

## Master Thesis

# Development of a Control System to Improve the Stability of the FLUTE Electron Gun

Marvin-Dennis Noll

Supervisors: Prof. Dr.-Ing. John Jelonnek (IHM)  
Prof. Dr. Anke-Susanne Müller (IBPT)  
Advisor: Dr. Nigel Smale (IBPT)

Period: 15.11.2020 – 07.07.2021

Karlsruhe, 07.07.2021



Institute for Beam Physics and Technology  
Hermann-von-Helmholtz-Platz 1  
Campus North, Building 348  
76344 Eggenstein-Leopoldshafen



Institute for Pulsed Power and Microwave Technology  
Hermann-von-Helmholtz-Platz 1  
Campus North, Building 421  
76344 Eggenstein-Leopoldshafen



# **Declaration**

I hereby declare that I wrote my master thesis on my own and that I have followed the regulations relating to good scientific practice of the Karlsruhe Institute of Technology (KIT) in its latest form. I did not use any unacknowledged sources or means, and I marked all references I used literally or by content.

Karlsruhe, 16.07.2021

Marvin-Dennis Noll



# Abstract

The compact accelerator Ferninfrarot Linac- und Test-Experiment (FLUTE) is currently under commission at Karlsruhe Institute of Technology (KIT). Its main purposes are to serve as a technology platform for accelerator research and the generation of strong and ultra short terahertz (THz) pulses.

The Radio Frequency (RF) photo injector, also called the electron gun and the Linear Accelerator (LINAC) are powered by a klystron. It is fed by a pulse forming network, which is driven by a high voltage source connected to mains power. For stable energies of the generated THz pulses, the electron energies have to be stable. To ensure stable energies of the emitted electron bunches, several parameters of the gun, such as temperature and the RF power supply from the klystron, have to stay inside tight tolerance bands.

In this work, instead of passively optimizing the stability of system components, such as the water coolers or power supplies, an active approach with a closed feed-back loop is evaluated. By means of a control system, the amplitude of the low power RF input signal of the klystron is manipulated to mitigate the effects of noise and drifts on the electron energy. As there is currently no non-destructive sensor to measure the electron energies of all the electron bunches, the RF power in the first gun cavity is used instead as an estimator for the electron energy stability.

As part of the development process, first the stability issue is analyzed and metrics for quantifying the stability are defined. Then, an appropriate solution, a linear, discrete time control system, is proposed. In order to implement it, all the necessary building blocks of such a control system are treated in detail. First the necessary sensors and actuators are selected. Then the controller and the measurement filter are designed. To verify the designed system, first an offline simulation on a computer is performed which shows qualitatively a satisfactory disturbance rejection with a measured disturbance signal from FLUTE.

Then the control system is implemented as an algorithm with a fixed-interval control loop using the Python programming language. A graphical user interface, written in Qml, provides the user with plots and status information and allows the fine-tuning of the controller.

The following experiments at FLUTE show results in accordance to the simulation. That is, the stability, when defined as the relative standard deviation, is improved greatly by about a factor of 25.

