

MEMS scanner driver

for CompOCT

Division: HuCE - optoLab
Author: Daniel Tschupp
Date: 13. July 2020
Advisor: Ch. Meier

Abstract

Optical coherence tomography (OCT) is a image measurement method to generate 3d images. The optical principle allows to measure through a translucent material along the optical axis. Each change of refractive index delivers a signal. To generate a 3d image this is repeated at different points on the sample. Therefore a scanner deflects the measurement laser beam to different locations. Often this is done by galvanometers, where a mirror sits on the axis of a motor for each axis. Even if they are small the amount of power needed to drive those motors makes the electronics and power supplies big. Hence the HuCE-optoLab wants to miniaturize the scanner by using a Micro-Electro-Mechanical System (MEMS) mirror. Another advantage of MEMS mirrors are that the pivot points of both axis are at the same location. But the difficulty lies in driving them linearly because of their strong resonances.

This work elaborates an open-loop approach to solve this problem in three steps. First, the mirror was characterized by measuring the amplitude response. Second, a State-Space description model was developed. And last, an infinite impulse response (IIR) filter was generated using the method of pole-zero cancellation.

*To test the IIR filter a driver electronics prototype was developed as well as a microcontroller unit (MCU) software that handles communication and signal generation by implementing the IIR filter. The whole bundle of electronics and mechanics became the size of about lxbxh: 35*25*20mm which is considerable smaller than that of a galvanometer. The measurements showed that the open-loop approach is feasible until about one forth of the resonance frequency of the MEMS. Going beyond that limit, it does not get unstable or hazardous but it becomes non-utilizable for OCT applications as the distortion through resonances get to big.*



Table of Content

Abstract	I
1. Introduction	1
1.1. Context	1
1.2. Goals and Scope	3
1.3. Method	3
2. MEMS mirror characterization	4
2.1. State-Space model derivation	4
2.2. Bodeplot, PZ-Plot and Step Response of Model	6
3. Driving signal generation	7
3.1. The challenge of sawtooth signals	7
3.2. Deriving IIR filter	9
3.2.1. Discretization of MEMS	10
3.2.2. Elaboration of IIR	10
3.3. Implementation of IIR-Filter	12
3.4. Simulations of IIR filter	13
4. Hardware	14
4.1. Electrical Concept	14
4.2. MEMS driving electronics	15
4.2.1. Signal path	15
4.2.2. Power Supply	22
4.3. Controller PCB	23
4.4. Mechanical Concept	25
5. MCU Software	26
5.1. Firmware Architecture	26
5.1.1. Signal Generation	27
5.1.2. Communication	28
6. MEMS performance	29
6.1. Measurement setup	29
6.2. MEMS mirror movement	30
6.3. Limits of open-loop control	31
7. Conclusion: Magnetic MEMS scan head	32
8. Outlook	33
8.1. Possible future activities	33
8.2. Feedback Control	33
8.2.1. Bode integral	33
8.2.2. Controllability and Observability	34
8.2.3. Feedback control with observer	35
A. Sercalo MEMS data	A1
A.1. Data sheet	A1
A.2. Measurement Data	A1
A.3. Matlab Code	A2

B. Discretization Methods	B1
C. CompOCT	C1
D. Circuits	D1
D.1. Heat dissipation of IRF7309	D1
D.2. USB-C	D3
D.3. Load-Switches	D3
E. Communication protocol	E1
E.1. Typical communication flow	E1
E.2. Commands	E2

1. Introduction

1.1. Context

Optical coherence tomography is a 3d image generation technique that measures a sample and generates a representative image. The core technology enables measurement along the measurement-axis inside a translucent material. This is called an A-Scan. If such a material is translucent depends on the chosen wavelength. As an example, human tissue works with wavelengths around $1\mu m$. To generate a 3d image multiple A-Scans are recorded. This needs a scanner that is able to control the light beam like shown in the figure 1.1.

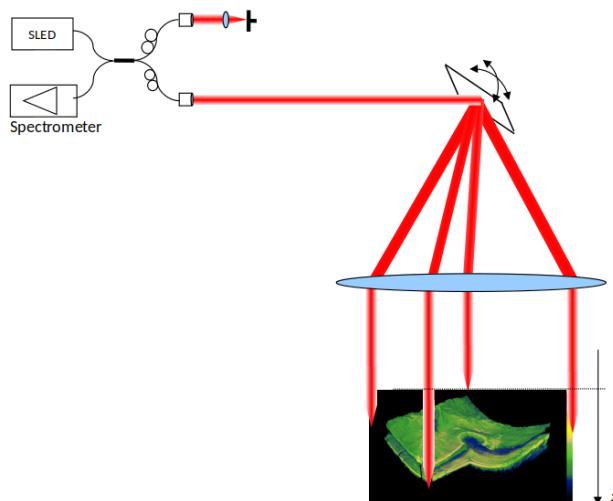


Fig. 1.1.: Schematic of 2d Scanner module.

The standard to realize such a scanner is by using galvanometers. One disadvantage of those is that they need bulky driving electronics. To miniaturize the scanner the HuCE-optolab tries another approach with a MEMS mirror. The first prototype with a electrostatic MEMS mirror from Mirrorcle turned out to be not very robust against dust. In this project a magnetic MEMS mirror from Sercalo will be tested. This should solve the issue. Another disadvantage of galvanometers is that their pivot points of the different axis are not at the same location. This amplifies second order aberrations from the lens of the scanner arm.

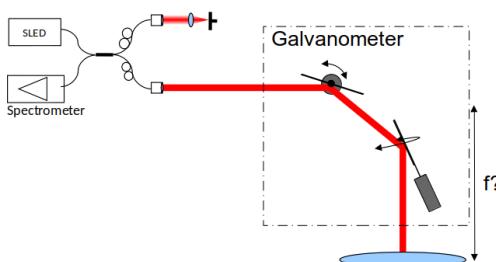


Fig. 1.2.: Galvanometer: Focal plane of lens can't be on both mirrors.

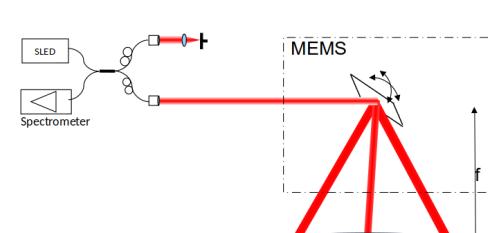


Fig. 1.3.: MEMS: Only one focal plane for the lens.

The magnetic MEMS is a current driven device. This needs to be considered in the design of the driver electronics. To be modular the hardware is split into two parts. One which is the intelligence that implements the driver algorithm and provides flexible and robust configuration possibilities. A STM32F4 MCU is used to achieve this. The other part provides the power supply and amplifiers to actually drive the MEMS. As a result the system architecture is as described by figure 1.4.

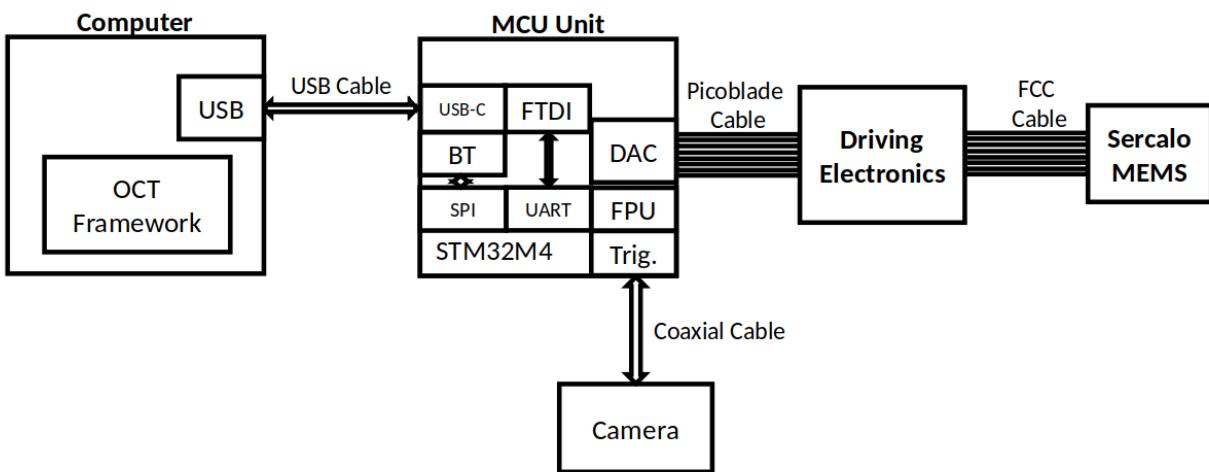


Fig. 1.4.: Blackbox diagram of the scanner module system.

The development of this system roughly followed these steps:

- Charakterization of MEMS mirror.
- Developement of open-loop control strategy.
- Simulation of control strategy.
- Hardware concept for MEMS scanner.
- Development of driving electronics.
- Development of MCU board.
- Implementation of Signal generation, communication and triggering on the STM32F4 MCU.
- Perform measurements.

1.2. Goals and Scope

To record a 3d image, called C-Scan, one stitches multiple 2d images, called B-Scans, together. To get a B-Scan one records multiple A-Scans along a line. For the scanner this means there is a fast axis along the B-Scan and a slow axis perpendicular to the B-Scan axis. A typical scan pattern looks like figure 1.5.

This project focuses on the signal generation, simulation and measurements. The developed MCU software and its documentation can be found on Gitlab[?]. The circuits can be found in the appendix D.

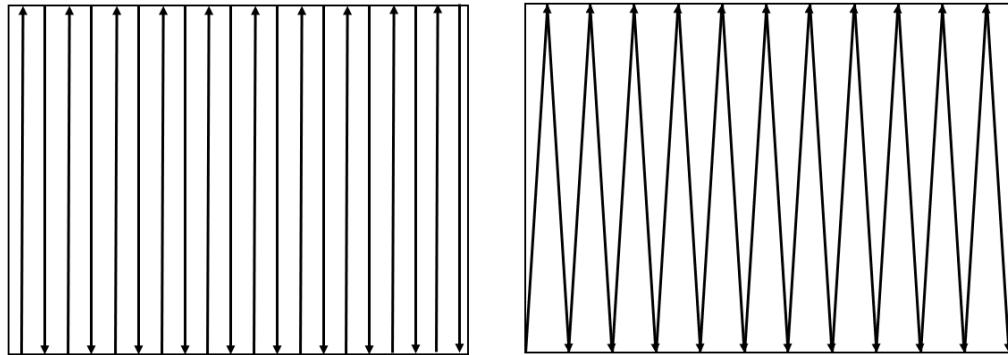


Fig. 1.5.: Ideal Scan pattern on the left and typical scan pattern on the right.

Within a B-Scan it is important that the measurement points are equally spaced. This means the mirror must move with a constant speed. So the waveform for mirror movements must be triangular. The goal of this work is to determine the driving signal needed to ensure that the mirror behaves like that.

1.3. Method

In a first step the mirror was characterized. To do so the amplitude response was measured manually. With this measurements a state-space representation was created to perform experiments on. Through a process called pole-zero cancellation a counteracting IIR filter was derived using Matlab. Additionally it was used to simulate the mirror movements. When simulations worked out the IIR filter was implemented in C++ on a MCU and a prototype of a driving electronics has been developed. With those 3 components it's possible to prove the concept by making measurements.

2. MEMS mirror characterization

In order to simulate the mirror movement an accurate model of the mirror is crucial. The following explains how this model was obtained. The data used to derive all the results, as well as the matlab code can be found in the appendix A.

2.1. State-Space model derivation

The basic physics behind the magnetic MEMS belongs to two domains, shown in figure 2.1. One is electrical and the other is mechanical. Those two domains are coupled together and build the functionality of the device. The state-space (SS) model represents the differential equations of both domains. These are based on Kirchhoff's and Newton's law. It's a slightly adopted version of a DC motor model.

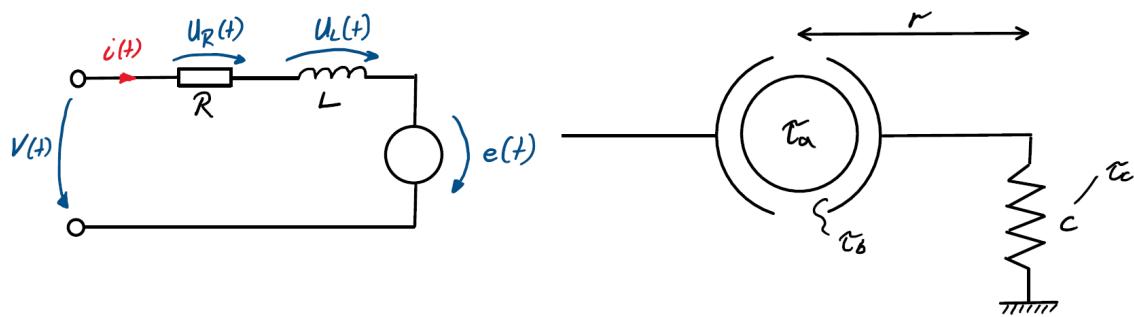


Fig. 2.1.: Electrical component on the left side and mechanical component on the right.

Faraday's Law:

$$\begin{aligned} e(t) &= -\frac{\partial \Phi_B}{\partial t} \\ \Phi_B &\propto \dot{\Theta} \\ \Rightarrow e(t) &= K * \dot{\Theta} = K * \omega \end{aligned}$$

Φ_B :	Magnetic flux
$\dot{\Theta}$:	Angle of mirror
P_{el} :	Electrical power
P_{mech} :	Mechanical power
ω :	Angular velocity of mirror
τ :	Generated torque
e :	Induced voltage
i :	Applied current
K :	Motor constant

Energy conservation law:

$$\begin{aligned} P_{el} &= P_{mech} \\ e * i &= \tau * \omega \\ K * \omega * i &= \tau * \omega \\ \Rightarrow \tau &= K * i \end{aligned}$$

Kirchhoff's Law:

$$\begin{aligned} \sum_i^N U_i = 0 &= U_R(t) + U_L(t) + e(t) - g * V(t) \\ 0 &= i(t) * R + \frac{\partial i}{\partial t} * L + K * \frac{\partial \Theta(t)}{\partial t} - g * V(t) \\ \Rightarrow i &= -\frac{K}{L} \dot{\Theta} - \frac{R}{L} i + \frac{g}{L} V \end{aligned}$$

$U_R(t)$:	Voltage over Resistor R
R :	Resistor value
$U_L(t)$:	Voltage over Coil L
L :	Coil inductance value
$e(t)$:	Induced voltage
g :	Scalar factor of electronics driver circuit.
Θ :	Angle of magnetic Mirror.
$i(t)$:	Current

Newton's Law:

$$\begin{aligned} \sum_i^N \tau_i &= J * \frac{\partial^2 \Theta(t)}{\partial^2 t} = \tau_a(t) - \tau_b(t) - \tau_c(t) \\ \frac{\partial^2 \Theta(t)}{\partial^2 t} &= \frac{\tau_a(t)}{J} - \frac{\tau_b(t)}{J} - \frac{\tau_c(t)}{J} \\ \Rightarrow \ddot{\Theta} &= \frac{K}{J} i - \frac{bf}{J} \dot{\Theta} - \frac{rc}{J} \Theta \end{aligned}$$

τ_a :	Moment induced by the driving current
τ_b :	Moment induced by friction
τ_c :	Moment induced by the spring
Θ :	Angle of mirror
$i(t)$:	Current
J :	Moment of inertia of the rotor and mirror.
bf :	Friction coefficient.
t :	Time

Defining the State vector \vec{x} :

$$\begin{aligned} \vec{x} &:= \begin{pmatrix} \dot{\Theta} \\ \Theta \\ i \end{pmatrix} \\ \Rightarrow \vec{x} &= \begin{pmatrix} \ddot{\Theta} \\ \dot{\Theta} \\ i \end{pmatrix} \end{aligned}$$

Defining the State-Space Matrices:

$$\begin{aligned} \vec{x} &= A\vec{x} + B\vec{u} \\ y &= C\vec{x} + D\vec{u} \end{aligned}$$

$$\begin{aligned} A &:= \begin{bmatrix} -\frac{bf}{J} & -\frac{rc}{J} & \frac{K}{J} \\ 1 & 0 & 0 \\ -\frac{K}{L} & 0 & -\frac{R}{L} \end{bmatrix} \\ B &:= \begin{bmatrix} 0 \\ 0 \\ \frac{g}{L} \end{bmatrix} \\ C &:= [0 \ 1 \ 0] \\ D &:= [0] \end{aligned}$$

2.2. Bodeplot, PZ-Plot and Step Response of Model

The model from the previous section was fitted to the measurements by least square error optimization criteria. The optimization parameters are the spring constant c for the dynamic and input gain g for the static behavior of the systems.

The resulting model can be seen in the first plot of the figure below.

