 by explicitly using the third order model of a sensor. Other profits of sensor fusion are the increase of disturbance rejection.

4.2 Contents

- Chapter 1 introduces linear systems and the corresponding properties and tools on the language of mathematics, which are indispensable in order to analyze, evaluate and verify linear sensor models. The chapter also motivates the Reader to consider to filter and fuse sensor signals in order to increase the accuracy of a position controlled systems and extend the bandwidth of control.

- Chapter 2 gives an overview of some of the available principles to measure displacement. It reports the results of the characterization of some of the available sensors of today's technology and their parameters derived out of the measurements, such as resolution, bandwidth, range, etc. In Section 2.1 the test setup is introduced, which is needed for characterizing sensors. Section 2.2 shows the data acquisition system based on a real time processor, which is used to evaluate filters under real conditions. Section 2.8 lists key parameters of the characterized sensors, to enable a quick comparison.

- Chapter 3 covers some basic principles to filter data in order to lower noise and also to fuse data of multiple sensors. Complementary filters in Section 3.2, various Kalman filters in Section 3.3 and noise-optimal filters in Section 3.4 are discussed, with their prerequisites, operation principles, advantages and disadvantages for a given application as well as implementation results to enable a quantification of performance improvement mainly in terms of resolution and bandwidth.

4.3 Outlook

Combining noise-optimal filters with a Kalman filter

The Kalman filter minimizes the mean square error of state estimates, while H_∞ filters the worst case error. For designing a Kalman filter, noise covariances need to be estimated, however for H_∞ filter synthesis they are not necessary, although PSD of sensor noise is required. Since noise variance and especially covariances between elements of the state vector (process noise) are more or less found by *tuning*[1] the method of optimal filtering is more straight forward from that aspect of view. For a general strategy of synthesizing noise-optimal filters, the LMI-method solved H_∞ synthesis should be preferred, since there are very few requirements on input models.

It should be noted, that the presented filtering in Section 3.4, titled *Optimal filters*, is a very basic and probably the most simple application of robust control without realizing any *control*. The focus is set to estimating the position as accurate as possible from an available set of sensor signals and linear models. The tools of filter design in the *Hardy space* enable the integration of a sort of Kalman filters right within the design, and offers much more possibilities compared to a filter design in the state-space [37]. Despite this, to preserve simplicity in design, a combination of model based filtering (Kalman filter) and noise-optimal filtering, as until now separately introduced and implemented, are proposed in followings.

In order to use benefits of minimal noise and advantages of model based filters, noise-optimal filters are combined with a linear MISO Kalman filter as shown in Figure 4.1. Filters $K\_S1(s)$ and $K\_S2(s)$ are designed according to Section 3.4.4 to compensate for unequal and uneven sensitivity of sensors (such as model inversion as in Section 3.1) to provide an optimal fusion of those in the manner that some norm of individual noise transfer functions are minimized[2]. For the filtered system, including actuation dynamics with transfer function $T_{uy}(s)$ (according to Figure 4.1), either a

[1] Or at least they are in lots of cases a result of an empirical process.
[2] Either the 2-norm, the ∞-norm or a combination of these, called mixed sensitivity filters.

ssKF or a non-linear Kalman filter (EKF) can be designed to handle non-linearities of piezoelectric actuators (such as hysteresis or creep) and known non-linearities of displacement sensors (such as the thermal drift).

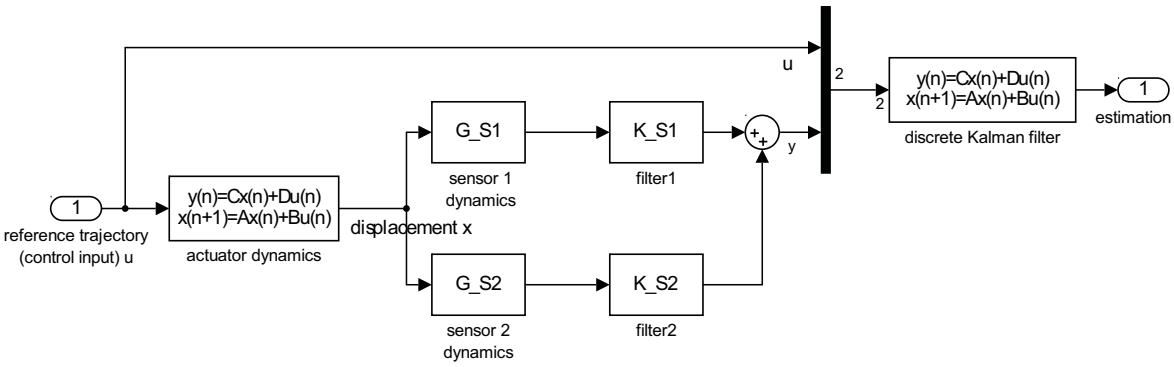


Figure 4.1: Model based estimation with noise-optimal filtered sensors

Advanced methods of filters and fusions

More advanced methods of filtering are available with more effort to design and to implement in order to get a more accurate position estimation in real time [36].

(a) The Moving Horizon Estimation (MHE) as an optimization approach to estimate unknown parameters seems promising in some cases [29]. Each time step a linear program or even a non-linear program is solved and prediction of states is based on the solution. Usually it requires a higher computational effort than deterministic approaches such as the Kalman filter. To use a MHE an accurate analytic system model is needed with understanding of as many sub-dynamics, which excludes the usage of a sensor model acquired by measurements and black-box model fitting [9].

(b) In case of simultaneously using multiple sensors to acquire a position, more advanced fusion algorithms are available if the equations for the integration of these measurements into a common state estimate are explicitly derived.

(c) Two stage robust Kalman filters might further improve the position estimation accuracy in the presence of model uncertainties [7].