Finally, ways to refine the control system are regarded. First, by using disturbance feed-forward of the change in waster temperature, the control system is made more robust and achieves the same results. Second, the usage of a Faraday cup, which measures total electron charge provides a potentially better representation of the electron energies, however as the electron beam is lost in the cup, its usages are limited.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Theoretical Framework</b>	<b>5</b>
2.1	Relativistic Electrons . . . . .	5
2.2	Signal Analysis . . . . .	6
2.2.1	Auto- and Cross-Covariance . . . . .	6
2.2.2	Estimating the Spectrum of a Stochastic Process . . . . .	7
2.3	Feedback Control Systems . . . . .	9
2.3.1	Disturbance Rejection and Input Tracking . . . . .	10
2.3.2	Stability . . . . .	10
2.4	Metrics to Quantify the Stability of a Signal . . . . .	11
<b>3</b>	<b>The Stability of FLUTE the Electron Gun and Proposed Stabilizing Solution</b>	<b>13</b>
3.1	The Electron Gun . . . . .	14
3.2	Relation between RF power and Electron Energy . . . . .	15
3.3	Current RF Stability and Proposed Solution . . . . .	16
<b>4</b>	<b>Interfacing FLUTE</b>	<b>19</b>
4.1	Inputs . . . . .	19
4.1.1	Accessing EPICS channels in Python with PyEpics . . . . .	19
4.1.2	Properties of the Available Process Variables . . . . .	20
4.1.3	Filtering the RF power Signals . . . . .	22
4.2	Output: Controllable RF Attenuator . . . . .	23
4.2.1	Defining Requirements . . . . .	23
4.2.2	Evaluation of the ZX73-2500-S+ Controllable RF Attenuator . . . . .	23
4.2.2.1	Common Measurement Setup . . . . .	26
4.2.2.2	Relation between Control Voltage and Attenuation . . . . .	27
4.2.2.3	Influence of Supply Voltage Noise on Attenuation . . . . .	30
4.2.2.4	Influence of RF Frequency on Attenuation . . . . .	32
4.2.2.5	Influence of Case Temperature Variations on Attenuation . . . . .	32
4.2.2.6	$V_{control}$ Frequency response . . . . .	35
4.2.2.7	Combined Maximum Error of the Attenuation and Conclusion . . . . .	36
4.2.3	Test of the Attenuator with FLUTE . . . . .	36
<b>5</b>	<b>Controller Design and Evaluation</b>	<b>39</b>
5.1	Plant Identification . . . . .	39
5.1.1	Principle . . . . .	39
5.1.2	Identifying the Plant Attenuator+RF . . . . .	39
5.2	Measurement Filter . . . . .	43
5.3	Controller Design . . . . .	48
5.3.1	Choosing a Controller Type . . . . .	48

5.3.2	Designing a Discrete Time PID Controller . . . . .	48
5.3.2.1	Controller Tuning . . . . .	50
5.3.3	Analyzing the Input Tracking and Disturbance Rejection . . . . .	52
5.3.4	Analyzing the Controller Stability . . . . .	55
5.3.5	Offline Evaluation with Measured Data . . . . .	57
5.4	Implementation of the Control System in Software . . . . .	58
5.5	Evaluation of the Control System (Online) . . . . .	60
5.6	Further Improvements to the Control System . . . . .	64
5.6.1	Changing the Controller Architecture: Disturbance Feed-Forward of the Gun Temperature . . . . .	64
5.6.2	Feedback on Faraday-Cup Charge Measurements . . . . .	65
<b>6</b>	<b>Summary and Outlook</b>	<b>69</b>
<b>Appendix</b>		<b>71</b>
A	Complementary Material Controller Design . . . . .	71
B	Lab Test and Measurement Equipment . . . . .	76
<b>Acknowledgments</b>		<b>79</b>
<b>Acronyms</b>		<b>80</b>

# List of Figures

1.1	FLUTE schematic with all components . . . . .	1
2.1	Control system structure . . . . .	9
3.1	FLUTE RF schematic . . . . .	13
3.2	Plot of the electric field $E_z(z)$ in the electron gun . . . . .	14
3.3	Cross-section of the electron gun . . . . .	15
3.4	Deviation of the cavity RF power over the course of one hour . . . . .	17
3.5	Periodogram of Figure 3.4; calculated using the Welch method . . . . .	17
3.6	Normalized cavity power and gun temperature . . . . .	18
3.7	Cross covariance analysis of cavity power and gun temperature . . . . .	18
3.8	FLUTE RF schematic with control unit . . . . .	18
4.1	Comparing three common FLUTE process variables available in EPICS . .	21
4.2	Histogram of EPICS CA sample times . . . . .	22
4.3	The controllable attenuator . . . . .	24
4.4	Design of a controllable attenuator . . . . .	25
4.5	Attenuation vs. frequency plot of the attenuator . . . . .	26
4.6	Measurement setup to evaluate the attenuator . . . . .	27
4.7	Attenuation vs control voltage . . . . .	29
4.8	Detailed version of Figure 4.7 . . . . .	29
4.9	Long term stability of the control voltage . . . . .	30
4.10	Influence of the supply voltage $V_+$ on the attenuation . . . . .	31
4.11	Long term stability of the supply voltage . . . . .	31
4.12	Influence of an offset frequency on attenuation . . . . .	32
4.13	Influence of the case temperature on attenuation . . . . .	33
4.14	Processed data from Figure 4.13 . . . . .	34
4.15	Measured temperature stability . . . . .	34
4.16	Spectrum showing maximum modulation speed . . . . .	35
4.17	Evaluation control voltage signal . . . . .	37
4.18	Evaluation measured attenuation . . . . .	37
4.19	Evaluation attenuation vs control voltage . . . . .	37
5.1	Identification sequence and response . . . . .	40
5.2	Validation of the process model . . . . .	41
5.3	Impulse Responses of different time discrete filters . . . . .	44
5.4	Filter effects of different time discrete filters . . . . .	45
5.5	FIR pole zero maps . . . . .	46
5.6	FIR magnitude response . . . . .	47
5.7	Block diagram of a PID controller . . . . .	48
5.8	System architecture for the Matlab PID tuner . . . . .	51
5.9	Input tracking of controller one and two . . . . .	53
5.10	Input tracking of controller three and four . . . . .	53