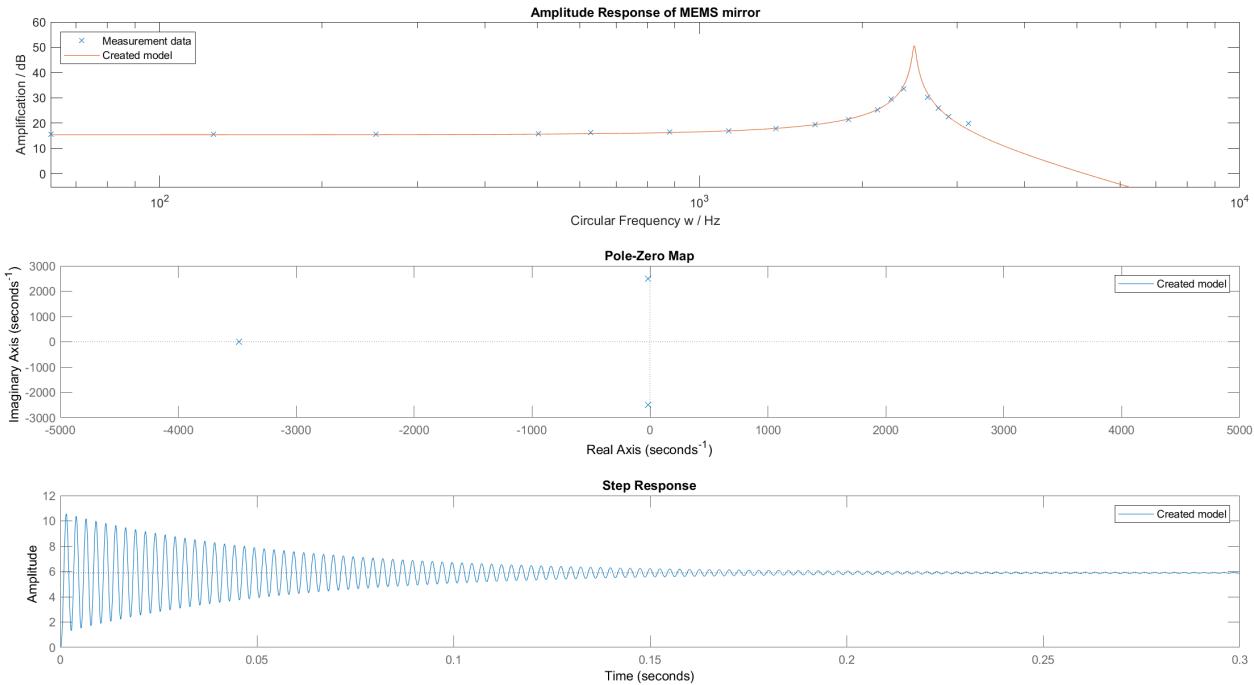


Fig. 2.2.: 1. Bode plot, 2. Pole-Zero plot, 3. Step response of SS model.

Analyzing Pole-Zero plot concludes that the system is in principle stable which means it is open-loop controllable. But the complex conjugated pole-pair is quite near to 0, that's why the step response has a very long transient effect.

3. Driving signal generation

As explained in the introduction the mirror movement we like to have is a triangular one. This chapter is all about how to achieve this.

First is explained why this is difficult. And afterwards one solution will be presented.

3.1. The challenge of sawtooth signals

A triangular function has a Fourier series like described below[?]:

$$f(x) = \frac{8}{\pi^2} \sum_{n=1,3,5,\dots}^{\infty} \frac{(-1)^{(n-1)/2}}{n^2} \sin\left(\frac{n\pi x}{L}\right)$$

The equation above states that a triangle signal consists of its fundamental sine frequency and all the odd harmonic sine frequencies. The higher the harmonic frequencies, the less impact it has as its amplitude decreases quadratically. The following simulations (fig. 3.1 to 3.4) show the impact of those harmonic waves on MEMS mirror movements.

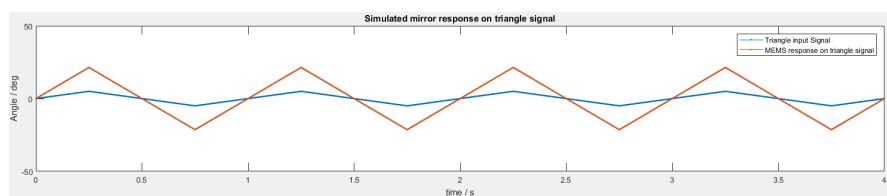


Fig. 3.1.: 1Hz triangle signal at MEMS mirror.

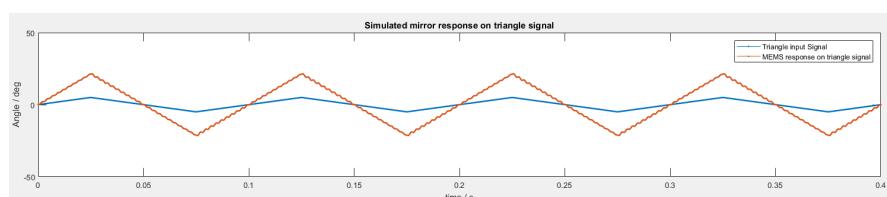


Fig. 3.2.: 10Hz triangle signal at MEMS mirror.

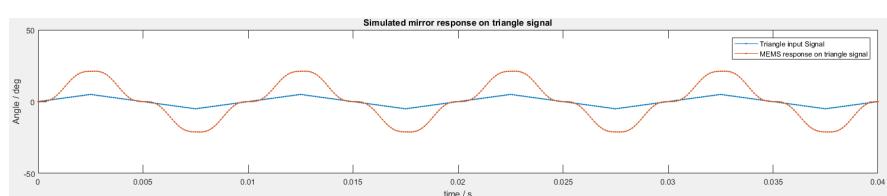


Fig. 3.3.: 100Hz triangle signal at MEMS mirror.

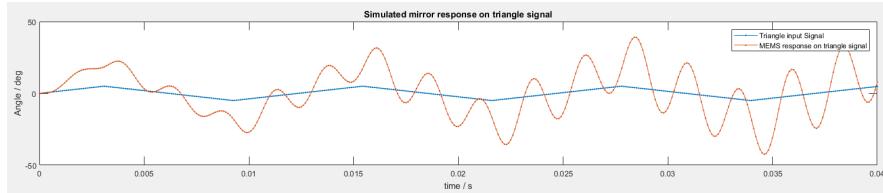


Fig. 3.4.: 81Hz triangle signal at MEMS mirror.

The simulate shows clearly that when operating at slow frequencies this works fine like in figure 3.1. But when the frequency is increased, the harmonic waves appear. In the case of figure 3.2 it's a high frequency harmonic wave with a small amplitude. In the case of 3.3 it's the first harmonic wave but it doesn't directly hit the resonance frequency of the mirror. Figure 3.4 is one of the worst case scenarios. There the third harmonic wave($81 \times 5 = 405$) directly hits the resonance frequency of the mirror. That's why this signal isn't usable anymore. Figures 3.5 to 3.10 illustrates the problem further by inspecting the power spectrum of a triangular wave and how the transfer function influences it. The six following graphs are arranged that input signals are on the left and MEMS responses are on the right.

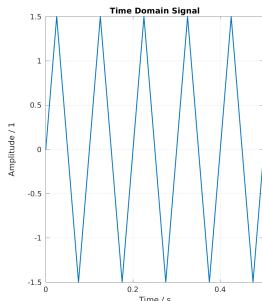


Fig. 3.5.: Left: 10 Hz Sawtooth signal.
Right: Power-Spectrum + MEMS Transfer function.

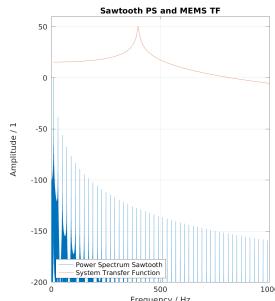


Fig. 3.6.: Left: Signal response
Right: Power-Spectrum response

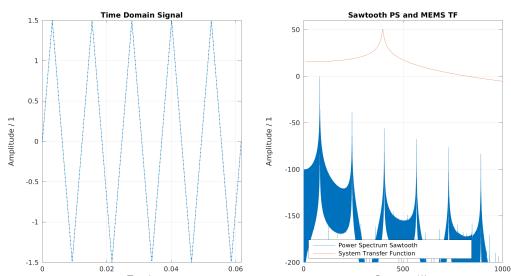


Fig. 3.7.: Left: 81 Hz Sawtooth signal.
Right: Power-Spectrum + MEMS Transfer function.

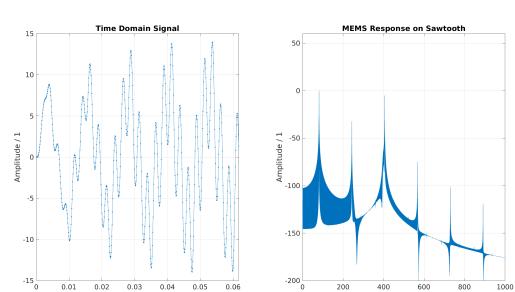


Fig. 3.8.: Left: Signal response
Right: Power-Spectrum response

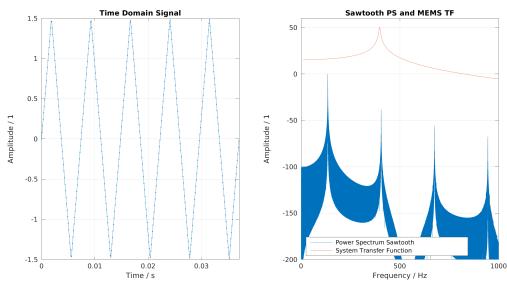


Fig. 3.9.: Left: 135 Hz Sawtooth signal.
Right: Power-Spectrum + MEMS Transfer function.

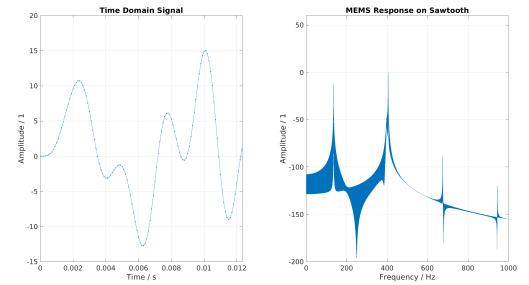


Fig. 3.10.: Left: Signal response
Right: Power-Spectrum response

Now it's clear that the driving signal must be manipulated to suppress the resonance frequency of the MEMS mirror. The next section will present a solution for this.

3.2. Deriving IIR filter

The idea behind this method is to suppress each pole in the mirror frequency response with a corresponding zero. Therefore a precise model of the MEMS is crucial, which means every mirror needs to be calibrated separately. As the signal is generated from the MCU which is time discrete one has to convert the transfer function to the z-domain. Multiple conversion methods have been investigated and the impulse response matching method was taken. Appendix B shows the different methods. Impulse response matching was chosen because it ensures stability after conversion as long as the continuous time system has been stable.

The derived state-space model from the chapter 1.3 can be analysed by a pole-zero plot. Figure 3.11 is such a plot.

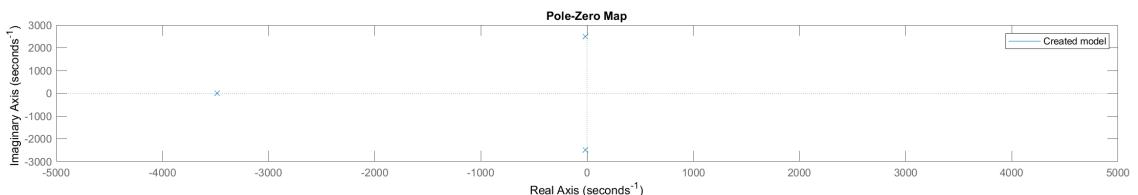


Fig. 3.11.: Continuous time pole-zero map

3.2.1. Discretization of MEMS

After the Impulse Response Matching the MEMS Pole-Zero map looks as follows (fig: 3.12):

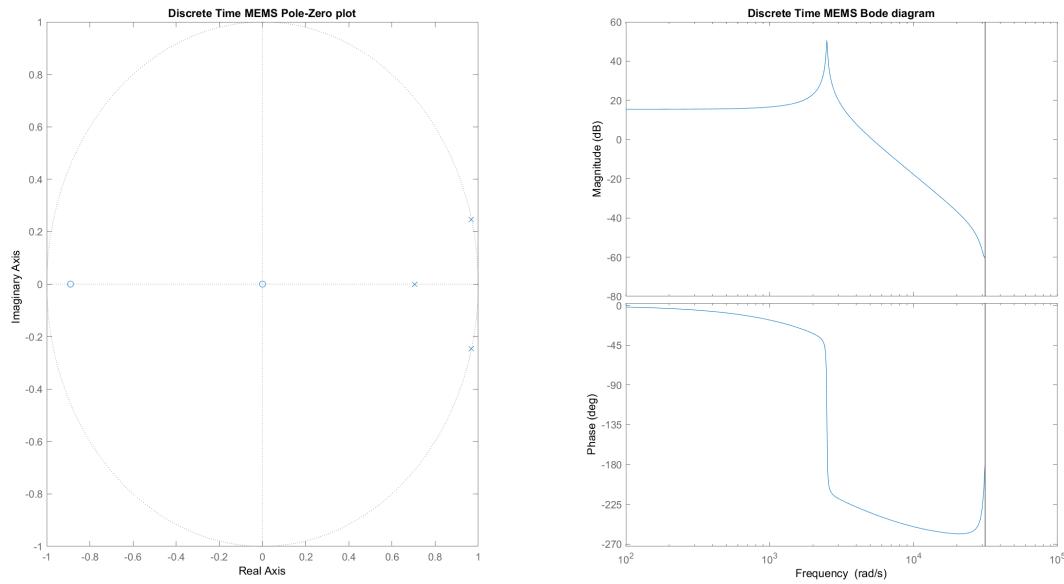


Fig. 3.12.: Discrete Time Bode Diagram and Pole-Zero map

3.2.2. Elaboration of IIR

The driving signal generator implements the reciprocal transfer function of the discrete time MEMS system. This can only be done when not only the poles but also the zeros are inside the unit circle because zeros become poles and vice-versa. Figure 3.13 shows the controller pole-zero plot.

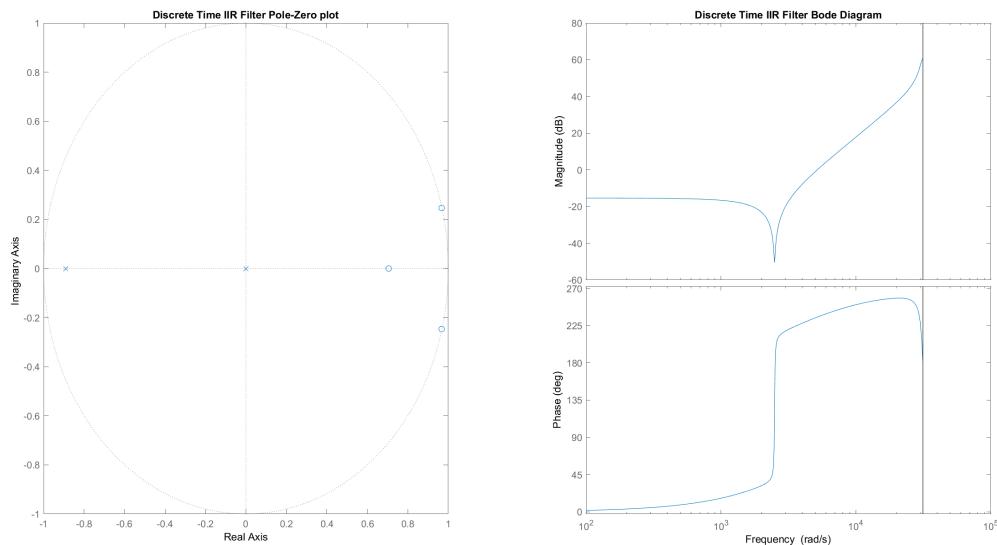


Fig. 3.13.: Discrete Time Bode Diagram and Pole-Zero map

This IIR-Filter still has two problems. Firstly there are more zeros than poles. This makes the system non-causal. This means that the filter would have to foresee the future. This could be tackled by artificially delaying the signal. Secondly the gain of the high frequencies are very high which means that noise will be amplified heavily.

Both problems can be solved by introducing additional low-pass filtering. Three first order LP-filter have been implemented at the following frequencies: $w_1 = 6\text{kHz}$, $w_2 = 10\text{kHz}$, $w_3 = 10\text{kHz}$. The resulting Bode diagram and Pole-Zero map can be examined in figure 3.14.

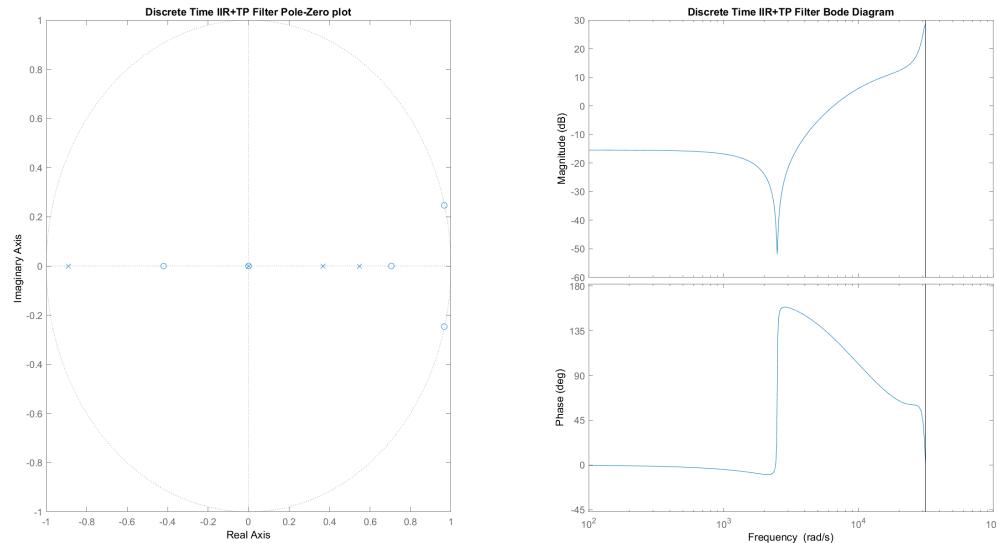


Fig. 3.14.: Discrete Time Bode Diagram and Pole-Zero map

The resulting coefficients for the IIR-Filter are:

$$\begin{array}{lllll} a_0 \dots a_4: & -1.0000 & 0.3948 & 0.6038 & -0.4054 & 0.0661 \\ b_0 \dots b_4: & 2.2292 & -4.9486 & 2.7908 & 0.6451 & -0.6588 \end{array}$$

3.3. Implementaion of IIR-Filter

The filter has been implemented in the direct form 2 with merged memory. Figure 3.15 illustrates this implementation schematically.