(d) Some practical systems have a non-linear state update or non-linear measurement equations, therefore Unscented Kalman Filter (UKF) might bring benefits compared to linear minimum variance estimators if non-linear system dynamics $\dot{\mathbf{x}}(\mathbf{t}) = \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t), t\right)$ is known. The potency of the UKF is in expansion of the system dynamics to Taylor series yielding a linear equation for error propagation.

(e) *Validation gates rejects outlier measurements, resulting in a more reliable estimation* [15]. It would be worth to investigate validation gates, or in general the particle filter for position estimation.

(f) In case of using the traditional Kalman filter, some assumption about noise distribution have to be met. If either process noise **w** or measurement noise **v** or both are *skewed* or otherwise *pathological* a Bayesian filter or particle filter might be useful.

(g) Kalman filters for unknown trajectories can deal with non-linearities, as proposed by [6]. They require at least a partially known system with some assumptions about the unknown input. According [6] they can estimate for input bias, as this is a well known phenomena for piezoelectric actuators as well as for some type of sensors.

(h) Non causal filters such as Kalman smoother were not handled since the usage in closed loop control is questionable, although some of them can be implemented with constant phase lag [29].

(i) Even a simple low-pass filter $\dot{x} = -x$ might be of interest. It is known, that it reduces noise by limiting the bandwidth. However, one major advantage when using such a filter is, that unknown system can be time-variant or even non-linear, no errors are made since there is no system model behind it.

à la fin
Daniel Piri 2014

Bibliography

[1] "Robust filters," in Computational Surface and Roundness Metrology. Springer London, 2009, pp. 87–92. [Online]. Available: http://dx.doi.org/10.1007/978-1-84800-297-5_11

[2] S. Abu and A. Pantazi, "Nanopositioning with multiple sensors, a case study in data storage," IEEE Transactions on Control System Technology, no. 20, 2012.

[3] J. Ackermann, A. Bartlett, D. Kaesbauer, W. Sienel, and R. Steinhauser, Robust control. Springer Berlin, 1993.

[4] Analog Devices, "AD7985 16bit 2.5MSPS SAR-ADC," 2013, accessed: 2014-08-01. [Online]. Available: http://www.analog.com/en/analog-to-digital-converters/ad-converters/ad7985/products/product.html

[5] M. J. Caruso, "A new perspective on magnetic field sensing," Sensors-Peterborough, vol. 15, 1998.

[6] M. Darouach and M. Zasadzinski, "Unbiased Minimum Variance Estimation for Systems with Unknown Exogenous Inputs," Automatica, no. Vol 33, No 4., 1997.

[7] F. Daum, "Nonlinear filters: beyond the kalman filter," Aerospace and Electronic Systems Magazine, IEEE, vol. 20, no. 8, pp. 57–69, 2005.

[8] H. Durrant-Whyte, "Multi sensor data fusion," Springer handbook of robotics, 585-610, 2008.

[9] P. Eaton and P. West, Atomic Force Microscopy. Oxford University Press, USA, 2004.

[10] A. A. Eielsen, T. Poloni, T. A. Johansen, and J. T. Gravdahl, "Experimental Comparison of Online Parameter Identification Schemes for a Nanopositioning Stage with Variable Mass," IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2011.

[11] A. J. Fleming, A. G. Wills, and S. Moheimani, "Sensor fusion for improved control of piezoelectric tube scanners," IEEE Transactions on Control Systems Technology, vol. 16, no. 6, pp. 1265–1276, 2008.

[12] A. J. Fleming, "Measuring and Predicting Resolution in Nanopositioning Systems," Mechatronics, 2013.

[13] A. J. Fleming, "A review of nanometer resolution position sensors: Operation and performance," Sensors and Actuators A: Physical, vol. 190, pp. 106–126, 2013.

[14] A. J. Fleming and K. K. Leang, "Integrated strain and force feedback for high-performance control of piezoelectric actuators," Sensors and Actuator:s A. Physical, vol. 161, no. 1-2, pp. 256–265, 2010.

[15] J. Gao and C. J. Harris, "Some remarks on Kalman filters for the multisensor fusion," Information Fusion, vol. 3, no. 3, pp. 191–201, 2002.

[16] HBM Test and Measurement, "HBM," accessed: 2014-06-20. [Online]. Available: http://www.hbm.com

[17] W. Heywang, K. Lubitz, and W. Wersing, Piezoelectricity: evolution and future of a technology. Springer, 2008, vol. 114.

[18] L. Kang-Zhi, Linear Robust Control. Corona Publishing Co., LTD., 2002.

[19] A. Kugi, Automatisierung. Automation and Control Institute, Vienna University of Technology, 2013, (german).

[20] S. Kuiper and G. Schitter, "Active damping of a piezoelectric tube scanner using self-sensing piezo actuation," Mechatronics, 2010.

[21] S. Kuiper and G. Schitter, "Mimo self-sensing control of a piezoelectric tube scanner," pp. 811–817, 2009.

[22] D.-J. Lee, "Nonlinear estimation and multiple sensor fusion using unscented information filtering," Signal Processing Letters, IEEE, vol. 15, pp. 861–864, 2008.

[23] M. M. Miller, G. A. Prinz, P. Lubitz, L. Hoines, J. J. Krebs, S. F. Cheng, and F. G. Parsons, "Novel absolute linear displacement sensor utilizing giant magnetoresistance elements," Journal of Applied Physics, 1997.

[24] C. Min-ge, "Design of Equiripple Linear-Phase FIR Filters Using MATLAB," IEEE, Conference on Control, Automation ans System Engineering, 2011.

[25] MIT., "Material parameters," accessed: 2014-05-18. [Online]. Available: http://web.mit.edu/course/3/3.11/www/modules/props.pdf

[26] R. Munnig Schmidt, G. Schitter, and J. van Eijk, "The design of high performance mechatronics," 2011.

[27] M. Neubauer, R. Oleskiewicz, K. Popp, and T. Krzyzynski, "Optimization of damping and absorbing performance of shunted piezo elements utilizing negative capacitance," Journal of Sound and Vibration, 2006.