5.11	Disturbance rejection of controller one and two . . . . .	54
5.12	Disturbance rejection of controller three and four . . . . .	54
5.13	Nyquist plot for all controllers . . . . .	56
5.14	Unstable system . . . . .	56
5.15	Output offline evaluation . . . . .	58
5.16	Time signal whole experiment . . . . .	61
5.17	Spectrogram of Figure 5.16 . . . . .	61
5.18	Controller success time signal . . . . .	62
5.19	Controller success spectrum . . . . .	62
5.20	Relative standard deviation $\%STD(t, T)$ . . . . .	63
5.21	Relative standard deviation $STD\%(T)$ . . . . .	63
5.22	Feed-forward schematic . . . . .	65
5.23	Inverting integrator schematic . . . . .	67
5.24	Cavity power . . . . .	68
5.25	Electron charge . . . . .	68
A.1	Matlab System Identification Toolbox . . . . .	71
A.2	Matlab PID Tuner . . . . .	71
A.3	Simulink Model for Offline Evaluation . . . . .	72
A.4	Control System GUI . . . . .	73

# List of Tables

4.1	Comparing quantization steps . . . . .	21
4.2	Requirements for the controllable RF attenuator . . . . .	23
4.3	Evaluation results . . . . .	36
5.1	Estimated process models . . . . .	41
5.2	Calculated PID parameters . . . . .	52
5.3	Quantitative assessment of the controllers performance . . . . .	60
B.1	Agilent 34411A - Specifications . . . . .	76
B.2	Agilent 34411A - Relevant SCPI commands . . . . .	76
B.3	Keysight 34470A - Specifications . . . . .	76
B.4	Keysight 34470A - Relevant SCPI commands . . . . .	76
B.5	Keysight 34972A - Specifications . . . . .	76
B.6	Keysight 34972A - Relevant SCPI commands . . . . .	77
B.7	Rohde and Schwarz SMC100A - Specifications . . . . .	77
B.8	Rohde and Schwarz SMC100A - Relevant SCPI commands . . . . .	77
B.9	HP E4419B - Specifications . . . . .	77
B.10	HP E4419B - Relevant SCPI commands . . . . .	77
B.11	Agilent E5071C - Specifications . . . . .	78
B.12	Holzworth HA7062C - Specifications . . . . .	78



# 1. Introduction

In this thesis, methods to improve the stability of the electron gun of the Ferninfrarot Linac- und Test-Experiment (FLUTE) accelerator are studied.

FLUTE is a Linear Accelerator (LINAC)-based accelerator test-facility and a terahertz (THz) photon source that is currently under commission at the Karlsruhe Institute of Technology (KIT). It aims to be a source of high field THz pulses (up to  $1200 \text{ MV m}^{-1}$ ) in the femtosecond range, provide a test facility for accelerator research and an injection device for compact Storage ring for Accelerator Research and Technology (cSTART) (see [SchaeferHaererPapash2019\_1000091183]) in the future. [Naknaimueang:2011zz]