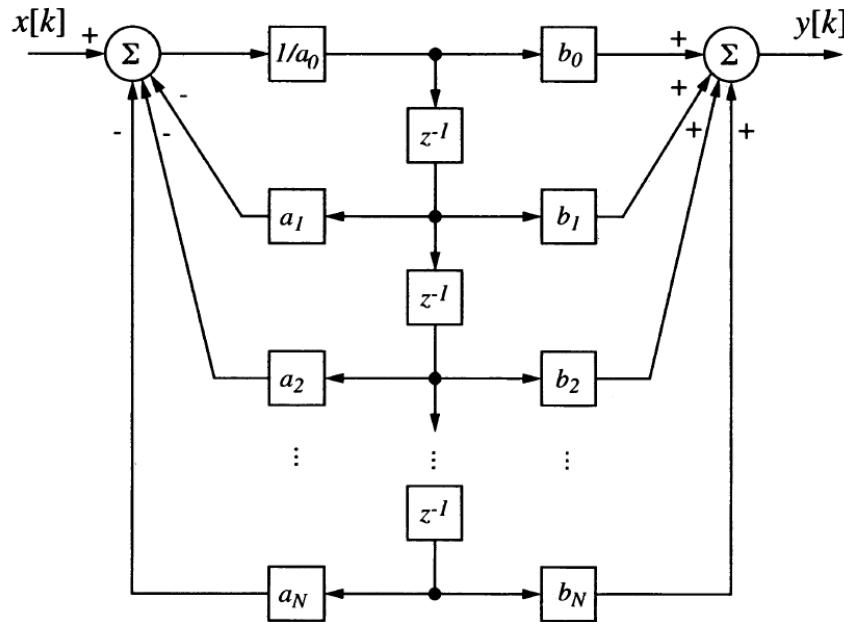


Fig. 3.15.: Transposed direct form 2 of a IIR-filter. [?, Systemtheorie p.334]

This structure was chosen to minimize the memory consumption as well as the number of calculations. As the MCU does come with a floating point unit (FPU) there are no coefficients quantization problems. But the calculations do take longer than with fixed point implementation. With a clock speed of 168MHz it should be no problem to calculate a sequence with 10kHz speed.[?, Systemtheorie p.334]

3.4. Simulations of IIR filter

The performance of the driving signal generator was simulated with the same frequencies as in chapter 3.1. Figure 3.16 to 3.21 display the results.

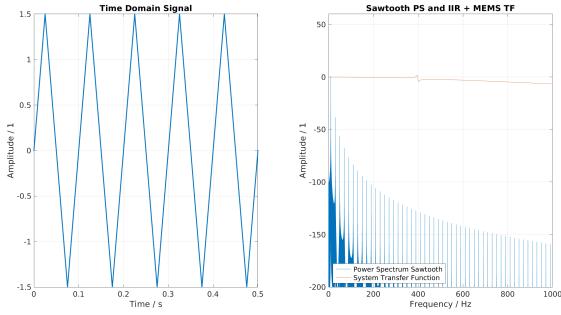


Fig. 3.16.: Left: 10 Hz Sawtooth signal.
Right: Power-Spectrum + Overall transfer function.

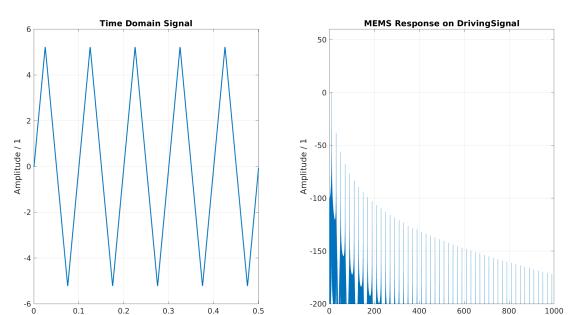


Fig. 3.17.: Left: Signal response
Right: Power-Spectrum response

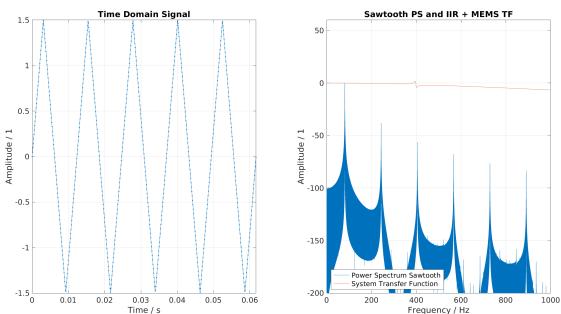


Fig. 3.18.: Left: 81 Hz Sawtooth signal.
Right: Power-Spectrum + Overall transfer function.

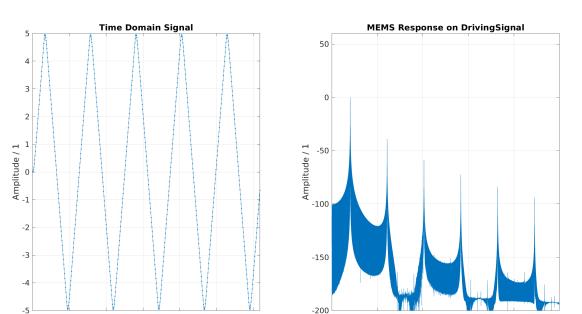


Fig. 3.19.: Left: Signal response
Right: Power-Spectrum response

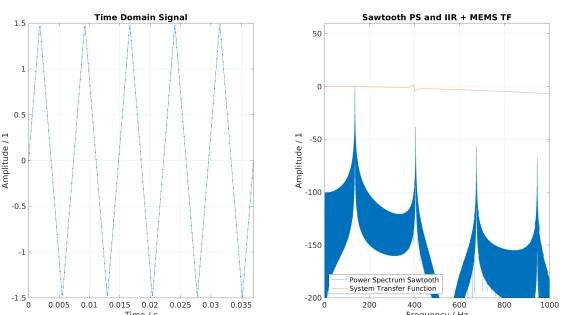


Fig. 3.20.: Left: 135 Hz Sawtooth signal.
Right: Power-Spectrum + Overall transfer function.

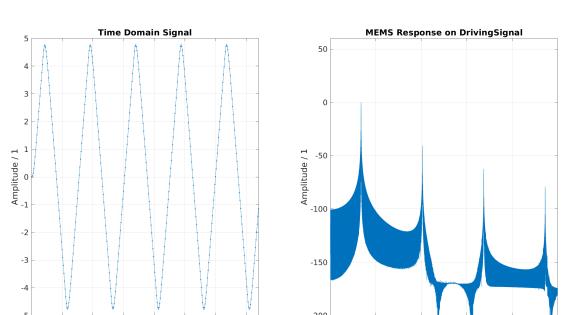


Fig. 3.21.: Left: Signal response
Right: Power-Spectrum response

Inspecting the power spectrum of the MEMS responses shows marginal differences to the one of the original triangle signal. So the resonances of the MEMS were successfully suppressed.

4. Hardware

4.1. Electrical Concept

Electrically the Scanner consists out of three printed circuit board (PCB)s and a MEMS mirror. The MCU-PCB provides intelligence and communication. The Amplifier-PCB provides galvanic isolation of the signal and the driving circuit to run the mirror. On top of it a isolating low noise power supply is mounted. This modularity allows to provide multiple Amplifier-PCBs for different mirrors. But from the operator side the communication and protocols stay the same. Figure 4.1 is a block diagram of this modular concept.

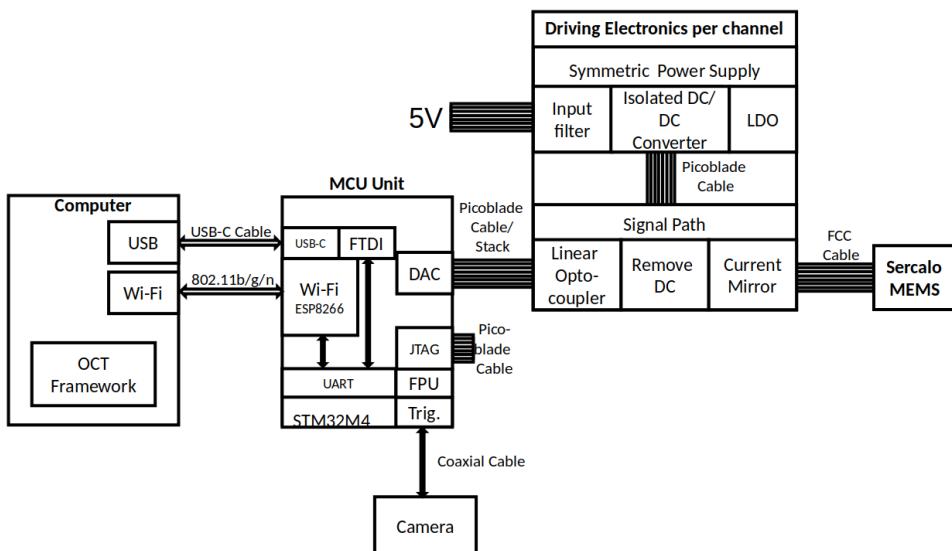


Fig. 4.1.: Scanner electronics concept

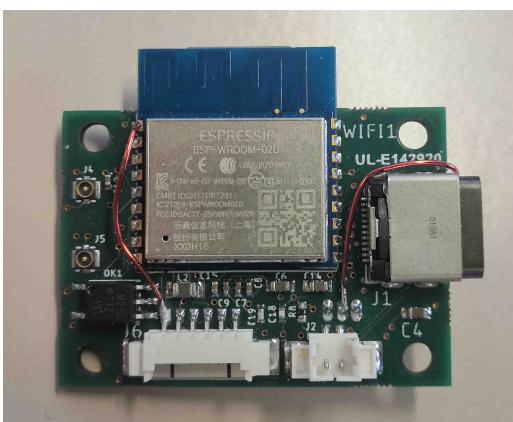


Fig. 4.2.: MCU PCB.

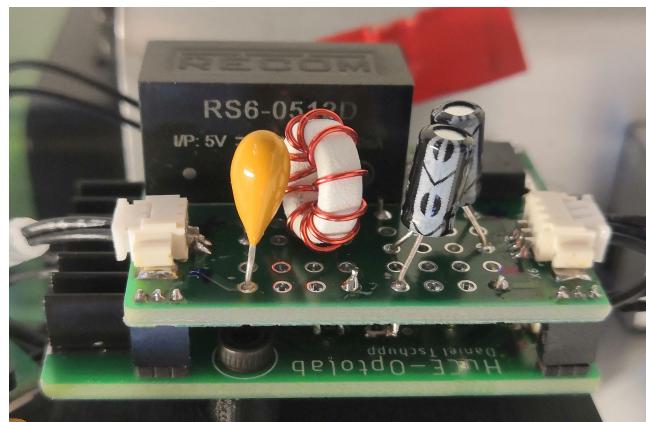


Fig. 4.3.: Driver Electronics PCBs stacked together.

4.2. MEMS driving electronics

4.2.1. Signal path

As the power supply is isolated, it's necessary to decouple the signal to bring it to the same potential. To do so a LOC110P optocoupler was used. This is the first stage of the circuit in figure 4.4. After decoupling the signal an difference amplifier removes the dc portion of the signal. It was more suitable than a filter because it must be possible to have a static deflection of the mirror. The last stage implements a current mirror that translates the signal into a current.

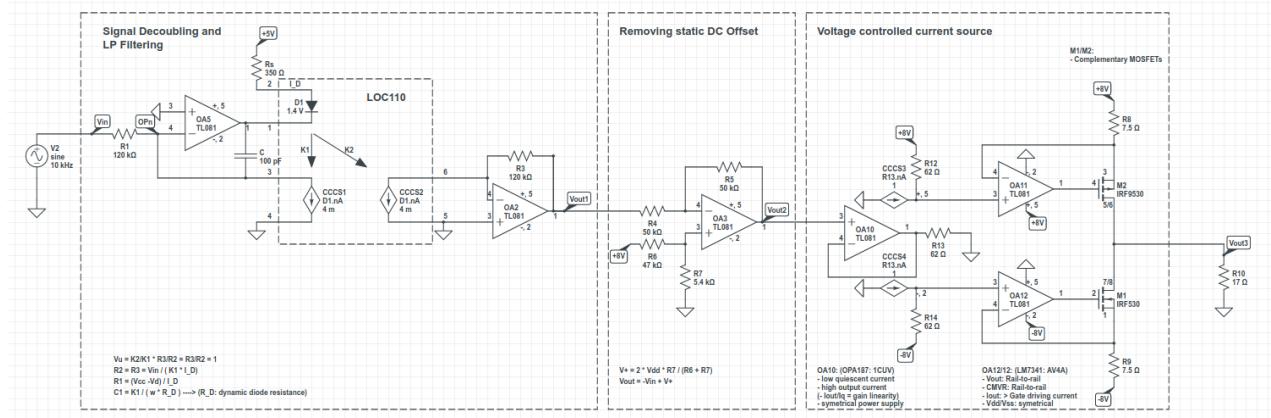


Fig. 4.4.: Whole signal path circuit.

The three parts of this circuit are discussed on the following pages one by one.

The AC behaviour of the stages is analyzed by signal flow graph (SFG) and Mason's Gain formula which is:

$$G = \frac{Y(s)}{U(s)} = \frac{\sum_{i=1}^N P_i \Delta_i}{\Delta}$$

Description of Variables:

- $Y(s)$ Output node
- $U(s)$ Input node
- G Transfer function from $U(s)$ to $Y(s)$
- P_i i^{th} forward path from $U(s)$ to $Y(s)$
- Δ $= 1 - (\text{sum of all individual loop gains})$
 - $+ (\text{sum of gain products of all possible two nontouching loops})$
 - $- (\text{sum of gains products of all possible three nontouching loops})$
 - $+ \dots$
- Δ_i Can be obtained from Δ by removing all the loops touched by the i^{th} forward path.

Decoupling of signal:

To analyze this stage a AC equivalent circuit (fig: 4.5) was developed to generate SFG.

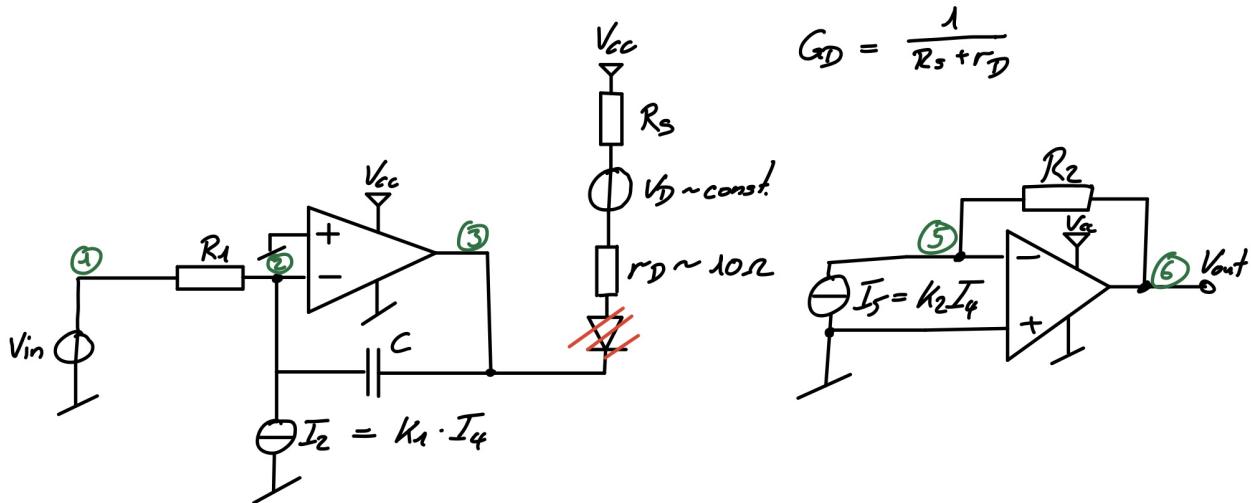


Fig. 4.5.: AC equivalent circuit of decoupling circuit.

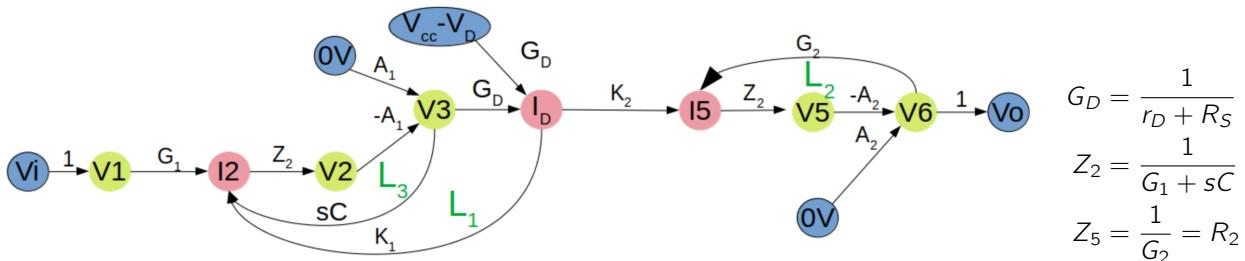


Fig. 4.6.: SFG of decoupling circuit.