[28] NVE Corporation, "GMR Sensor Catalog," 2010, accessed: 2014-06-10. [Online]. Available: http://www.nve.com/Downloads/apps.pdf

[29] T. Poloni, A. A. Eielsen, B. Rohal-Ilkiv, and T. A. Johansen, "Moving Horizon Observer for Vibration Dynamics with Plant Uncertainties in Nanopositioning System Estimation," American Control Conference, 2012.

[30] P. Ripka, M. Tondra, J. Stokes, and R. Beech, "AC-driven AMR and GMR magnetoresistors," Sensors and Actuators A: Physical, 1999.

[31] G. Schitter, R. Stark, and A. Stemmer, "Sensor for closed-loop piezo control: strain gauges versus optical sensors," Measurement Science and Technology, no. 13, 2002.

[32] G. Schitter and A. Stemmer, "Identification and open-loop tracking control of a piezoelectric tube scanner for high-speed scanning-probe microscopy," IEEE Transactions on Control Systems Technology, vol. 12, no. 3, pp. 449–454, 2004.

[33] Y. Shan, J. E. Speich, and K. K. Leang, "Low-Cost IR Reflective Sensors for Submicrolevel Position Measurement and Control," IEEE/ASME Transactions on Mechatronics, 2008.

[34] D. Simon, "My Philosophy as a Professor," Cleveland State University (Department of Electrical and Computer Engineering). [Online]. Available: http://academic.csuohio.edu/simond/

[35] D. Simon, "Kalman filtering," Embedded Systems Programming, vol. 14, no. 6, pp. 72–79, 2001.

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

[37] S. Skogestad and I. Postlethwaite, Multivariable feedback control: analysis and design. Wiley New York, 2007, vol. 2.

[38] S. Soloman, Sensors handbook. McGraw-Hill, Inc., 2009.

[39] Stanford Research Systems Inc., "Model SR560 lowe noise preamplifier operational handbook," 2011.

[40] S. L. Star, "Scientific work and uncertainty," Social studies of science, vol. 15, no. 3, pp. 391–427, 1985.

[41] J. Steininger, S. Kuiper, S. Ito, and G. Schitter, "Schnelle Rasterkraftmikroskopie durch moderne Regelungstechnik und mechatronische Systemintegration," e & i Elektrotechnik und Informationstechnik, 2012.

[42] Texas Instruments, Incorporated, "Photodiode Monitoring with Op Amps," Application Bulletin, accessed: 2014-06-12. [Online]. Available: http://www.ti.com/lit/an/sboa035/sboa035.pdf

[43] K. Uchino, Piezoelectric actuators and ultrasonic motors. Springer, 1997, vol. 1.

[44] M. Vopálenský, P. Ripka, and A. Platil, "Precise magnetic sensors," Sensors and Actuators A: Physical, 2003.

[45] Wikipedia, "Bulk modulus," accessed: 2014-05-18. [Online]. Available: http://en.wikipedia.org/wiki/Bulk_modulus

[46] Wikipedia Commons, "Magnetic Hysteresis," accessed: 2014-06-25. [Online]. Available: http://en.wikipedia.org/wiki/Magnetic_hysteresis

[47] H. W. Yoo, S. Ito, M. Verhaegen, and G. Schitter, "Transformation-Based Iterative Learning Control for Non-Collocated Sensing of a Galvanometer Scanner," European Control Conference, 2013.

[48] L. Yung-Tien, F. Rong-Fong, and H. Tai-Kun, "Dynamic responses of a precision positioning table impacted by a soft-mounted piezoelectric actuator," Precision Engineering, 2004.

Appendix

1 Kalman filtering in Matlab

The LTI discrete-time dynamic system (actuation including sensor) is given by

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{G}\mathbf{w}_k$$
$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k$$

with input $\mathbf{u} \in \mathbb{R}^p$, measurement $\mathbf{y} \in \mathbb{R}^q$ and random variables $\mathbf{w} \in \mathbb{R}^r$, $\mathbf{v} \in \mathbb{R}^q$ with restrictions of being white noise with normal distribution and zero mean, uncorrelated to each other, with covariances

$$\mathrm{cov}(\mathbf{w}) = \mathbf{Q} \in \mathbb{R}^{r\times r}$$
$$\mathrm{cov}(\mathbf{v}) = \mathbf{R} \in \mathbb{R}^{q\times q}$$

representing process and measurement noise, respectively. Matrices have appropriate dimension $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times p}$, $\mathbf{G} \in \mathbb{R}^{n\times r}$, $\mathbf{C} \in \mathbb{R}^{q\times n}$.
The Kalman Filter (EKF) is given by the following function with inputs u (control input, reference trajectory) and y (noisy measurement) and outputs est (estimation of displacement), states, and 2-norm of covariance matrix $\mathbf{P}$.

```
function [est,normP,states] = kalman(u,y,par)
    % inputs       u (control trajectory)
    %              y (noisy sensor signal)
    %              struct p holds system parameters
    % outputs      est (output estimation)
    %              normP (2-norm of covariance matrix P (for observing performance))
    %              states (state vector)
    persistent x P

    A = par.A;  % system matrix (n,n)
    B = par.B;  % input matrix (n,1)
    C = par.C;  % output matrix (1,n)
    Q = par.Q;  % covariance of process noise (1,1)
    R = par.R;  % covariance of sensor noise (1,1)
    G = par.G;  % process noise is related to input, G = B was set
```

```
    if isempty(x)
        x = pinv(C)*y;
        P = pinv(C)*R*pinv(C');
        disp('initial vectors set')
    end

    % Compute Kalman gain factor:
    K = P*C'*inv(C*P*C'+R);

    %Kalman measurement update
    x=x+K*(y-C*x);

    %covariance update
    P = (eye(3)-K*C)*P;

    %prediction for state and covariance
    x=A*x+B*u;
    P = A*P*A' + G*Q*G';

    %output estimation
    est=C*x;

    normP = norm(P,2);
    states = x;
end
```