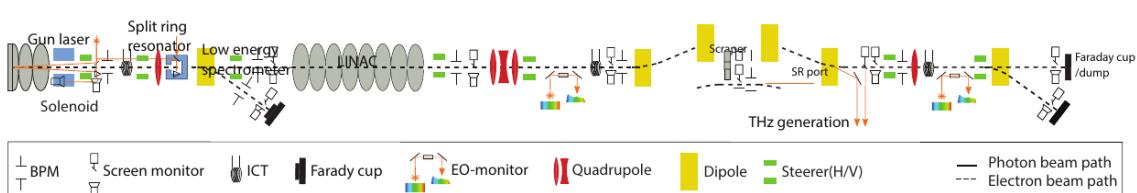
The accelerator is designed for a final electron momentum of  $41 \text{ MeV/c}$  and bunch charges of  $1 \text{ pC}$  to  $3000 \text{ pC}$  with lengths of  $1 \text{ fs}$  to  $300 \text{ fs}$ . The bunches are emitted with a repetition frequency of up to  $10 \text{ Hz}$ . [Malygin2018]

Figure 1.1 shows the finished accelerator schematically. It consists of the low energy section, the LINAC, the four-dipole bunch compressor and several diagnostic instruments.

Along with several diagnostic devices, the low energy section contains the electron gun that pre-accelerates electrons to  $7 \text{ MeV/c}$ . The electrons are generated photo-electrically at the cathode inside the gun through stimulation with ultraviolet (UV) radiation (270 nm) generated by a Ti:Sa laser. After that a solenoid is used to focus the electron beam for injection into the LINAC section. The LINAC, a 156-cell travelling wave structure, is then used to accelerate the electrons to  $41 \text{ MeV/c}$ . With a setup of four dipole magnets, the bunches are compressed longitudinally before the last dipole is used to generate Coherent Synchrotron Radiation (CSR) and Coherent Edge Radiation (CER) for THz experiments. [Nasse:IPAC13-WEPWA010]

At the time of writing, the low energy section is fully operational and at the end, a Faraday cup is installed as a beam dump to measure the electron bunch charges. It has to be removed when the LINAC section is commissioned.

Scientific experiments, such as THz spectroscopy or the study of THz radiation effects on biological tissue, rely on a known and stable wavelength of the THz radiation.



**Figure 1.1:** Schematic of the finished accelerator showing all installed and planned components (reprinted from [Yan2018])

The THz pulses used for experiments is mostly Coherent Transition Radiation (CTR). This type of radiation is emitted when the short electron bunches interact with the vacuum-metalfoil interface at the screen downstream of the compressor section (see Figure 1.1). [Yan:IPAC2016-WEPOY037; Thongbai2007]

According to [ding2019coherent], the energy spectrum of the CTR emitted by a single electron (per frequency  $\omega$  and solid angle  $\Omega$ ) is given

$$\frac{d\xi}{d\omega d\Omega} = \frac{q^2}{\pi^2 c} |S(\beta, \varphi, \phi)|^2 \quad (1.1)$$

with the electron charge  $q$ , the speed of light  $c$ , the angle  $\varphi$  between the electron beam and the incidence plane, the angle  $\phi$  between the incidence plane and the observation direction, the normalized electron velocity  $\beta$  and

$$S(\beta, \varphi, \phi) = \frac{\beta \cos \varphi (\sin \phi - \beta \sin \varphi)}{(1 - \beta \sin \phi \sin \varphi)^2 - (\beta \cos \phi \cos \varphi)^2}. \quad (1.2)$$

Therefore, the energy or the wavelength of the generated THz photons depend on the electron velocity. The spread in THz photon energy is thereby also a function of the electron velocity, so ultimately the electron energy.

Also, FLUTE being a test facility, adding and changing out components, possibly developed by other research institutes, is a common routine. To ensure compatibility among these components, reliable beam parameters at the interfaces between them are necessary. These parameters include the beams position in the horizontal and vertical direction, the beam steering angle, the electron energy, the emittance and the charge of the bunch and its dimensions. Since focusing and steering of the beam is done with electromagnets, it is also effected by the electron energy, as the deflection of an electron in a magnetic field is a function of its velocity, so ultimately its energy.