Analysis by Mason's gain rule:

$$\begin{aligned}
 G_1 &:= \frac{V_{out}}{V_{in}} = \frac{\sum P_i \Delta_i}{\Delta} \\
 &= \frac{P_1}{1 - [L_1 + L_3 + L_2] + [L_1 L_2 + L_2 L_3]} \\
 &= \frac{G_1 Z_2 (-A_1) G_D K_2 (-A_2) R_2}{1 - [-A_1 Z_2 K_1 G_D - A_2 G_2 R_2 - A_1 Z_2 sC] + [A_1 A_2 R_2 G_2 G_D K_1 Z_2 + A_1 A_2 G_2 R_2 sC Z_2]} \\
 \lim_{A_1, A_2 \rightarrow \infty} (G_1) &= \frac{G_1 Z_2 G_D K_2 R_2}{R_2 G_2 G_D K_1 Z_2 + G_2 R_2 sC Z_2} \\
 &= \frac{R_2 K_2}{R_1 K_1 + R_1 sC (r_D + R_S)} \\
 \lim_{A_1, A_2 \rightarrow \infty, K_1 \rightarrow K_2, w \rightarrow 0} (G_1) &= \underline{\underline{\frac{R_2}{R_1}}}
 \end{aligned}$$

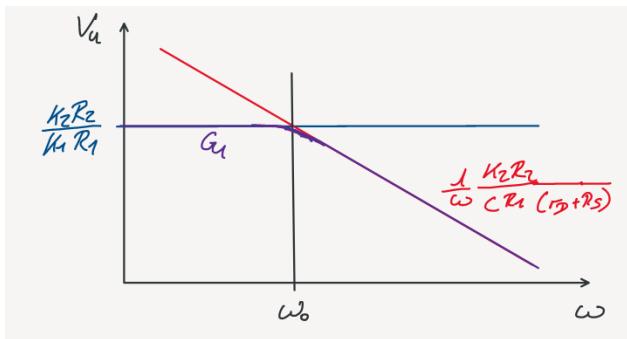


Fig. 4.7.: Estimated Bode diagram of decoupling circuit.

$$\begin{aligned} \frac{K_2 R_2}{K_1 R_1} &= \frac{1}{2\pi f_0} \frac{K_2 R_2}{C R_1 (r_D + R_S)} \\ \Rightarrow C &= \frac{K_1}{2\pi f_0 (r_D + R_S)} \\ &= \frac{0.004}{2\pi \cdot 10\text{kHz} \cdot (10\Omega + 350\Omega)} \approx 180\text{pF} \Rightarrow \underline{\underline{150\text{pF}}} \end{aligned}$$

Design:

$$\begin{aligned} R_S &= \frac{V_{cc} - V_{Dmax}}{I_{Dmax}} = \frac{5V - 1.4V}{10mA} \approx 360\Omega \Rightarrow \underline{\underline{350\Omega}} \\ \text{Def : } I_D &\in \{2mA, 7mA\} \\ R_1 &= \frac{V_{in}}{I_{R_1}} = \frac{V_{in}}{I_D \cdot K_1} = \frac{3.3V}{7mA * 0.004} \approx \underline{\underline{120k\Omega}} \\ R_2 &= R_1 \end{aligned}$$

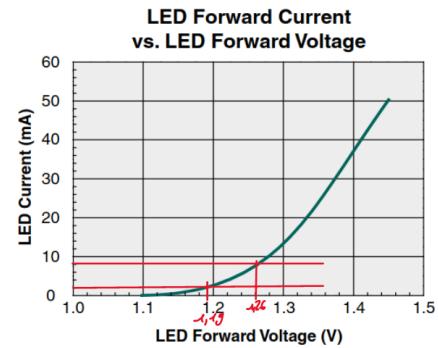


Fig. 4.8.: Diode characteristics of LOC110D.

Removing DC Part:

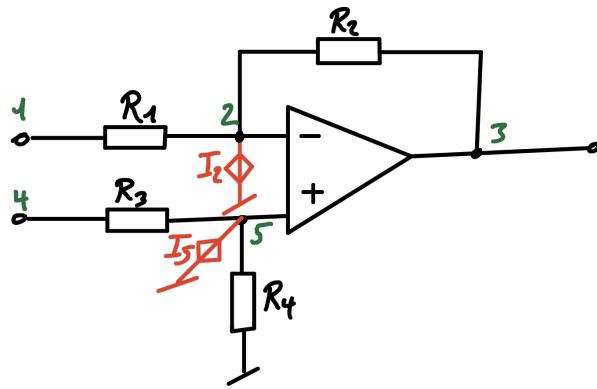


Fig. 4.9.: Circuit of DC removal amplifier.

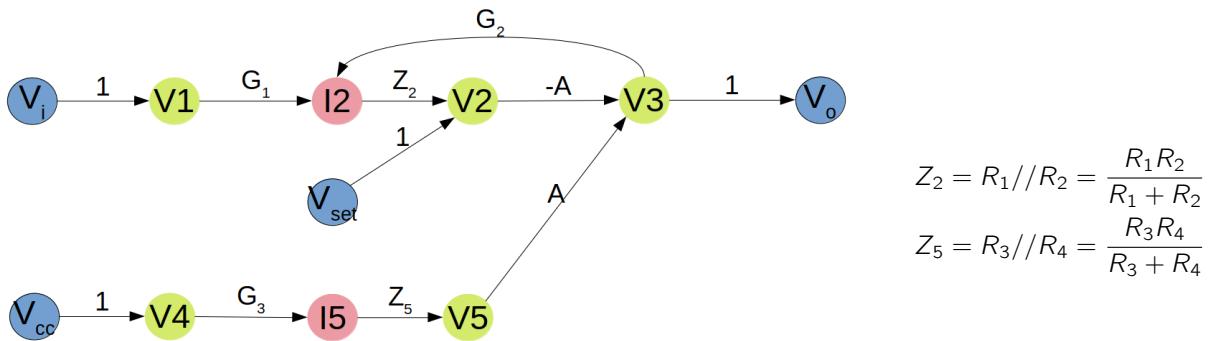


Fig. 4.10.: SFG of decoupling circuit.

AC Analysis with Mason's gain rule:

Design:

$$G_1 = \frac{V_o}{V_i} = \frac{-AG_1 Z_2}{1 - [-AG_2 Z_2]} = \frac{-G_1}{\frac{1}{A Z_2} + G_2}$$

$$\Rightarrow \lim_{A \rightarrow \infty} (G_1) = -\frac{G_1}{G_2} = -\frac{R_2}{R_1}$$

$$G_2 = \frac{V_o}{V_{set}} = \frac{-A}{1 + AG_2 Z_2} = -\frac{1}{\frac{1}{A} + G_2 Z_2}$$

$$\Rightarrow \lim_{A \rightarrow \infty} (G_2) = -\frac{1}{G_2 Z_2} = -\frac{R_1 + R_2}{R_1}$$

$$\Rightarrow \lim_{A \rightarrow \infty, R_1 \rightarrow R_2} (G_2) = -2$$

$$\Rightarrow V_{dc} = \underline{\underline{2V_{set}}}$$

$$\text{Def : } R_1 = R_2 = \underline{\underline{51k\Omega}}$$

$$\text{Def : } R_3 = \underline{\underline{47k\Omega}}$$

$$V_{set} = \frac{V_{cc}}{R_3 + R_4} R_4$$

$$\Rightarrow \underline{\underline{R_4}} = \frac{V_{dc}}{2V_{cc} - V_{dc}} R_3$$

$$= \frac{1.65V}{2 \cdot 8V - 1.65V} \cdot 47k\Omega \approx \underline{\underline{5.4k\Omega}}$$

Current Mirror:

As the magnetic MEMS mirror is driven by current it needs an amplifier circuit that controls the output current. The following current mirror circuit from Tietze/Schenk has been implemented. For this stage, it's not possible to do a SFG analysis because the MOSFET is a nonlinear component.

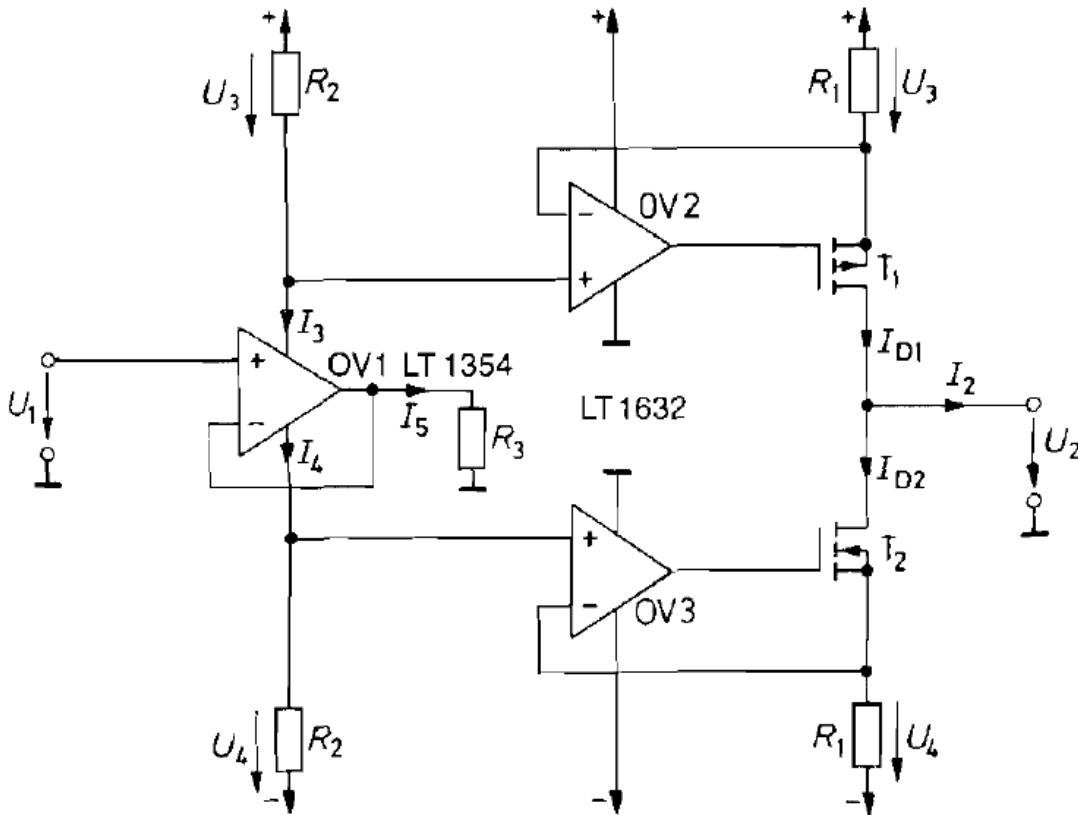


Fig. 4.11.: Voltage regulated current source.[?, p. 803, Tietze/Schenk]

Circuit Description:

This circuit implements a current mirror. It manages to do so with the first Operational amplifier (OpAmp) OV1. This translates the voltage U_1 into a current I_5 . OV1 operates as node, like Kirchhoff's current law describes. Therefore the current I_5 must be provided through the resistors R_2 . This generates, depending on the direction, the voltages U_3 or U_4 respectively. The OpAmp OV2/OV3 ensures the same voltage drop over R_1 and therefore defines the current flowing to the output. The supply of the OV2/OV3 ensures only one MOSFET may be active at a given time. In the following the equations to the circuit.[?, p. 803, Tietze/Schenk]

$$U_1 > 0$$

$$\Rightarrow I_3 = I_5 = \frac{U_1}{R_3}$$

$$R_2 = R_3 \Rightarrow U_1 = U_3 = U_{R_1}$$

$$\Rightarrow I_{D1} = I_2 = \frac{U_1}{R_1}$$

$$U_4 = 0 \Rightarrow I_{D2} = 0$$

$$U_1 < 0$$

$$\Rightarrow I_4 = -I_5 = -\frac{U_1}{R_3}$$

$$R_2 = R_3 \Rightarrow U_1 = U_4 = U_{R_1}$$

$$\Rightarrow I_{D2} = -I_2 = \frac{U_1}{R_1}$$

$$\Rightarrow U_3 = 0 \Rightarrow I_{D1} = 0$$

$$G_{cm} = \frac{I_2}{U_1} = \frac{1}{\underline{R_1}}$$

$$R_2 = R_3 = \frac{U_{1max}}{I_{OV1Outmax}}$$

$$Zerodrift = I_{OV1_{quiescence}}$$

$$\text{Linearity\%} = \Delta R_{3\%} + \Delta R_{2\%} + \Delta R_{1\%}$$

In order for this circuit to work properly to operational amplifiers must fulfill certain conditions:

OV1:

Supply Voltage:	>±12V
Supply Current:	Very small
Output Current:	High

OV2/OV3:

Supply Voltage:	±12V and Asymmetric
Output Voltage Swing:	Rail-To-Rail
Common Mode Voltage Range:	Rail-To-Rail

R1:

Power Rating:	Sufficient high
---------------	-----------------

T1/T2:

Power Rating:	Sufficient high
Leakage Current:	Small
R_{DS} :	Small
U_{DSmax} :	$>2 \cdot U_B$

Simulations:

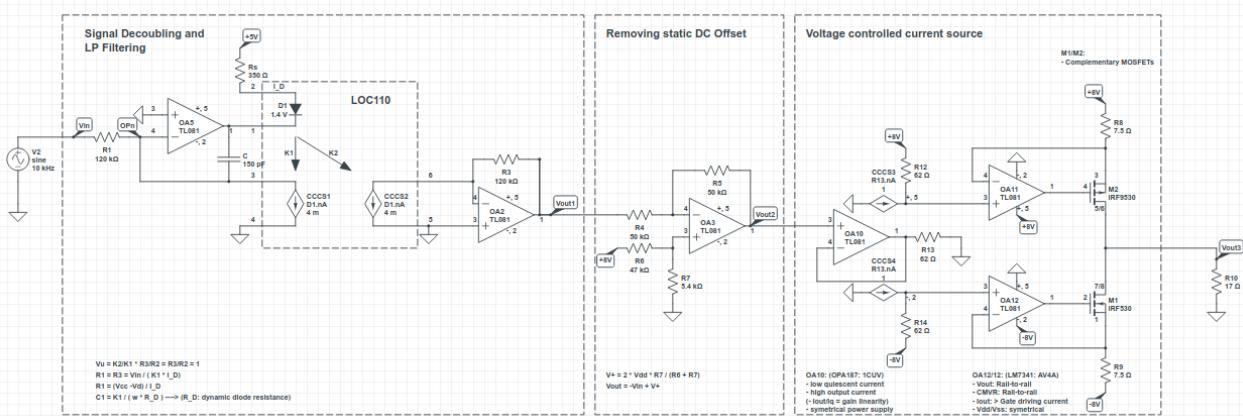


Fig. 4.12.: Simulation circuit.

Result: Bode Diagram

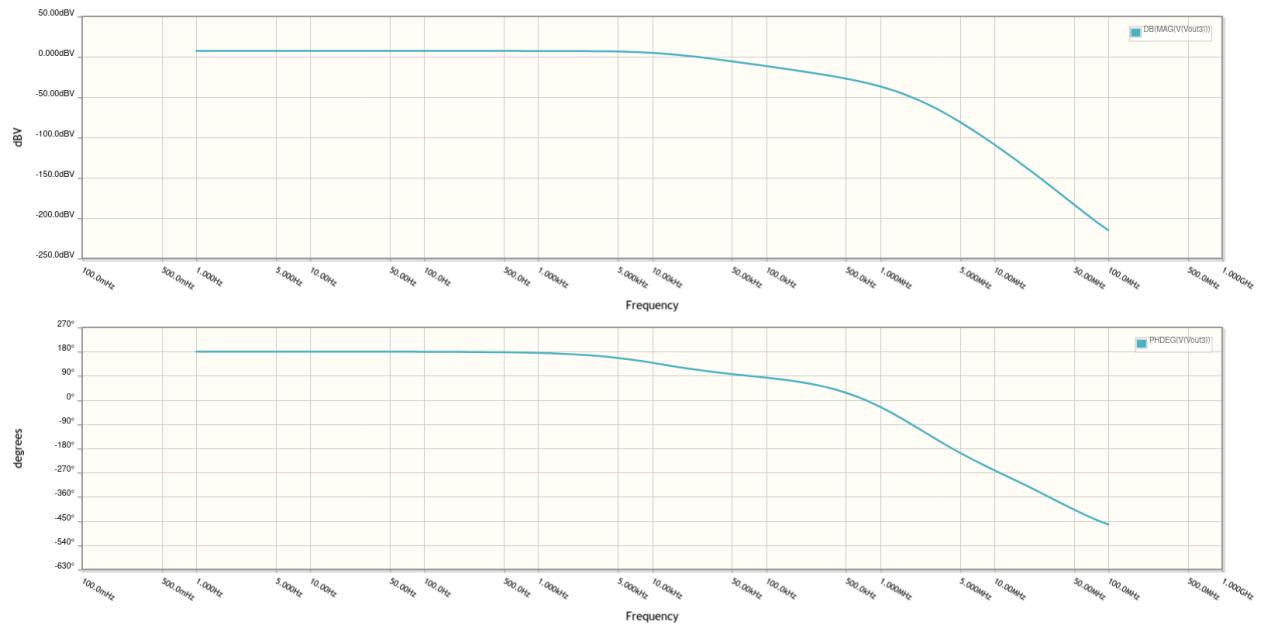


Fig. 4.13.: Frequency analysis of signal path

Result: Transient Analysis

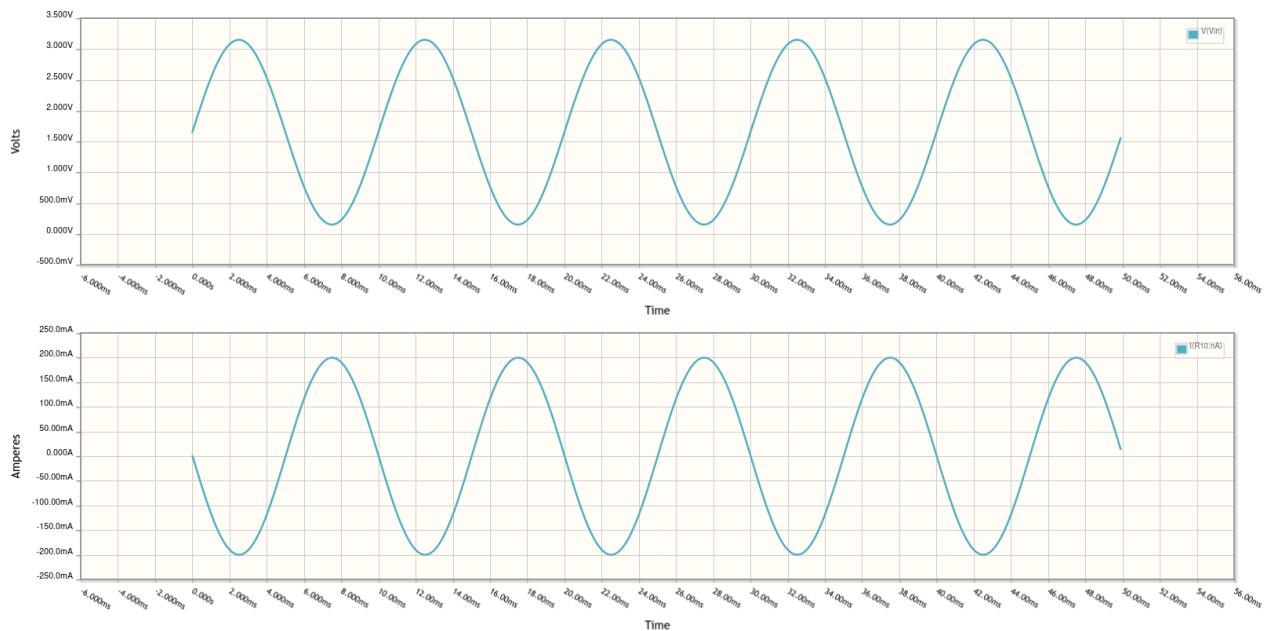


Fig. 4.14.: Transient analysis of signal path at 100Hz and full deflection.