2 H_2, H_∞ filter synthesis in Matlab

For the sensors with continuous-time transfer function models *TF_S1* and *TF_S2* as well as noise models *N_S1* and *N_S2* the H_2 (case 1) or H_∞ filters (exact: case (a), LMI-method: case (b)) are computes as follows:

```
clear all;
clc;

% plot settings
bode_opt = bodeoptions;
bode_opt.mapCSTPrefs;
bode_opt.FreqUnits = 'Hz';
bode_opt.PhaseVisible = 'off';
bode_opt.Xlim = [10^1, 10^7];
bode_opt.Grid = 'on';

% weight defines
N_S1 = getSensorModel_1();  % Noise transfer function of 2nd order in state space
N_S2 = getSensorModel_2();  % Noise transfer function of 2nd order in state space
T_W = zpk([-1e6*2*pi],[-1e5*2*pi],1e-1);  % Weight for fine tuning

% transfer functions
W0 = 1E3*2*pi;
W1 = W0*100;
TF_S1 = 0.1 * tf([1/W1 1],[1/W0 1]);
```

```
W0 = 1E4*2*pi;
W1 = W0*1E3;
TF_S2 = 0.55 * tf([1/W1 1],[1/W0 1]);

subplot(2,1,1); cla
bode_opt.Title.String = 'Weights and Sensors';
h=bodeplot(N_S1,N_S2, TF_S1, TF_S2, T_W, bode_opt);
set(findobj('type','line'),'LineWidth',2);
legend show;

% General plant definition
[Ga,Gb,Gc,Gd] = linmod('generalizedModel_m4');
disp(size(Ga))
G = ltisys(Ga,Gb,Gc,Gd);
[G, Gsig] = sysbal(G);
[Ga,Gb,Gc,Gd,Ge] = ltiss(G);
G = ss(Ga,Gb,Gc,Gd);

if 0
    % H2 Riccati (case 1)
    [gopt, K] = h2syn(G, 2, 1);
elseif 1
    % Hinf Riccati (case a)
    [K,CL,GAM, INFO]=hinfsyn(G, 2, 1);
elseif 0
    % Hinf LMI (case b)
    [gopt, K] = hinflmi(G, [2, 1]);
end

% Post processing
K_S1 =  K(1);   % Transfer function from S1 to Est
K_S2 =  K(2);   % Transfer function from S2 to Est

% time continous full orded filters
series_1 = series(TF_S1,K_S1); %filtered sensor S1
series_2 = series(TF_S2,K_S2); %filtered sensor S2
K_total = series_1 + series_2;
   % plot filters
      figure;
      bode_opt.Title.String = 'Filters';
      bode_opt.PhaseVisible = 'on';
      bode(K_S1, K_S2, bode_opt);
      set(findobj('type','line'),'LineWidth',2);
      legend show;

   % plot filteres sensors
      figure;
      bode_opt.Title.String = 'Filtered sensors';
      bode(series_1,series_2,K_total,bode_opt);
      set(findobj('type','line'),'LineWidth',2);
      l=legend('$G_{yf1,x}(s)$','$G_{yf2,x}(s)$','$G_{\hat{x}x}(s)$');
      set(l,'Interpreter','Latex');

figure; step(K_S1,K_S2) % step response of filters
```

```
figure; pzmap(K_S1,K_S2) % pole-zero map of filters

figure; bode(TF_S1,TF_S2,bode_opt); % TF of sensors

%model order reduction
   bode_opt.MagLowerLim=-20;
   clc;
   close all;
   figure('OuterPosition',[50 50 900 750]); %[left, bottom, width, height]
   SPS = 500e3;

   [K_R1, dK1] = FilterSimplify(K_S1,SPS,1e-1);
   [K_R2, dK2] = FilterSimplify(K_S2,SPS,1e-1);
   bode_opt.Title.String = 'Filters';
   bode_opt.PhaseVisible = 'on';

   bodeplot(K_S1, K_R1, dK1, K_S2, K_R2, dK2, bode_opt);
   set(findobj('type','line'),'LineWidth',2);
```