Besides depending on the wavelength of the UV pulses hitting the cathode, the electron energies also strongly dependent of the on the geometrical, electrical and thermal characteristics from the electron gun and its Radio Frequency (RF) power supply. These characteristics are not independent from each other and variations of them can have a multitude of intrinsic or extrinsic causes.

To improve the stability of the electron energy, in this thesis these causes are analyzed and measures against them or their effects are developed. Eventually this leads to the design of a *closed-loop* control system.

This approach is different from past efforts to improve the stability, which primarily treated the stability of (sub-) components that supply the electron gun. For instance, instabilities due to temperature variations were dealt with by stabilizing the water cooling system of the electron gun, or disturbances from the pulse forming network were reduced by improving the power supply. These measures can be thought of changes to an *open-loop*. This is because there is no feeding back of the success or error from the system output, the electron energy, to the water cooler or power supply. Hence for these improvements to work, deep knowledge of the relation between the components and their effect on the electron energy is required. Also, if parameters of the components change due to aging, environmental factors, such as temperature or noise on the mains power, these effects can be carried over to the electron energy.

These are the main two reasons which motivate a closed-loop approach. It is often not required to know the exact relationships between certain input disturbances and the electron energy. Most of the times qualitative descriptions or correlations are sufficient, at least as a starting point, to achieve sufficiently good rejection of the disturbances. And with a

closed-loop the system can react to changes of system parameters without the need for manual modifications.

That is why in the following chapter such a closed-loop control system for the FLUTE electron gun is treated in detail. In order to design an appropriate control system, some prerequisites are needed. First, the present disturbances are analyzed and metrics to later measure the favorable effect of the control system are defined. Then, ways of connecting the finished control system to the existing hard- and software are shown. With this preliminary work done, the control system can be designed. After the design process, the whole system is simulated on a computer to verify the solution, before the control system is implemented and tested with the real hardware, i.e. at FLUTE.



## 2. Theoretical Framework

### 2.1 Relativistic Electrons

In a (linear) particle accelerator, charged particles, such as electrons, are accelerated to increase their total energy over their energy at rest.

Compared to heavier particles, such as protons ( $m_p = 938.27 \text{ MeV}/c^2$ ), electrons are light particles ( $m_e = 0.511 \text{ MeV}/c^2$ ). They are typically accelerated to speeds comparable to the speed of light to achieve kinetic energies usable for scientific experiments. For this reason, relativistic mechanics are needed to describe their movements.<sup>1</sup> [Hinterberger1997]

With the speed of light  $c = 2.997\,924\,58 \times 10^8 \text{ m s}^{-1}$  and the particle velocity  $v$ , it is common to define [Wangler2008]:

$$(\text{normalized velocity}) \quad \beta = \frac{v}{c} \quad (2.1)$$

$$(\text{relativistic mass factor}) \quad \gamma = \frac{1}{\sqrt{1 - \beta^2}} \quad (2.2)$$

$$(\text{relativistic momentum}) \quad p = \gamma mv \quad (2.3)$$

The total energy of a particle is [Hinterberger1997]

$$W = \sqrt{(mc^2)^2 + (pc)^2}, \quad (2.4)$$

with the rest energy  $mc^2$  and the kinetic energy is

$$T = W - mc^2. \quad (2.5)$$

With electrons (mass  $m_e$ ) leaving the FLUTE electron gun with a momentum of  $p = 7 \text{ MeV}/c$ , this is equivalent to

$$W = \sqrt{(m_e c^2)^2 + (pc)^2} = 7.0186 \text{ MeV} \quad (2.6)$$

$$\gamma = \frac{W}{m_e c^2} = 13.7351 \quad (2.7)$$

$$\beta = \sqrt{1 - \gamma^{-2}} = 0.9973 \quad (2.8)$$

The force exerted on an electron (charge  $q$ ) by an electric field  $E$  and a magnetic flux density  $B$  is given by the Lorentz force [Hinterberger1997]

$$F_L = q(E + v \times B). \quad (2.9)$$

---

<sup>1</sup>As relativistic mechanics are a super set of classical mechanics, the equations also apply for slower particles.