4.2.2. Power Supply

For the power supply it's important to be able to deliver enough current ($\pm 250\text{mA}$) per channel and be as compact as possible. The choice was the RS6-0512D of Recom. This DC-DC converter operates at an internal operating frequency of 200kHz. This converter generates an isolated $\pm 12\text{V}$ supply voltages.

High stability and low noise are fundamental requirements for the power supply. It is necessary to guarantee the functionality of the current mirror circuit, as the supply voltages have direct influence on the load. Therefore low dropout regulator (LDO) are used to stabilize the $\pm 12\text{V}$ power supply and a $10\mu\text{F}$ (C_4/C_5)electrolyte capacitor provides additional power.

The input of the power supply is filtered by a LC low pass filter. As currents may get as high as $I_{max} = \frac{6W}{5V} = 1.2\text{A}$ the requirements for the coil is a saturation current of at least 1.2A.

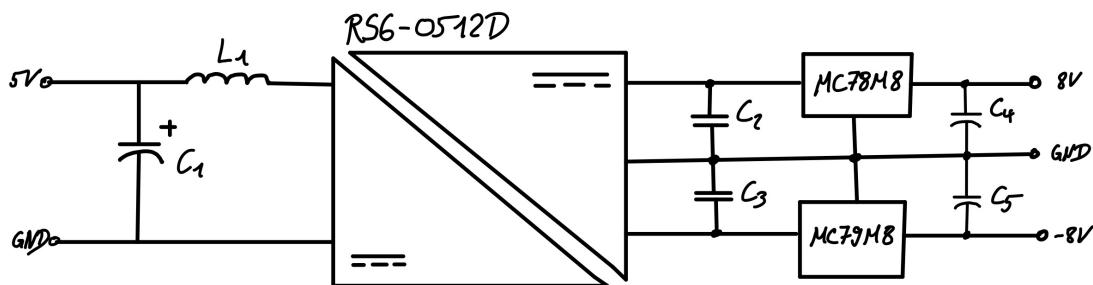


Fig. 4.15.: Power Supply Circuit per Channel.

C_2/C_3 are for optional equipment if necessary. Their value is as specified in the MCxx datasheet $1\mu\text{F}$. For the input filter, the coil is a self made ring coil.

Core specifications:

R :	$\emptyset_o = 9.5\text{mm}$
r :	$\emptyset_i = 5.4\text{mm}$
r_m :	$\frac{R+r}{2} = 7.45\text{mm}$
Length b:	3.4mm
Material:	4C65
f_{sat} :	$\approx 40\text{MHz}$
μ_r :	$125 \pm 20\%$
H_{sat} :	200A/m

Calculating max number of Windings:

$$N = \frac{H \cdot 2\pi r_m}{I_{max}} = \frac{100\text{A}/m \cdot 2\pi \cdot 7.45\text{mm}}{21.2\text{A}} \approx 8$$

Calculating inductance:

$$\begin{aligned} L &= N^2 \cdot \frac{\mu_0 \mu_r b}{2\pi} \ln \left(\frac{R}{r} \right) \\ &= 8 \cdot \frac{1.256637 \cdot 10^{-6} \frac{N}{\text{A}^2} \cdot 3.4\text{mm}}{2\pi} \ln \left(\frac{9.5\text{mm}}{5.4\text{mm}} \right) \approx 3\mu\text{H} \end{aligned}$$

Calculating capacity:

$$\begin{aligned} f_g &= f_{recom} \cdot 0.1 = 20\text{kHz} \\ C &= \frac{1}{(2\pi f_g)^2 L} \\ &= \frac{1}{(2\pi \cdot 20\text{kHz})^2 \cdot 3 \cdot 10^{-6}} \approx 22\mu\text{F} \end{aligned}$$

4.3. Controller PCB

The requirements for the controller PCB were the following:

- Able to calculate driving signal in real-time.
- Able to generate driving signal with digital to analog converter (DAC) or pulse-width modulation (PWM).
- Triggering signal generation.
- Communication via Universal serial bus (USB).
- Wireless communication.
- Powering through USB.
- Powering through external 5V power supply.
- Programming with serial wire debug (SWD).

Those requirements resulted in the design below:

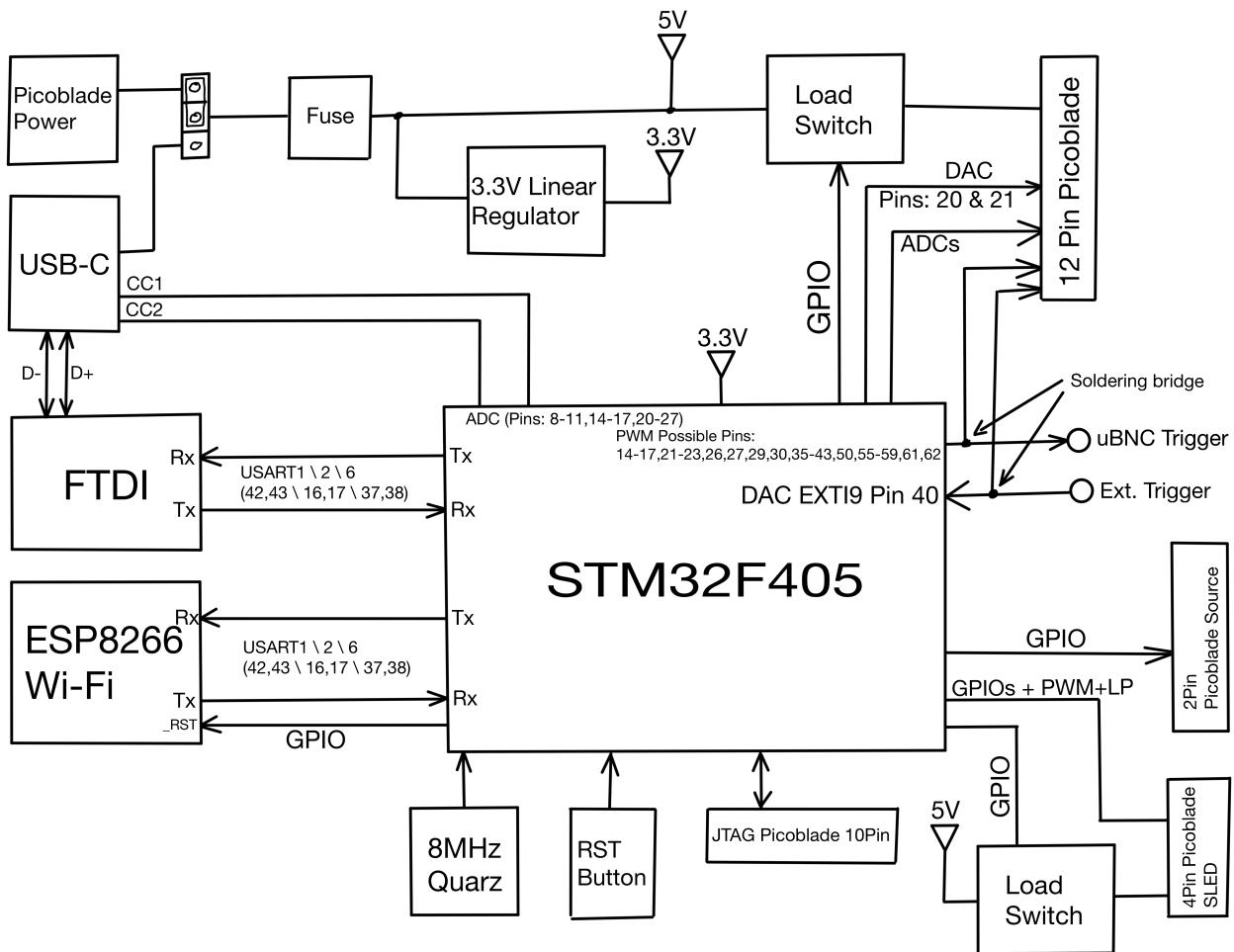


Fig. 4.16.: Planned PCB Layout.

Transmission line considerations for Trigger signals:

As the A-Scan trigger signal can be as high as the maximal camera speed a differential driver was considered for this signal.

The MCU has a minimal rise time for IO signal of 100ns in slow speed IO mode and 2.5ns in high speed IO mode [?, p. 117/118, STM32F4 datasheet].

Signal speed inside wire[?, p. 49, High-Speed Signal Propagation]:

$$v_{ph} = \frac{c}{\sqrt{\mu_r \epsilon_r}}$$

Transmission line condition:

$$2l \leq t_r \cdot v_{ph}$$

Critical length for wire:

$$l_{crit} = \frac{t_r \cdot c}{2\sqrt{\mu_r \epsilon_r}}$$

$$t_r = 100\text{ns} \Rightarrow l_{crit} = 7.5\text{m}$$

$$t_r = 2.5\text{ns} \Rightarrow l_{crit} = 18.8\text{cm}$$

A low voltage differential signal (LVDS)-Driver is therefore not necessary for this application.

4.4. Mechanical Concept

The mechanical concept is modular too. The collimator (5) is inside a scanner head holder part (1) which is fixed to the CompOCT. It holds the scan header with two guiding pins (2) and a screw (3). So the coupling between CompOCT and scan head is free space. This allows the scan module to be independent of the rest. This enables a Plug & Play design approach for scan heads. And different scan heads may easily be exchanged. The proof of concept was made with a telecentric design (fig. 4.17) including the MEMS (6), its electronics (4) and a 1 inch lens (7).

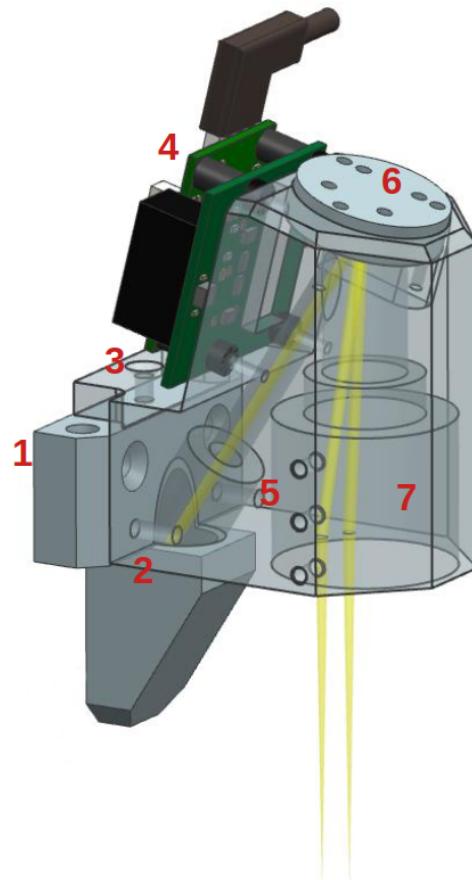


Fig. 4.17.: Planned PCB Layout.

5. MCU Software

5.1. Firmware Architecture

The scanner firmware is composed out of two parts. A real-time part that handles signal generation and triggering. And a background task scheduler that, for the moment, handles communication and command interpretation. Figure 5.2 and 5.1 show the two domains and their responsibilities.

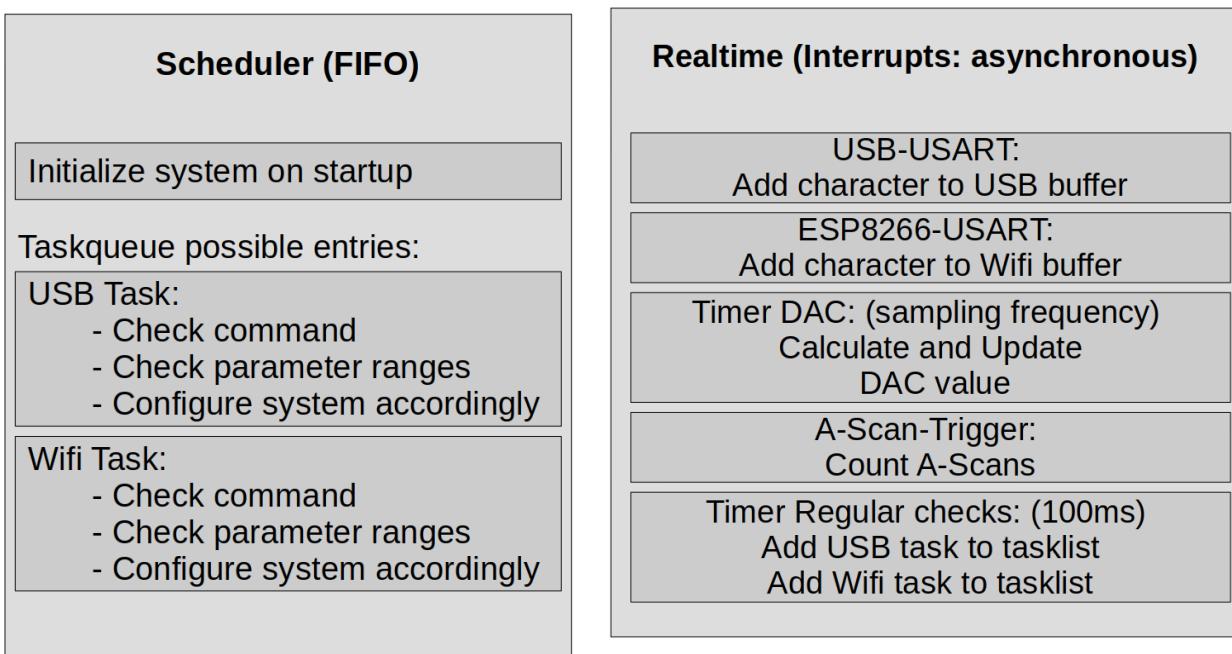


Fig. 5.1.: Task scheduler running on background.

Fig. 5.2.: Real-time part of firmware handled by interrupts.

5.1.1. Signal Generation

The signal generation is implemented as a state machine. This ensure proper ramping to signal start and MEMS idle position. Figure 5.3 shows this state machine.

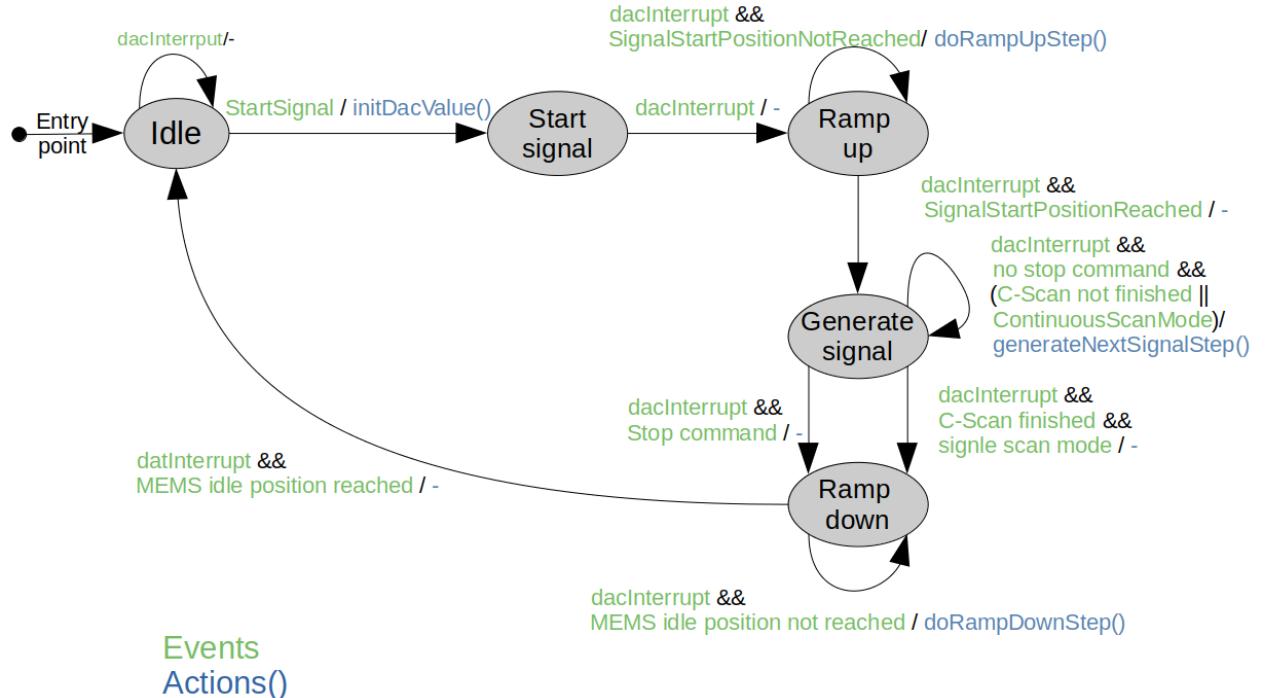


Fig. 5.3.: State Machine for driving signal generation.