3 STM32 C-program of Kalman filtering

main.c

```
#include "stm32f4xx.h"
#include <stdio.h>
#include <math.h>
//#define TURBO 1
#define INTR    1
//#define TEST 1

#define CONT 0

#define __FPU_PRESENT 1
#define __FPU_USED 1

#define BAUDRATE 115200
#define BS 1000
float reference[BS];
float measure[BS];
float estimate[BS];
uint16_t bi = 0;

GPIO_InitTypeDef  GPIO_InitStructure;
EXTI_InitTypeDef  EXTI_InitStructure;
NVIC_InitTypeDef   NVIC_InitStructure;
DMA_InitTypeDef DMA_InitStructure;
TIM_TimeBaseInitTypeDef TIM_TimeBaseStructure;
TIM_OCInitTypeDef  TIM_OCInitStructure;
USART_InitTypeDef USART_InitStructure;
SPI_InitTypeDef  SPI_InitStructure;
ADC_InitTypeDef ADC_InitStructure;
ADC_CommonInitTypeDef ADC_CommonInitStructure;
```

```
void Delay(__IO uint32_t nCount);
void USART_Config(void);
void GPIO_config(void);
void transmitUART(void);
void InitAdc(void);
void SPI_Config(void);
void ADC_Config(void);
void EXTI_Config(void);
void TIM3_Config(void);
uint16_t SPI_send(uint16_t data);
void USART_puts(USART_TypeDef* USARTx, volatile char *s);
void USART_cc(uint16_t cc);

__IO uint16_t CCR1_Val = 1000;  //max 65535 counts. Lower count higher
    freq
// 1000 ... 3kHz toggling freq
uint16_t PrescalerValue = 0;

uint16_t cc=13, cu=13; //adc readouts
uint16_t flag_usart=0;
int i;
char buffer[64];

// ------- generated by ss2C() in Matlab at_10:53 on_05-Aug-2014
// discrete time plant with sampling time_1e-05s (fs=100000Hz)
const float a11=-0.797918, a12=-135827, a13=-9.14032e+09, a14=-4.20646e
    +14, a15=-8.16826e+18, a16=-4.42156e+22,
a21=6.91459e-07, a22=-0.736756, a23=-88793.1, a24=-7.21547e+09, a25
    =-2.07282e+14, a26=-1.2937e+18,
a31=2.02313e-11, a32=2.48099e-06, a33=0.639392, a34=-32474.4, a35
    =-9.72682e+08, a36=-6.13999e+12,
a41=9.60193e-17, a42=2.87245e-11, a43=9.0123e-06, a44=0.906685, a45
    =-2845.61, a46=-1.80477e+07,
a51=2.82236e-22, a52=1.20984e-16, a53=4.79225e-11, a54=9.79798e-06, a55
    =0.993775, a56=-39.5884,
a61=6.19096e-28, a62=3.36997e-22, a63=1.63096e-16, a64=4.96459e-11, a65
    =9.98901e-06, a66=0.99993;
const float b11=6.91459e-07,
b21=2.02313e-11,
b31=9.60193e-17,
b41=2.82236e-22,
b51=6.19096e-28,
b61=1.09471e-33;
const float c11=0, c12=1.0821e+06, c13=1.94506e+11, c14=9.32798e+16, c15
    =4.18467e+21, c16=-3.02862e+27;
const float k11=1.2847e-08,
k21=-1.99394e-13,
k31=-2.71934e-18,
k41=4.8536e-23,
k51=6.46724e-28,
k61=-3.56276e-32;   //Kalman gain

    float x1=0,x2=0,x3=0,x4=0,x5=0,x6=0,inov,est, y, u;
```

```
//-------------------------------------------------------------
int main(void) {
//-------------------------------------------------------------
    GPIO_config();
    USART_Config();
    InitAdc();
    SPI_Config();
    Delay(0xFFFF);

    #ifdef TURBO //Turbo mode! continous conversion
        //GPIO_SetBits(GPIOD, GPIO_Pin_10); //set TURBO mode (no stop btw
            . conv)
        GPIO_SetBits(GPIOD, GPIO_Pin_15); //blue for AQU
        while(1){ //should run at 100kHz (ss)
            //sample y (16 bit)
            GPIO_ResetBits(GPIOD, GPIO_Pin_9); //initiates conversion
            GPIO_SetBits(GPIOD, GPIO_Pin_9);
            GPIO_ResetBits(GPIOD, GPIO_Pin_9);
            cc = SPI_send(0xFFFF);  //measurement 0..2^15
            GPIO_SetBits(GPIOD, GPIO_Pin_9);

            //sample u
            ADC_SoftwareStartConv(ADC1); // Start the conversion
            while(!ADC_GetFlagStatus(ADC1, ADC_FLAG_EOC));//Processing
               the conversion
            cu = ADC_GetConversionValue(ADC1); //measurement 0..4095

            //scale
            y = (float)(cc) / 32768;
            u = (float)(cu) / 4096;


            if (bi<BS){
                measure[bi] = y; //measurement
                reference[bi] = u; //reference
                bi++;
            }
            else{  //ready
#if (CONT == 1)
                    bi=0;
#else           //ready
                GPIO_ResetBits(GPIOD, GPIO_Pin_15); //blue off for AQU
                   end
                transmitUART();
                while(1){
                    GPIO_ToggleBits(GPIOD, GPIO_Pin_12); //blink green
                    Delay(0x8FFFFF);
                }
#endif
            }
        }
    #endif
```

```
    #ifdef INTR //
       ------------------------------------------------------------------------

    // normal operation: either PD9 in TIM_IRQn initiates conversion and
       transfer
    // or external CNV triggers. Enable EXTI
    EXTI_Config();
    //TIM3_Config();
    while(1) {

        if (flag_usart){
            flag_usart=0;
            if (bi<BS){
                measure[bi] = y; //measurement
                estimate[bi] = est; //estimation
                reference[bi] = u; //reference
                bi++;
            }
            else{
                #if (CONT == 1)
                    bi=0;
                #else           //read
                    EXTI_DeInit();
                    transmitUART();
                    while(1){
                        GPIO_ToggleBits(GPIOD, GPIO_Pin_12); //blink
                           green
                        Delay(0x8FFFFF);
                    }
                #endif
                }
                GPIO_ResetBits(GPIOD, GPIO_Pin_13);
            }
    }
    #endif
    #ifdef TEST //
       ------------------------------------------------------------------------

        GPIO_SetBits(GPIOD, GPIO_Pin_15);
    while(1){
        GPIO_ResetBits(GPIOD, GPIO_Pin_9); //initiates conversion
        GPIO_SetBits(GPIOD, GPIO_Pin_9);
        GPIO_ResetBits(GPIOD, GPIO_Pin_9);
        cc = SPI_send(0xFFFF); //measurement
        GPIO_SetBits(GPIOD, GPIO_Pin_9);
    }

    #endif
}

//-------------------------------------------------------------
void transmitUART(void){
//-------------------------------------------------------------
    GPIO_SetBits(GPIOD, GPIO_Pin_14); //red on for transmission
```

```
        printf("mmmmmmeasurement");
        for(bi=0; bi<BS; bi++){
            sprintf(buffer,"%-1.20f,",measure[bi]);
            for (i=0;buffer[i]!='\0';i++){
                USART_SendData(USART3, buffer[i]);
                while (USART_GetFlagStatus(USART3, USART_FLAG_TC) ==
                    RESET);
            }
        }
        printf("estimation");
        for(bi=0; bi<BS; bi++){
            sprintf(buffer,"%-1.20f,",estimate[bi]);
            for (i=0;buffer[i]!='\0';i++){
                USART_SendData(USART3, buffer[i]);
                while (USART_GetFlagStatus(USART3, USART_FLAG_TC) ==
                    RESET);
            }
        }
        printf("reference");
        for(bi=0; bi<BS; bi++){
            sprintf(buffer,"%-1.20f,",reference[bi]);
            for (i=0;buffer[i]!='\0';i++){
                USART_SendData(USART3, buffer[i]);
                while (USART_GetFlagStatus(USART3, USART_FLAG_TC) ==
                    RESET);
            }
        }
    printf("\n\r\n\r"); //Terminator
    GPIO_ResetBits(GPIOD, GPIO_Pin_14); //red off
}
//-------------------------------------------------------------
void InitAdc(){ //init 12 bit ADC
//-------------------------------------------------------------
  ADC_InitTypeDef ADC_InitStructure;
  //ADC_CommonInitTypeDef ADC_CommonInitStructure;
    GPIO_InitTypeDef gpioInit;


  RCC_APB2PeriphClockCmd(RCC_APB2Periph_ADC1, ENABLE);
    RCC_AHB1PeriphClockCmd(RCC_AHB1ENR_GPIOAEN,ENABLE);
  //RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA , ENABLE);
  /* Configure PA1 as analog input */
  gpioInit.GPIO_Pin = GPIO_Pin_1;
  gpioInit.GPIO_Mode = GPIO_Mode_AN;
    gpioInit.GPIO_PuPd = GPIO_PuPd_NOPULL;
  GPIO_Init(GPIOA, &gpioInit);

 ADC_DeInit();
 ADC_InitStructure.ADC_DataAlign = ADC_DataAlign_Right;//data converted
    will be shifted to right
 ADC_InitStructure.ADC_Resolution = ADC_Resolution_12b;//Input voltage is
     converted into a 12bit number giving a maximum value of 4096
 ADC_InitStructure.ADC_ContinuousConvMode = ENABLE; //the conversion is
    continuous, the input data is converted more than once
 ADC_InitStructure.ADC_ExternalTrigConv = ADC_ExternalTrigConv_T1_CC1;
```

```
 // conversion is synchronous with TIM1 and CC1
 ADC_InitStructure.ADC_ExternalTrigConvEdge =
    ADC_ExternalTrigConvEdge_None;//no trigger for conversion
 ADC_InitStructure.ADC_NbrOfConversion = 1;//I think this one is clear :p
 ADC_InitStructure.ADC_ScanConvMode = DISABLE;//The scan is configured in
     one channel
 ADC_Init(ADC1,&ADC_InitStructure);//Initialize ADC with the previous
    configuration
 //Enable ADC conversion
 ADC_Cmd(ADC1,ENABLE);
 //Select the channel to be read from
 ADC_RegularChannelConfig(ADC1,ADC_Channel_1,1,ADC_SampleTime_3Cycles);
    //15

//  Delay(0x3FF);
//  ADC_ResetCalibration(ADC1);
//  while(ADC_GetResetCalibrationStatus(ADC1));
//  ADC_StartCalibration(ADC1);
//  while(ADC_GetCalibrationStatus(ADC1));
//  ADC_SoftwareStartConv(ADC1);
}

//-------------------------------------------------------------
void SPI_Config(void){
//-------------------------------------------------------------
    GPIO_InitTypeDef GPIO_InitStructure;

    /* Enable the SPI periph */
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA | RCC_AHB1Periph_GPIOE,
       ENABLE);
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_SPI1, ENABLE); //Use SPI 1
    GPIO_PinAFConfig(GPIOA, GPIO_PinSource5, GPIO_AF_SPI1); // SCK
    GPIO_PinAFConfig(GPIOA, GPIO_PinSource6, GPIO_AF_SPI1); // MISO
    GPIO_PinAFConfig(GPIOA, GPIO_PinSource7, GPIO_AF_SPI1); // MOSI

    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructure.GPIO_PuPd  = GPIO_PuPd_DOWN;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    //SPI SCK, MISO, MOSI
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_5 | GPIO_Pin_6 | GPIO_Pin_7;
    GPIO_Init(GPIOA, &GPIO_InitStructure);

    //config SPI
    SPI_I2S_DeInit(SPI1);
    SPI_InitStructure.SPI_Direction = SPI_Direction_2Lines_FullDuplex;
    SPI_InitStructure.SPI_DataSize = SPI_DataSize_16b;
    SPI_InitStructure.SPI_CPOL = SPI_CPOL_Low;
    SPI_InitStructure.SPI_CPHA = SPI_CPHA_1Edge;
    SPI_InitStructure.SPI_NSS = SPI_NSS_Soft | SPI_NSSInternalSoft_Set;
    SPI_InitStructure.SPI_BaudRatePrescaler = SPI_BaudRatePrescaler_2; //
       2..256 //84MHz will be divided by prescaler
    SPI_InitStructure.SPI_FirstBit = SPI_FirstBit_MSB;
    //SPI_InitStructure.SPI_CRCPolynomial = 7;
    SPI_InitStructure.SPI_Mode = SPI_Mode_Master;
```

```
    SPI_Init(SPI1, &SPI_InitStructure);

    NVIC_InitStructure.NVIC_IRQChannel = SPI1_IRQn;
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
    NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
    NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
    NVIC_Init(&NVIC_InitStructure);

    //SPI_I2S_ITConfig(SPI1, SPI_I2S_IT_RXNE, ENABLE);
    SPI_Cmd(SPI1, ENABLE);

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_3;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_Init(GPIOE, &GPIO_InitStructure);
    GPIO_SetBits(GPIOE, GPIO_Pin_3); //chip select pin for LIS
}


//-------------------------------------------------------------
void EXTI_Config(void){
//-------------------------------------------------------------
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOB, ENABLE);
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_SYSCFG, ENABLE);

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0;   //B0 .. input CNV
       interrupt
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_100MHz;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP;
      GPIO_Init(GPIOB, &GPIO_InitStructure);

    SYSCFG_EXTILineConfig(EXTI_PortSourceGPIOB, EXTI_PinSource0)    ;
    EXTI_InitStructure.EXTI_Line = EXTI_Line0;
    EXTI_InitStructure.EXTI_Mode = EXTI_Mode_Interrupt;
    EXTI_InitStructure.EXTI_Trigger = EXTI_Trigger_Rising;
    EXTI_InitStructure.EXTI_LineCmd = ENABLE;
                                                    EXTI_Init(&
       EXTI_InitStructure);

      // Enable and set EXTI Line0 Interrupt to the lowest priority
    NVIC_InitStructure.NVIC_IRQChannel = EXTI0_IRQn;
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0x0F;
    NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0x0F;
    NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
    NVIC_Init(&NVIC_InitStructure);
}

//-------------------------------------------------------------
uint16_t SPI_send(uint16_t data){
//-------------------------------------------------------------
    SPI1->DR = data; // write data to be transmitted to the SPI data
       register
```

```
    while( !(SPI1->SR & SPI_I2S_FLAG_TXE) ); // wait until transmit
       complete