5.1.2. Communication

There are two channels to communicate with the scanner module. One is through the serial port, the other over WiFi. In figure 5.4 the serial port channel is displayed and in figure 5.5 the WiFi channel.

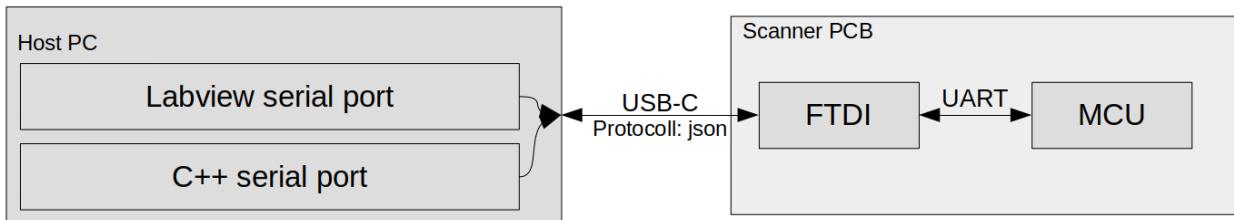


Fig. 5.4.: Black box diagram of serial communication channel.

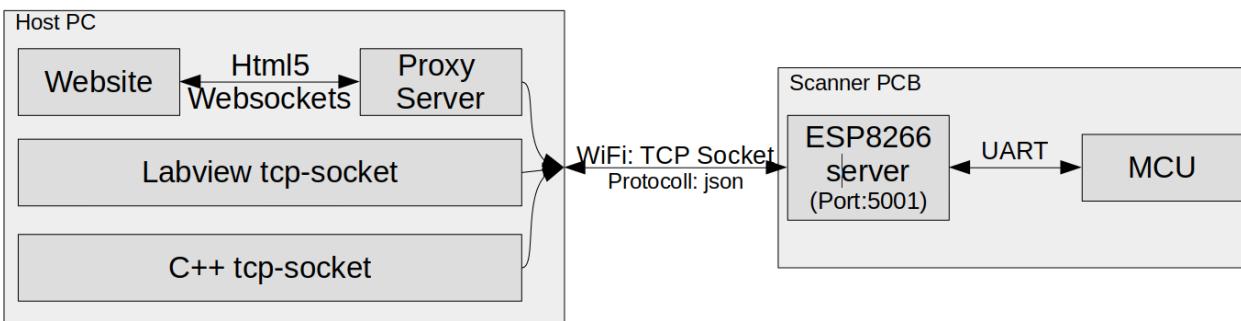


Fig. 5.5.: Black box diagram of tcp-socket communication channel.

After receiving a message the firmware checks several things if something does not check out the command will be ignored and an error returned.

- Check whether the message is in javascript object notation (JSON) style. If not it will return a wrong JSON error.
- Check whether the message is a valid command. If not it will return an unknown Command error.
- Check whether the parameter set of the command is complete. If not it will return a missing parameter error or an too many attributes error.
- Check whether all the parameters are inside the bounds of allowed values. If not, it will return a wrong parameter range error.
- Execute the command.

The complete communication protocol can be found in the appendix E.

6. MEMS performance

After creating the software and electronics, measurements were taken to demonstrate the functionality and performance of the scanner. To assess the performance it is compared to the standard Galvanometer (galvo)-scanners usually used for OCT applications.

6.1. Mesurement setup

To measure the mirror movement an B-Scan was acquired of a target with slopped surface. The setup was as follows:

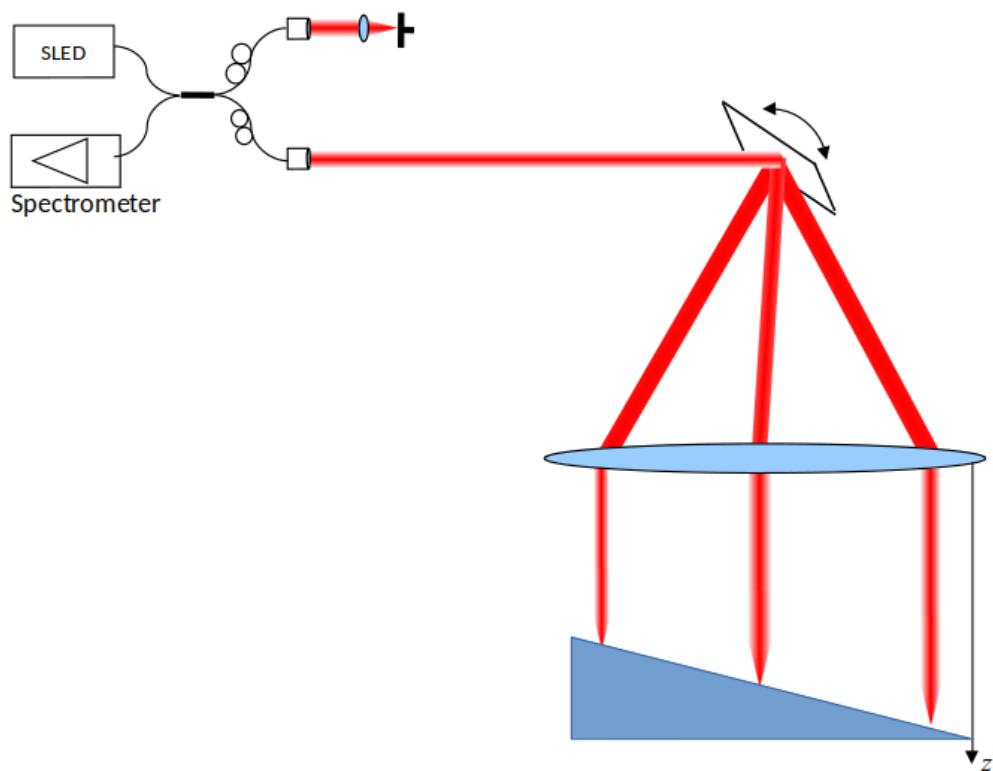
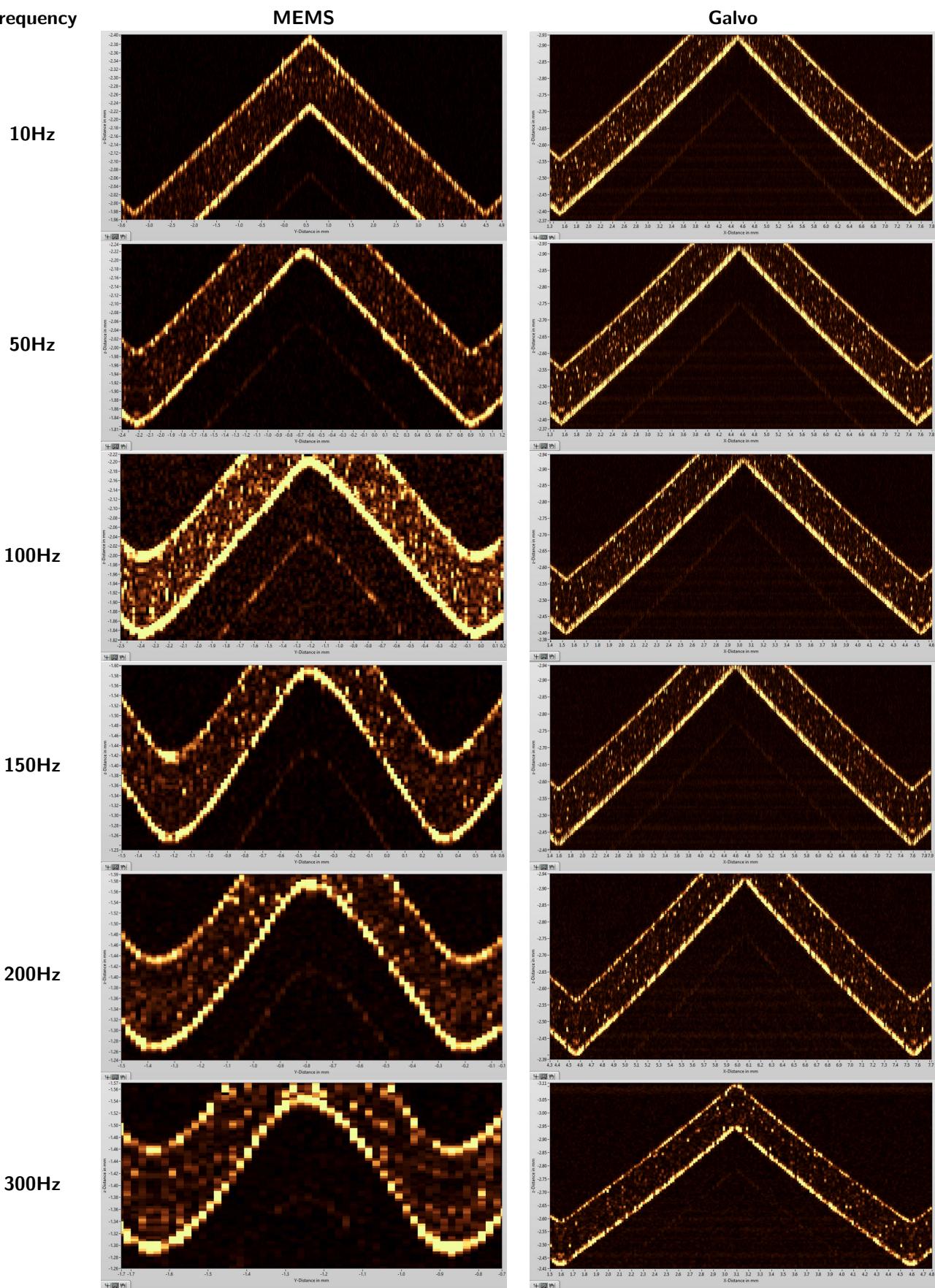


Fig. 6.1.: 1Hz saw tooth signal at MEMS mirror.

The MEMS measurements were take with the CompOCT system. But the Galvo measurements were made with another OCT system which was already build to use a galvo scan head. This is why the galvo measurements have a higher resolution. To analyze the mirror movement this doesn't matter.

6.2. MEMS mirror movement

Frequency



The measurements show that the mirror movement is very close to linear to a frequency of about 100Hz. In this frequency band the MEMS delivers similar results as the galvo scanner. Above 100Hz, the higher harmonic frequencies are damped too much to compensate for it. Therefore the triangular waveform turns step by step into a sine wave.

6.3. Limits of open-loop control

If the frequencies are selected specifically so that the first or second harmonic wave of the triangular waveform hits the resonance point of the MEMS, then the controller can't compensate it completely. Figures 6.2 and 6.3 show this.

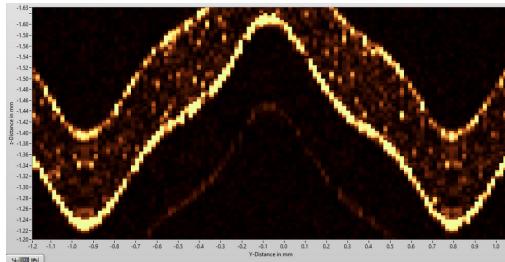


Fig. 6.2.: First harmonic wave hits resonance point.

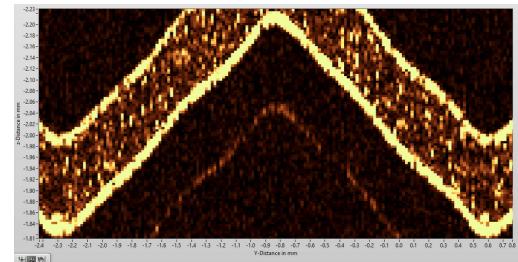


Fig. 6.3.: Second harmonic wave hits resonance point.

In the daily usage one should avoid the frequencies: $\left\{ \frac{f_g}{2N+1} \mid N \in \mathbb{N} \right\}$.

7. Conclusion: Magnetic MEMS scan head

This project combined control theory, optics, electronics and micro-controller software to generate a final prototype. The integration of the different parts is still in process. It could be shown, that it is possible to use open-loop controlling to drive a MEMS scan mirror with a frequency to a forth of the resonance frequency of the mirror. Above that the movement gets distorted. Further the magnetic MEMS mirror is indeed more robust against dust as there were never any issues on that account.

Additionally it was proven that a STM32F4 MCU is sufficient to calculate an IIR filtered signal to drive such a MEMS scan head in real-time ($<100\mu s$) and handle the JSON protocol based communication through universal asynchronous receiver transmitter (UART) connections in the background.

The developed analog electronics to drive the MEMS mirror which is a current driven device distorts the signal because positive and negative waves of the signal are handled by different components. So production variances of them translate to distorted, generated signals. Further, after integration, it started flipping between V_{dd} and V_{ss} when the resistance got higher than 100Ω . This phenomena has to be invested further because it didn't happen either with the prototype nor when used as standalone device.

8. Outlook

8.1. Possible future activities

Controlling Part:

- Automate calibration of mirror, for example with a frequency sweep signal.
- Try implementing a feedback control strategy like described in appendix ??

Electronics Part:

- Stabilize analog circuit.
- Change strategy to digital circuit.
- Control the CompOCT from a next unit of computing (NUC) like Nvidia Jetson.

Software Part:

- Implement NI-DAQ functionality.

Hardware Part:

- Adding another scanner design.

8.2. Feedback Control

8.2.1. Bode integral

The bode integral defines the maximal bandwidth a feedback controlled system can have. For plants with no unstable poles it's defined as:

$$\int_0^{W_{max}} \ln|S(j\omega)| d\omega = 0$$

What it says is that the area of the sensitivity function inside the usable bandwidth of a controlled plant must be 0. [?]

The sensitivity function $S(j\omega)$ and Complementary sensitivity function $T(j\omega)$ are defined as follows:

$$S(j\omega) = \frac{1}{1 + L(j\omega)}$$

$$T(j\omega) = \frac{L(j\omega)}{1 + L(j\omega)}$$

$L(j\omega)$: Loop transfer function.

Closing the loop around the MEMS plant provides an sensitivity function from which the potential benefit of a closed loop implementation can be estimated.

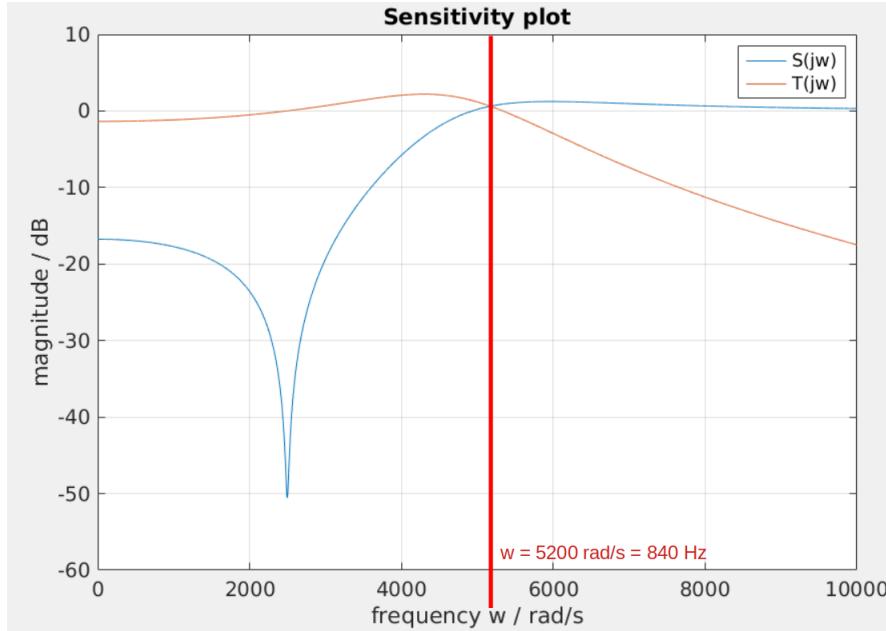


Fig. 8.1.: 1Hz Triangle signal at MEMS mirror.

Figure 8.1 shows potential to even further speed up the MEMS mirror with a feedback control strategy.

8.2.2. Controllability and Observability

The main problem for feedback controlling is that the only signal the MEMS mirror allows to measure is the current. For a motor this means it's possible to generate a constant torque or constant speed, but it's not possible to move to a position. That's because the initial position is unknown. But for the MEMS mirror, the initial position is known. To find out if the MEMS is controllable and observable it's possible to generate a controllability matrix and an observability matrix and check whether they are of full rank.

Given is the MEMS SS system with A, B, C and D. C defines the measured signal which must be the driving current.

$$P_c = [B \quad BC \quad BC^2]$$

$$P_o = \begin{bmatrix} C \\ CA \\ C^2A \end{bmatrix}$$

Both matrices are of full rank when inserted with the actual values. This means the MEMS system is controllable and observable.

8.2.3. Feedback control with observer

For an H_∞ or H_2 design the full state vector is needed as feedback. Because only the current can be measured an observer must estimate the other states. Such a design could look like in figure 8.2.[?]