    while( !(SPI1->SR & SPI_I2S_FLAG_RXNE) ); // wait until receive
       complete
    while( SPI1->SR & SPI_I2S_FLAG_BSY ); // wait until SPI is not busy
       anymore
    return SPI1->DR; // return received data from SPI data register
}

//---------------------------------------------------------------
void GPIO_config(void){
//---------------------------------------------------------------
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOD, ENABLE);

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_9 | GPIO_Pin_10 | GPIO_Pin_12
       | GPIO_Pin_13 | GPIO_Pin_14 | GPIO_Pin_15;
    // Pin 12-green, 13-orange, 14-red, 15-blue
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
    GPIO_Init(GPIOD, &GPIO_InitStructure);

    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA, ENABLE);
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0; //pushbutton
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_DOWN;
    GPIO_Init(GPIOA, &GPIO_InitStructure);
}

//---------------------------------------------------------------
void TIM3_Config(void){
//---------------------------------------------------------------
    NVIC_InitTypeDef NVIC_InitStructure;
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM3, ENABLE); // TIM3 clock
       enable

    NVIC_InitStructure.NVIC_IRQChannel = TIM3_IRQn;   // Enable the TIM3
       gloabal Interrupt
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
    NVIC_InitStructure.NVIC_IRQChannelSubPriority = 1;
    NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
    NVIC_Init(&NVIC_InitStructure);

    /* Compute the prescaler value */
    PrescalerValue = (uint16_t) ((SystemCoreClock / 2) / 6000000) - 1;

    /* Time base configuration */
    TIM_TimeBaseStructure.TIM_Period = 65535;
    TIM_TimeBaseStructure.TIM_Prescaler = 0;
    TIM_TimeBaseStructure.TIM_ClockDivision = 0;
    TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
    TIM_TimeBaseInit(TIM3, &TIM_TimeBaseStructure);
```

```
    // Prescaler configuration
    TIM_PrescalerConfig(TIM3, PrescalerValue, TIM_PSCReloadMode_Immediate
        );

    /* Output Compare Timing Mode configuration: Channel1 */
    TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_Timing;
    TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
    TIM_OCInitStructure.TIM_Pulse = CCR1_Val;
    TIM_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_High;
    TIM_OC1Init(TIM3, &TIM_OCInitStructure);

    TIM_OC1PreloadConfig(TIM3, TIM_OCPreload_Disable);
    TIM_ITConfig(TIM3, TIM_IT_CC1, ENABLE); // TIM Interrupts enable
    TIM_Cmd(TIM3, ENABLE);  // TIM3 enable counter
}

//-------------------------------------------------------------
void Delay(__IO uint32_t nCount){
//-------------------------------------------------------------
    while(nCount--)
        ;
}

//-------------------------------------------------------------
void USART_Config(){
//-------------------------------------------------------------
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_USART3, ENABLE);
    // enable the peripheral clock for the pins PC10 for TX and PC11 for
       RX
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOC, ENABLE);

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_10 | GPIO_Pin_11;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF; // the pins are
       configured as alternate function so the USART peripheral has
       access to them
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz; // this defines the
        IO speed and has nothing to do with the baudrate!
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP; // this defines the
       output type as push pull mode (as opposed to open drain)
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP; // this activates the
       pullup resistors on the IO pins
    GPIO_Init(GPIOC, &GPIO_InitStructure);

    /* The RX and TX pins are now connected to their AF
     * so that the USART1 can take over control of the pins  */

    GPIO_PinAFConfig(GPIOC, GPIO_PinSource10, GPIO_AF_USART3);
    GPIO_PinAFConfig(GPIOC, GPIO_PinSource11, GPIO_AF_USART3);

    /* Now the USART_InitStruct is used to define the
     * properties of USART1          */
    USART_DeInit(USART3);
    USART_InitStructure.USART_BaudRate = BAUDRATE*3.072;
       // BUG !!!
```

```
    USART_InitStructure.USART_WordLength = USART_WordLength_8b;// we want
        the data frame size to be 8 bits (standard)
    USART_InitStructure.USART_StopBits = USART_StopBits_1;       // we
       want 1 stop bit (standard)
    USART_InitStructure.USART_Parity = USART_Parity_No;      // we don't
       want a parity bit (standard)
    USART_InitStructure.USART_HardwareFlowControl =
       USART_HardwareFlowControl_None; // we don't want flow control (
       standard)
    USART_InitStructure.USART_Mode = USART_Mode_Rx | USART_Mode_Tx; // we
        want to enable the transmitter
    USART_Init(USART3, &USART_InitStructure);

    USART_Cmd(USART3, ENABLE);
    USART_SendData(USART3, '0'); //dummy byte at first
}

//-------------------------------------------------------------
void USART_cc(uint16_t cc) {
//-------------------------------------------------------------
    USART_SendData(USART3, (cc&0xFF00) >> 8); //upper byte
    while (USART_GetFlagStatus(USART3, USART_FLAG_TC) == RESET);
    USART_SendData(USART3, cc&0x00FF);  //lower byte
    while (USART_GetFlagStatus(USART3, USART_FLAG_TC) == RESET);
    USART_SendData(USART3, ','); //delimiter 0x2C
    while (USART_GetFlagStatus(USART3, USART_FLAG_TC) == RESET);

}

//-------------------------------------------------------------
int fputc(int ch, FILE *f) {
//-------------------------------------------------------------
    // Retargets the C library printf function to the USART.
    USART_SendData(USART3, (uint8_t) ch);

    // Loop until the end of transmission
    while (USART_GetFlagStatus(USART3, USART_FLAG_TC) == RESET);
    return ch;
}

//-------------------------------------------------------------
void USART_puts(USART_TypeDef* USARTx, volatile char *s){
//-------------------------------------------------------------
    while(*s){
        // wait until data register is empty
        while( !(USARTx->SR & 0x00000040) );
        USART_SendData(USARTx, *s);
        *s++;
    }
}
```

stm32f4xx_it.c

```
#include "stm32f4xx_it.h"
#include <stdio.h>


extern uint16_t cc, cu;
extern uint16_t flag_usart;
extern const float a11, a12, a13, a14, a15, a16, a21, a22, a23, a24, a25,
    a26, a31, a32, a33, a34, a35, a36, a41, a42, a43, a44, a45, a46, a51,
    a52, a53, a54, a55, a56, a61, a62, a63, a64, a65, a66;
extern const float b11, b21, b31, b41, b51, b61;
extern const float c11, c12, c13, c14, c15, c16;
extern const float k11, k21, k31, k41, k51, k61;
extern float x1,x2,x3,x4,x5,x6,inov,est,y,u;

uint16_t SPI_send(uint16_t data);

...

//-------------------------------------------------------------
void EXTI0_IRQHandler(void){
//-------------------------------------------------------------
 float x1n,x2n,x3n,x4n,x5n,x6n;
    //Check if EXTI_Line0 is asserted
    if(EXTI_GetITStatus(EXTI_Line0) != RESET) {
        if (flag_usart){            //if flag could not be resetted (too
           high sampling)
            GPIO_SetBits(GPIOD, GPIO_Pin_14); //red fault
        }
            GPIO_SetBits(GPIOD, GPIO_Pin_13); //orange
            //sample y (16 bit)
            cc = SPI_send(0xFFFF);                  //measurement
               0..32000

            //sample u
            ADC_SoftwareStartConv(ADC1); // Start the conversion
            while(!ADC_GetFlagStatus(ADC1, ADC_FLAG_EOC));//Processing
               the conversion
            cu = ADC_GetConversionValue(ADC1); //measurement 0..4096

            //scale
            y = (float)(cc) / 65535;
            u = (float)(cu) / 4095;

// ------- generated by ssequation2C() in Matlab at_11:05 on_05-Aug-2014
//Kalman prediction x=A*x+B*u;
x1n=a11*x1+a12*x2+a13*x3+a14*x4+a15*x5+a16*x6+ b11*u;
x2n=a21*x1+a22*x2+a23*x3+a24*x4+a25*x5+a26*x6+ b21*u;
x3n=a31*x1+a32*x2+a33*x3+a34*x4+a35*x5+a36*x6+ b31*u;
x4n=a41*x1+a42*x2+a43*x3+a44*x4+a45*x5+a46*x6+ b41*u;
x5n=a51*x1+a52*x2+a53*x3+a54*x4+a55*x5+a56*x6+ b51*u;
x6n=a61*x1+a62*x2+a63*x3+a64*x4+a65*x5+a66*x6+ b61*u;
x1=x1n;x2=x2n;x3=x3n;x4=x4n;x5=x5n;x6=x6n;
```

```
//Kalman measurement update x=x+K*(y-C*x);
inov=y-(c11*x1+c12*x2+c13*x3+c14*x4+c15*x5+c16*x6);
x1=x1+k11*inov;
x2=x2+k21*inov;
x3=x3+k31*inov;
x4=x4+k41*inov;
x5=x5+k51*inov;
x6=x6+k61*inov;

//output estimation y=C*x;
est=c11*x1+c12*x2+c13*x3+c14*x4+c15*x5+c16*x6;

        flag_usart = 1;
        GPIO_ResetBits(GPIOD, GPIO_Pin_13);
    }
        EXTI_ClearITPendingBit(EXTI_Line0);
}

//
void SPI1_IRQHandler(void){
//  while (SPI_I2S_GetFlagStatus(SPI1, SPI_I2S_FLAG_TXE) == RESET);
//  SPI_I2S_SendData(SPI1, 0x00);
//  while(SPI_I2S_GetFlagStatus(SPI1, SPI_I2S_FLAG_BSY) != RESET){}

}


//-------------------------------------------------------------
void TIM3_IRQHandler(void){
//-------------------------------------------------------------
    TIM_ClearITPendingBit(TIM3, TIM_IT_CC1);
    GPIO_ToggleBits(GPIOD, GPIO_Pin_15); //LED

        GPIO_SetBits(GPIOD, GPIO_Pin_9);    //initiates conversion
        GPIO_ResetBits(GPIOD, GPIO_Pin_9);

    capture = TIM_GetCapture1(TIM3);
  TIM_SetCompare1(TIM3, capture + CCR1_Val);
}
```

4 Statistics

The relevant, most frequently used words and phrasis with the corresponding frequencies are:

sensor/s	194+140
noise	165
transfer function	98
frequency/frequencies	85+27
actuator	85
measurement	76
voltage	69
bandwidth	68
displacement	60
Kalman filter	59
resolution	56
⋮	
fusion	49

Most of the words begins with an *s* (331) followed by the initial *c* (303). The letter frequency in Chapters 1-3 (in sum 135831 letters) is compared to that in English (statistics taken from Concise Oxford Dictionary (11th edition revised, 2004)).