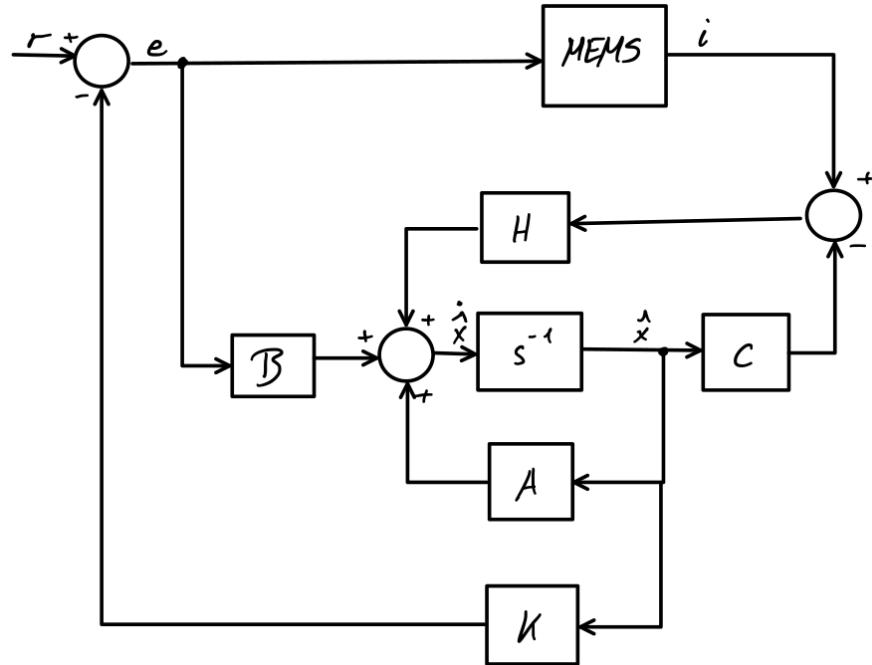


Fig. 8.2.: 1Hz Triangle signal at MEMS mirror.

Here A , B , C are the model, which is fed by the same input as the MEMS. It generates a estimated state vector (\hat{x}). The difference of the measured current (i) and the estimated current is fed back through matrix H to correct the error of the state estimate. The full estimated state vector is fed back through the pole placement vector K to build the error signal (e) together with the reference signal (r). [?, p. 22, State Space Control]

A. Sercalo MEMS data

A.1. Data sheet

X-Axis

Mechanical Resonance Frequency	404.64	Hz
Maximum Mechanical Angle	5.0	deg
Coil Resistance	11.46	Ω
DC Response	42.752	mA/ $^{\circ}$
Damping Ratio	3.223E-03	-

Y-Axis

Mechanical Resonance Frequency	243.32	Hz
Maximum Mechanical Angle	5.0	$^{\circ}$
Coil Resistance	8.16	Ω
DC Response	44.109	mA/ $^{\circ}$
Damping Ratio	3.551E-03	-

A.2. Measurement Data

Measurements of X-Axis of MEMS mirror

U[V] F[Hz] Height[mm]

0.1 10 0.115
 0.1 20 0.115
 0.1 40 0.116
 0.1 80 0.119
 0.1 100 0.125
 0.1 140 0.128
 0.1 180 0.135
 0.1 220 0.15
 0.1 260 0.18
 0.1 300 0.225
 0.1 340 0.35
 0.1 360 0.58
 0.1 380 0.93
 0.1 420 0.625
 0.1 440 0.38
 0.1 460 0.26
 0.1 500 0.19

Measurements of X-Axis of MEMS mirror

U[V] F[Hz] Height[mm]

0.1 10 0.115
 0.1 20 0.115
 0.1 40 0.116
 0.1 80 0.119
 0.1 100 0.125
 0.1 140 0.128
 0.1 180 0.135
 0.1 220 0.15
 0.1 260 0.18
 0.1 300 0.225
 0.1 340 0.35
 0.1 360 0.58
 0.1 380 0.93
 0.1 420 0.625
 0.1 440 0.38
 0.1 460 0.26
 0.1 500 0.19

A.3. Matlab Code

```

1 %% Measurement of X-Axis MEMS-Mirror
2 clf
3 clear
4 clc
5
6 [V, f, h] = textread("../Measurements/xAxisMeasData.txt", "%f %f %f", "headerlines", 2);
7
8 % Scanner data
9 angleOfTarget = 7
10 dMemsToTarget = 45
11
12 % Analysis of Measurement Data
13 Angle = atan2( (h / tand(angleOfTarget)), (2*dMemsToTarget)) % Calculating Angle from height
14
15 AngleDeg = Angle * 180/pi;
16 Vu = AngleDeg ./ V;
17 VuDB = 20*log10(Vu)';
18 w=f*2*pi;
19
20 %% fit state space model to measurement data
21 % data from datasheet
22 R = 11.46;
23 r = 0.0062/2;
24 dcGain = 0.042752;
25
26 % assumptions
27 bf = 0.0000001; % friction of mirror movement
28 wbreak_el = 3500; % break frequency of electronics
29 J = 4.565e-9; % Inertia of mems mirror
30
31
32 % Initial conditions for optimization
33 c = 100; % Spring coefficient of mems mirror
34 gain = 1000; % Frequency independent gain of mems mirror
35 param = [c, gain]; % Parameter set to optimize
36
37 % Do optimization to find c and gain.
38 options = optimset("PlotFcns", @optimplotfval, "TolFun", 1e-12);
39 [paramNew, error] = fminsearch(@calcSquareError, param, [], w, wbreak_el, VuDB, dcGain, bf, R, J, r);
40
41 % Overwrite the newly found parameters
42 c = paramNew(1)
43 gain = paramNew(2)
44
45 % Calculate the amplitude response as well as the state space plant
46 [VuDBFit2, sys] = createSSModel(w, wbreak_el, dcGain, bf, R, J, r, c, gain);
47 wSys = linspace(10*2*pi, 1000*2*pi, 100000);
48 [VuDBFit, sys] = createSSModel(wSys, wbreak_el, dcGain, bf, R, J, r, c, gain);
49 MEMS = sys;
50 save MEMS.mat MEMS
51
52 % Visualize Model
53 figure(1)
54 subplot(3,1,1)
55 semilogx(w, VuDB, "x", wSys, VuDBFit, "-");
56 title("Amplitude Response of MEMS mirror")
57 xlabel("Circular Frequency w / Hz")
58 ylabel("Amplification / dB")
59 legend("Measurement data", "Created model", "Location", "northwest")
60
61 subplot(3,1,2)
62 pzplot(sys)
63 axis([-5000, 5000, -3000, 3000])
64 legend("Created model", "Location", "northeast")
65
66 subplot(3,1,3)
67 step(sys)
68 legend("Created model", "Location", "northeast")
69 fig = gcf;
70 fig.PaperPositionMode = "auto";
71 fig.PaperUnits = "inches";
72 fig.PaperPosition = [0 0 18 9];
73 saveas(gcf, "MEMS_ModelPlots", "png");
74
75 finalSquareError = sum((VuDBFit2' - VuDB).^2) ./ length(w)
76 %% Calc Sensitivity
77
78 figure(99)
79 s=tf("s")
80 %bode( 1/(1+5*sys), 1*s^0, 5*sys / (1+5*sys), 5*sys)
81
82 w = linspace(1, 10000, 10000)
83 [Smag, Sangle] = bode(1/(1+sys), w)

```

```

84 [Tmag, Tangle] = bode(sys/(1+sys),w)
85 Smag = 20*log10(squeeze(Smag));
86 Tmag = 20*log10(squeeze(Tmag));
87 plot(w, Smag, w, Tmag)
88 title("Sensitivity plot")
89 legend("S(jw)", "T(jw)")
90 grid on
91 xlabel("frequency w / rad/s")
92 ylabel("magnitude / dB")
93
94 A = sys.A
95 B = sys.B
96 C = [0 0 1];%sys.C
97
98 Po = [C; C*A; C*A*A]
99 Pc = [B A*B A*A*B]
100
101 if(rank(Po) == 3)
102     SystemIsObservable = "yes"
103 else
104     SystemIsObservable = "no"
105 end
106 if(rank(Pc) == 3)
107     SystemIsControllable = "yes"
108 else
109     SystemIsControllable = "no"
110 end
111 if(rank(Po)==3 && rank(Pc)==3)
112     SystemIsMinimal = "yes"
113 else
114     SystemIsMinimal = "no"
115 end
116
117 %% Match poles
118 Tsys = tf(sys);
119 fs = 10000;
120 sysZ = c2d(Tsys, 1/fs, "impulse")
121 iir1 = 1/sysZ
122
123 nPer = 200;
124 fsig = 150;
125 N = fs/fsig;
126
127 t = linspace(0,nPer/fsig, N*nPer+1);
128 x = 1.5*sawtooth(2*pi*fsig*t+pi/2, 0.5);
129
130
131 s = tf("s");
132 TP = c2d(1/(1+s/6000)*1/(1+s/10000)*1/(1+s/10000), 1/fs, "impulse");
133 iir = iir1 * TP
134
135 figure(2)
136 subplot(1,2,1)
137 pzplot(iir,TP)
138 title("Discrete Time IIR Filter Pole-Zero plot")
139 subplot(1,2,2)
140 bode(iir)
141 title("Discrete Time IIR Filter Bode Diagram")
142 yDirect = zeros(1, length(x));
143 fig = gcf;
144 fig.PaperPositionMode = "auto";
145 fig.PaperUnits = "inches";
146 fig.PaperPosition = [0 0 18 9];
147 saveas(gcf,"IIR_TP_Discrete_Bode_impulse_fs10k","png");
148
149 %% Using direct form implementation of iir
150 clear directFormIIR % deletes persistant data inside directFormIIR
151 den = iir.den{1}
152 a = den / abs(den(1))
153 num = iir.num{1};
154 b = num / abs(den(1))
155 figure(3)
156 bode(iir)
157
158
159 %% creating std vector with coefficients
160 aStr = "std::vector<float> den = {/*a*/ ";
161 for i=1:length(a)
162     if(i<length(a))
163         aStr = aStr + a(i) + "f, ";
164     else
165         aStr = aStr + a(i)+ "f";
166     end
167 end
168 aStr = aStr + "};"

```

```

169 bStr = "std::vector<float> num = {/*b*/ " ;
170 for i=1:length(b)
171     if(i<length(b))
172         bStr = bStr + b(i) + "f, ";
173     else
174         bStr = bStr + b(i)+ "f";
175     end
176 end
177 bStr = bStr + "};"
178
179 %% Using coefficients to simulate driver
180 FilterGain = 3.5;
181 for i=1:length(x)
182     yDirect(i) = single(directFormIIR(b,a,single(x(i)))*FilterGain;
183     if(yDirect(i) > 1.5)
184         yDirect(i) = 1.5;
185     end
186     if(yDirect(i) < -1.5)
187         yDirect(i) = -1.5;
188     end
189 end
190
191 % Quantization to a 12 bit resolution
192 yDirect = yDirect * 4096;
193 yDirect = floor(yDirect);
194 yDirect = yDirect / 4096;
195
196 %% Eighter calc electronic gain and use datasheet dc gain of mirror
197 % electronicGainCompensation = 8.2/62; % (8.20hm / 620hm = gain)
198 % FilterGain = 1; % (620hm / 8.20hm) * (1V / 1.5V) = 5.04
199 % yDirect = yDirect*FilterGain;
200 yMD = lsim(sys, yDirect, t);
201 yMU = lsim(sys, x, t);
202
203 PerVis = 15;
204 startx = (nPer-PerVis)*N/fs;
205 stopx = startx + PerVis/fs*N;
206 figure(4)
207 subplot(3,1,1)
208 [hAx, hline1, hline2] = plotyy(t, x, t , yMD)
209 hline1.LineStyle = "-";
210 hline1.Marker = ".";
211 hline2.LineStyle = "-";
212 hline2.Marker = ".";
213 legend("Reference Signal / V","MEMS response on driving signal / deg")
214 xlabel("time / s")
215 title("Simulated mirror response on driving signal")
216 ylabel(hAx(1),"DAC Voltage / V") % left y-axis
217 ylabel(hAx(2),"Mirror Angle / deg") % right y-axis
218 xlim(hAx, [startx, stopx]);
219 ylim(hAx(1), [-5.5, 5.5]);
220 ylim(hAx(2), [-5.5, 5.5]);
221
222 subplot(3,1,2)
223 plot(t, yDirect+1.65)
224 title("MCU DAC Driving Signal")
225 xlabel("time / s")
226 ylabel("DAC Voltage / V")
227 legend( "Driving Signal simulation", "Driving Signal iir implementation")
228 axis([startx, stopx, 0, 3.3]);
229
230 subplot(3,1,3)
231 plot(t, x, "-.", t, yMU, "-.")
232 legend("Triangle input Signal", "MEMS response on triangle signal")
233 xlabel("time / s")
234 ylabel("Angle / deg")
235 title("Simulated mirror response on pure triangle signal")
236 axis([startx, stopx, -20, 20]);
237 fig = gcf;
238 fig.PaperPositionMode = "auto";
239 fig.PaperUnits = "inches";
240 fig.PaperPosition = [0 0 18 9];
241 saveas(gcf,"MEMS_Response","png");
242
243 %% Plotting Power Spectrum of Triangle Signal
244 TransferFunction = Tsys*d2c(iir,"tustin");
245 [mag, phase] = bode(TransferFunction, wSys);
246 AmpResp = 20*log10(squeeze(mag));
247
248 function err = calcSquareError(param, w, wbreak_el, VuDB, dcGain, bf, R, J, r)
249     VuDBModel = createSSModel(w, wbreak_el, dcGain, bf, R, J, r, param(1), param(2))';
250     err = sum((VuDB - VuDBModel).^2);
251 end
252
1 function [Vu, sys] = createSSModel(w, wbreak_el, dcGain, bf, R, J, r, c, gain)

```

```
2     L = R/wbreak_el;          % Calculate Induction to meet el. break frequency
3     K = c*r^2/(4*dcGain);    % Calculate generator coefficient to meet dc response of data sheet
4
5     % Create State Space matrices
6     A = [   -bf/J,           -r*c/(J*2),      K/J;
7             1,                  0,                 0;
8             -K/L,              0,      -R/L    ];
9
10    B = [    0,                0,      gain/L   ]';
11
12    C = [    0,                1,      0        ];
13    D = [    0                 ]';
14
15    sys = ss(A, B, C, D, "StateName", {"Velocity", "Position", "Current"}, "InputName", "Voltage");
16
17    [mag, phase] = bode(sys, w);
18    Vu = 20*log10(squeeze(mag));
19 end
```

B. Discretization Methods

A system controlled by a MCU is time discrete. Therefore it has to be converted from s-domain to z-domain. There are several conversion methods. These will be discussed briefly:[?]

zero order holder (ZOH):

Assumes the control inputs are piecewise constant over the sample time.

first order holder (FOH):

Assumes the control inputs are piecewise linear over the sample time.

Tustin method:

This method uses a bilinear transformation to map the s-domain to the z-domain. [?]

Impulse response matching:

This method is an adaptive approach to match the impulse response of the z-domain system to the s-domain system.

Pole-Zero matching:

This method is an adaptive approach to match the poles and zeros in z-domain to s-domain.

To visualize this figure B.1 shows how a step answer of the transfer function B.1 looks like after the conversion to discrete time.

$$H(s) = \frac{0.1s + 1}{s^3 + 4s^2 + 5s + 5} \quad (\text{B.1})$$

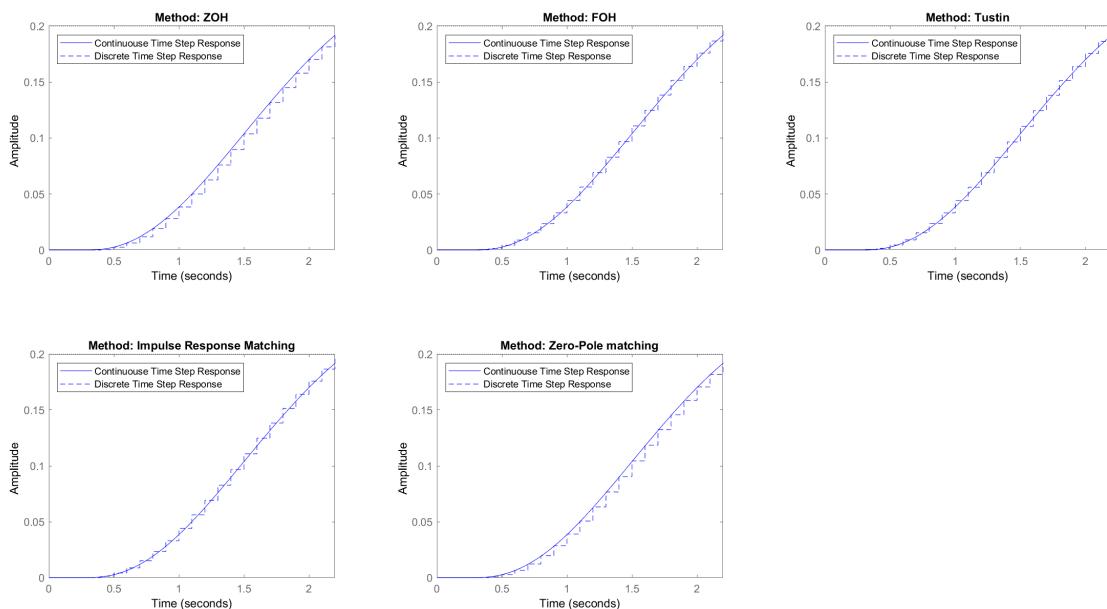


Fig. B.1.: Descrete time step response

As can be seen the differences of the step response are marginal. But in the pole-zero map shown in figure B.2 they're fundamentally different.

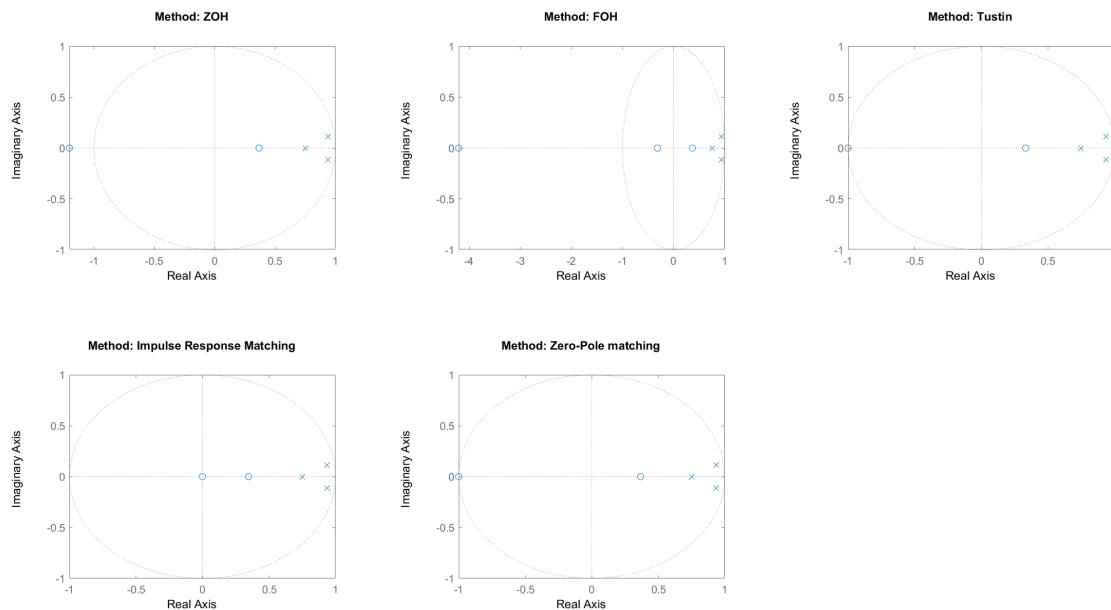


Fig. B.2.: Discrete pole-zero map

Considering that the intend is to switch poles and zeros to ideally control the mirror it is crucial that not only all poles but also all zero remain inside the unit circle for stability reasons. As the 'ZOH' as well as the 'FOH' method can produce poles outside of the unit circle, those are no options. Furthermore the 'Tustin' as well as the 'Zero-Pole-Matching' method create poles on the unit circle. These are going to cause problems when switching to a floating point resolution on the micro-controller unit. Remaining is the 'Impulse Response Matching' which is the method of choice for this work.

C. CompOCT

The CompOCT project at HuCE-optoLab tries to find the technological limits of miniaturizing an OCT system. The optics is embedded in a 3d printed (fig. C.4)metal part which uses all three dimensions for a miniaturized build. Figure C.2 shows this optic graphically and fig. C.3 schematically. The whole device looks as follows:

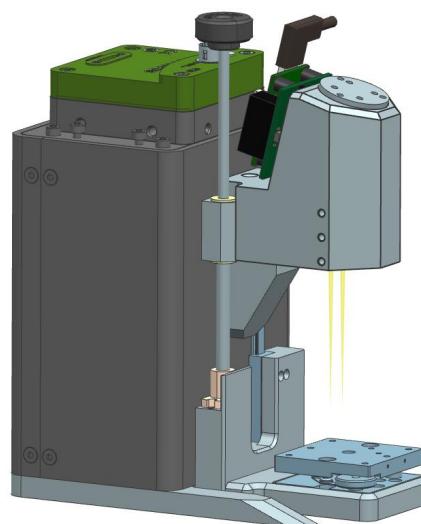


Fig. C.1.: CompOCT

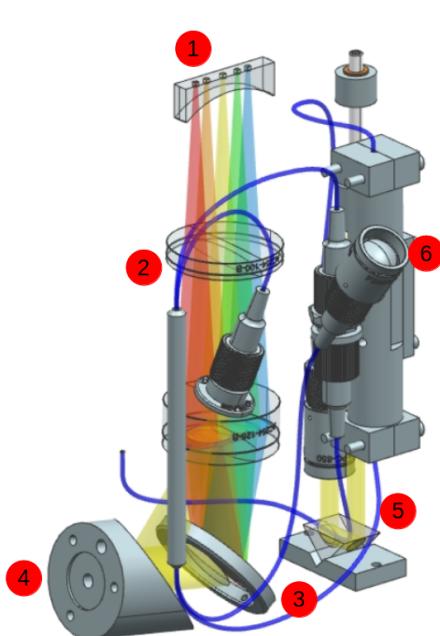


Fig. C.2.: Optics of CompOCT

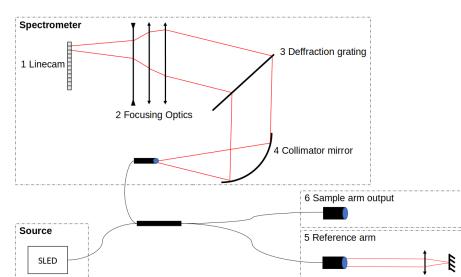


Fig. C.3.: Schematic of optics



Fig. C.4.: Printed metal skeleton.

D. Circuits

D.1. Heat dissipation of IRF7309

As only one of the MOSFETs is active at a given time, the approximated power dissipation is calculated as follows:

$$\begin{aligned} R_{Coil} &= 17\Omega \\ V_{dd} &= 12V \\ I_{Dmax} &= 0.2A \\ P_D &= (V_{dd} - R_{Coil} \cdot I_{Dmax}) \cdot I_{Dmax} \approx 1.7W \end{aligned}$$

The thermal resistance from junction to lead of a SO8 package is: $28.7^{\circ}\text{C}/\text{W}$. [?, Infineon, p. 2]

$$\begin{aligned} T_{max} &= 130^{\circ}\text{C} \\ T_{amb} &= 25^{\circ}\text{C} \\ \Rightarrow \Theta_{JA} &= \frac{T_{max} - T_{amb}}{P_D} \approx 60^{\circ}\text{C}/\text{W} \\ \Theta_{JL} &= 28.7^{\circ}\text{C}/\text{W} \\ A_{PCB} &= \frac{500 \frac{{}^{\circ}\text{C}cm^2}{W}}{\Theta_{JA} - \Theta_{JL}} = 15cm^2 \end{aligned}$$

As the calculations show, a heat sink is needed to dissipate the heat properly. Calculations with heat sink:
Proposal: APF19-19-06CB/A01

$$\begin{aligned} T_{amb} &= 25^{\circ}\text{C} \\ \Theta_{LA} &\approx \Theta_{Cooler} = 17^{\circ}\text{C}/\text{W} \\ T_{max} &= T_{amb} + (\Theta_{JL} + \Theta_{LA}) \cdot P_D \approx 100^{\circ}\text{C} \end{aligned}$$

This would generally work, but requires a lot of space. Instead the decision was to attach the PCB over large copper areas and thermal foils directly to the aluminum case which is massive and should dissipate enough energy to cool the amplifier print. [?, Thermal Design, p. 5-6]

$$\Theta_{J-A} = \Theta_{J-L} + \Theta_{L-P} + \Theta_{P-Al} + \Theta_{Al-A}$$

$$\Theta_{J-L} = 28.7^\circ C/W$$

$$\Theta_{L-P} = \Theta_{Vias} = \frac{261}{N} \Rightarrow N = 50 \Rightarrow \Theta_{L-P} = 5.22^\circ K/W$$

$$N_{max} = \frac{A_{Pad}}{A_{Via}}$$

$$\Theta_{P-L} = \Theta_{Foil} \approx 0.4^\circ K/W$$

$$\Theta_{Al-A} = \frac{1}{\alpha_{Al} \cdot A_{Al}} \approx 13.3^\circ K/W$$

$$\alpha_{Al} = 3.5...35$$

$$A_{Al} = 0.015 m^2$$

$$P_D \approx 1.5W$$

$$T_{Al} = 2 \cdot P_D \cdot \Theta_{Al-A} + T_{amp} \approx 65^\circ C$$

$$T_J = P_D (\Theta_{J-L} + \Theta_{L-P} + \Theta_{P-Al} + 2 \cdot \Theta_{Al-A}) + T_{amp} \approx 115^\circ C$$

$$T_{amb} = 25^\circ C$$

Thermal foil: CD-02-05-247-N: $\Theta_{Foil} = 0.7^\circ K \cdot cm^2/W$ (data sheet)

Θ_{J-A} :	Thermal junction to ambient resistance
Θ_{J-L} :	Thermal junction to lead resistance
Θ_{L-P} :	Thermal lead to pad resistance
Θ_{P-Al} :	pad junction to aluminum block resistance
Θ_{Al-A} :	Thermal aluminum to ambient resistance
Θ_{Vias} :	Thermal resistance of a 12mil via
α_{Al} :	Free convection coefficient aluminum to air
A_{Al} :	Surface area of aluminum block
T_J :	Junction temperature
T_{Al} :	Temperature of aluminum block
T_{amb} :	Ambient temperature
P_D :	Dissipated power from device

D.2. USB-C

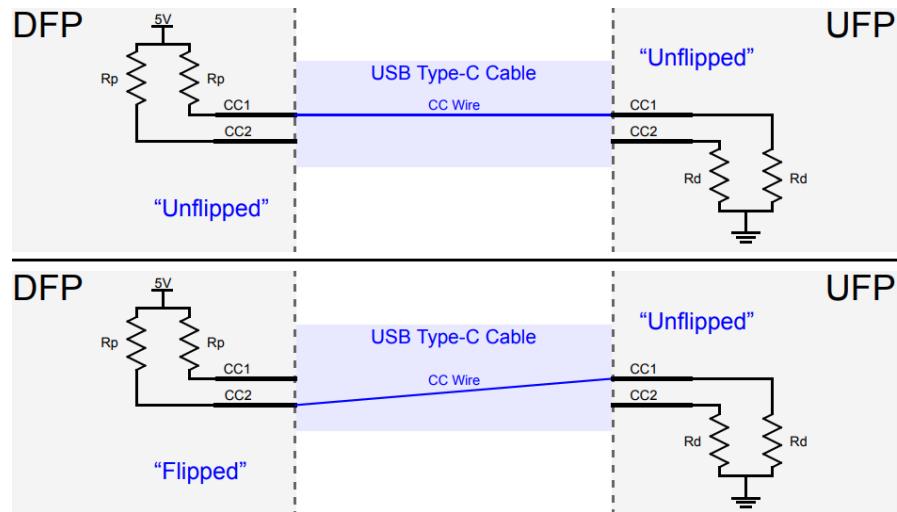


Fig. D.1.: USB-C CC-Pin connection scheme.

CC-Pin Config:	R_p	R_d	Voltage Range
USB-2.0 500mA	$56\text{k}\Omega \pm 20\%$	$5.1\text{k}\Omega \pm 5\%$	0.3-0.6V
USB-3.0 900mA	$56\text{k}\Omega \pm 5$	$5.1\text{k}\Omega \pm 5\%$	0.34-0.5V
USB-3.0 1.5A	$22\text{k}\Omega \pm 5$	$5.1\text{k}\Omega \pm 5\%$	0.78-1.12V
USB-3.0 3.0A	$10\text{k}\Omega \pm 5$	$5.1\text{k}\Omega \pm 5\%$	1.42-1.98V

The table shows that the two CC pin must be monitored with an ADC-Converter and evaluated by the microprocessor. It is the up stream facing port (UFP)s job to ensure its power consumption is lesser than the down stream facing port (DFP) can provide.

D.3. Load-Switches

To Enable Ampli-PCB as well as the superluminescent light emitting diode (SLED) Source.
Proposal: TPS22919DCKR

E. Communication protocol

The communication is based on a simplified JSON protocol from the open-source, light weight jsmn library. [?] As the MCU has a limited amount of RAM, the maximal number of characters of such a JSON-command is 1023. The Device always returns the actual state.

E.1. Typical communication flow

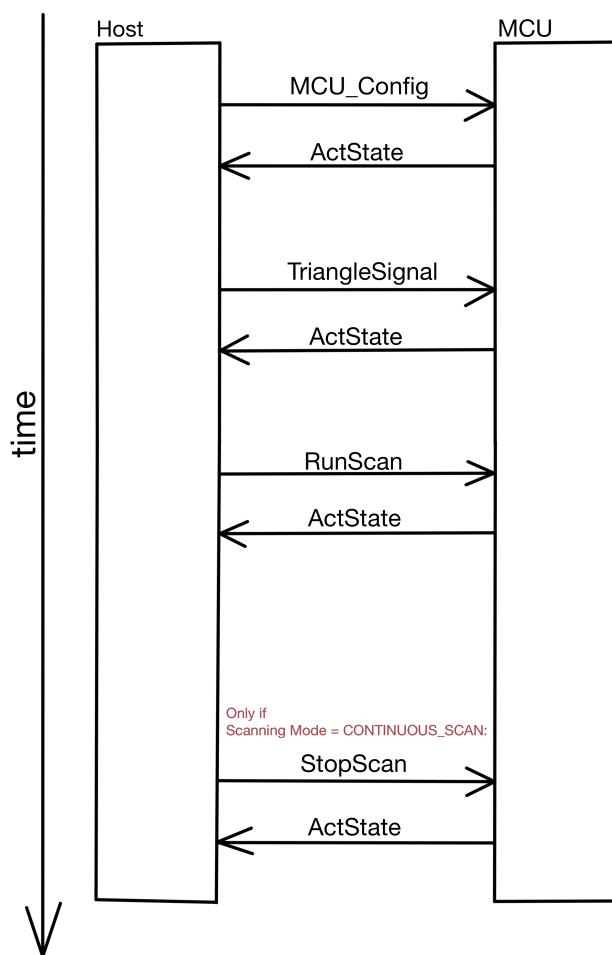


Fig. E.1.: Typical communication procedure.

Do not send multiple commands without waiting for the respective ActState answer as it will generate an error. A configuration commands mustn't be made while scanning.

E.2. Commands

General command build up:

```

1 {
2   "Command": {
3     "IntAttribute": IntValue,
4     "FloatAttribute": FloatValue,
5     "StringAttribute": "StringValue"
6   }
7 }
```

Configuration commands:

```

1 {
2   "MCUConfig": {
3     "OCTType": "SpectrometerOCT", /* Options: "SpectrometerOCT", "SSOCT" */
4     "AScanFrequency": 20000,      /* 1-1'000'000 [Hz] */
5     "ScanningMode": "SingleScan" /* Options: "SingleScan", "ContinuousScan" */
6   }
7 }
8
9 {
10  "SetFilter": {
11    "FilterName": "IIR_X" /* Options: "IIR_X", "IIR_Y" */
12    "Fs": 10000,          /* Hz */
13    "Gain": 1.4,          /* float */
14    "a0": 1.1,            /* float */
15    "a1": 1.1,            /* float */
16    "a2": 1.1,            /* float */
17    "a3": 1.1,            /* float */
18    "a4": 1.1,            /* float */
19    "b0": 1.1,            /* float */
20    "b1": 1.1,            /* float */
21    "b2": 1.1,            /* float */
22    "b3": 1.1,            /* float */
23    "b4": 1.1            /* float */
24  }
25 }
```

Response of Microcontroller:

```

1 {
2   "ActState": {
3     "OCTType": "SpectrometerOCT",           /* Options: "SpectrometerOCT", "SSOCT" */
4     "DACSamplingFrequency": 100000,          /* 10k [Hz] */
5     "AScanFrequency": 20000,                 /* 1-120'000 [Hz] */
6     "ScanningMode": "SingleScan",           /* Options: "SingleScan", "ContinuousScan",
7                               "NOT_CONFIGURED" */
8     "RunningState": "Idle",                  /* SingleScan, "Run", "Idle" */
9     "XAxisSignalType": "SineSignal",         /* Options: "SineSignal", "TriangleSignal", "ConstantSignal" */
10    "YAxisSignalType": "ConstantSignal",     /* Options: "SineSignal", "TriangleSignal", "ConstantSignal" */
11
12    "AScansPerBScan": 500,                   /* Range: 0-65535 */
13    "BScansPerCScan": 500,                   /* Range: 0-65535 */
14    "ErrorState": "NoError"                  /* Errors: NoError, WrongJSON, UnknownCMD,
15                                         MissingParameter, ScanModeNotConfigured,
16                                         WrongParameterRange, ToManyAttributes,
17                                         EmptyJSON */
18 }
```

Signal Configuration commands

```

1   "RunScan"
2 }
3
4 {
5   "StopScan"
6 }
7
8 {
9   "GetState"
10}
11
12{
13  "SineSignal": {
14    "OutputStage" : "XAxis",      /* Options: "XAxis", "YAxis" */
15    "Frequency" : 120,          /* 0.001 - 1500 [Hz] */
16    "Amplitude" : 1.5,          /* 0 - 1.5 [V] */
17    "Phase": 0,                /* 0-360 [deg] */
18    "Offset": 0,                /* -1.5 - 1.5 [V] */
19    "Filter": "NoFilter"        /* Options: "NoFilter", "IIR_X", "IIR_Y", "MagnMemsIIR10000" */
20  }
21 }
22
23
24{
25  "TriangleSignal": {
26    "OutputStage" : "XAxis",      /* Options: XAxis, YAxis */
27    "Frequency" : 200,           /* 0.01 - 1000 [Hz] */
28    "Amplitude" : 1.5,           /* 0 - 1.5 [V] */
29    "Phase": 90,                /* 0-360 [deg] */
30    "Offset": 0,                /* -1.5 - 1.5 [V] */
31    "Width": 0.5,              /* 0 - 1[1] Defines stepness of slopes*/
32    "Filter": "IIR_X" /* Options: "NoFilter", "IIR_X", "IIR_Y", "MagnMemsIIR10000" */
33  }
34 }
35
36{
37  "ConstantSignal": {
38    "OutputStage" : "XAxis",      /* Options: XAxis, YAxis */
39    "Duration" : 0.05,           /* 000001 - 15000 [s] */
40    "DCValue" : 0,               /* -1.5 - 1.5 [V] */
41    "Offset": 0,                /* -1.5 - 1.5 [V] */
42    "Filter": "NoFilter"        /* Options: "NoFilter", "IIR_X", "IIR_Y", "MagnMemsIIR10000" */
43  }
44 }
```