As the force due to the magnetic field is scaled with the velocity  $v$ , it is technically more viable to steer and focus a relativistic electron with magnets instead of electric fields. This is mainly because strong magnetic fields are easier to create and work with than high electric fields, which are difficult to handle. [Krieger2013]

Equation 2.9 also motivates the use need for stable electron energies. Since the velocity depends on the energy, the force on the electrons is also a function of their energy. The magnetic fluxes of electromagnets can only be changed slowly compared to the repetition rate of an accelerator, so corrections to  $B$  are not possible, which is why  $v/E$  needs to be stable for a stable trajectory/focusing.

## 2.2 Signal Analysis

### 2.2.1 Auto- and Cross-Covariance

The *cross covariance* between two stochastic processes  $x[n]$  and  $y[n]$  is a measure of the similarity between  $x[n]$  at index  $n_1$  and  $y[n]$  at index  $n_2$  and is defined as

$$r_{xy}[n_1, n_2] = E \{ (x[n_1] - \mu_x[n_1])(y[n_2] - \mu_y[n_2])^* \}. \quad (2.10)$$

For the special case of  $y[n] := x[n]$ ,  $r_{xx}[n_1, n_2]$  is called *auto covariance* and is a measure of self similarity of  $x[n]$ . [Park2017]

The processes  $x[n]$  and  $y[n]$  are called *Wide Sense Stationary (WSS)* if the following two properties hold. [Park2017] First, their means  $\mu_\xi[n]$  are constant, i.e. they do not depend on the sample index:

$$\mu_x[n] = \mu_x \quad (2.11)$$

$$\mu_y[n] = \mu_y \quad (2.12)$$

Also, the auto covariance does not depend on the absolute sample indices  $n_1$  and  $n_2$ , but merely on the difference between them:

$$r_{xy}[n_1, n_2] = r_{xy}[m], \quad \text{with: } m := n_2 - n_1 \quad (2.13)$$

If both processes in Equation 2.10 are WSS, Equation 2.10 simplifies to

$$r_{xy}[m] = E \{ (x[n] - \mu_x)(y[n-m] - \mu_y)^* \}. \quad (2.14)$$

For the auto covariance both means are identical and can be moved outside the expectation operator:

$$r_{xx}[m] = E \{ (x[n])(x[n-m])^* \} - \mu_x^2. \quad (2.15)$$

When analyzing signals, the stochastic processes are often unknown and only one realization  $x[n]$  is known. But if the process generating  $x[n]$  is (*weakly*) *ergodic*, then one realization is enough to determine the auto covariance of the process. [Puente2019] Then the auto covariance can be estimated with

$$\hat{r}[m] = \frac{1}{N} \sum_{n=m+1}^N x[n] x^*[n-m] \quad m \in [0, N-1]. \quad (2.16)$$

### 2.2.2 Estimating the Spectrum of a Stochastic Process

For a deterministic, time-discrete signal  $x[n] \in \mathcal{L}_1$ , the Discrete Fourier Transform (DFT) exists (see [Lapidoth2019]) and is defined as

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn} \quad k, n \in [0, N-1], \quad (2.17)$$

using  $k = \frac{N}{2\pi} \omega = N f$  as the independent, discrete frequency variable. From the complex sequence  $X[k]$ , often only the magnitude (or energy) is of greater interest while the phase information are neglected. Therefore,  $S_{xx}$  is defined as

$$S_{xx} = |X[k]|^2, \quad (2.18)$$

called the *Energy Spectral Density (ESD)*.

If  $x[n]$  is the realization of a stochastic process, then it is of random nature rather than deterministic. Because realizations of physical processes do not possess finite energy, they are not in the  $\mathcal{L}_1$  set and their DFT is not defined. [Stoica1997]

In this case instead of an energy spectral density, the spectrum of the average power of the process, called the *Power Spectral Density (PSD)*, can be used instead. To compute the PSD, there are two possibilities:

$$\Phi_{xx}[k] = \sum_{m=-\infty}^{\infty} r[m] e^{-j\frac{2\pi}{N}km} \quad (2.19)$$

$$\Phi_{xx}[k] = \lim_{N \rightarrow \infty} E \left\{ \frac{1}{N} \left| \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn} \right|^2 \right\} \quad (2.20)$$

When assuming  $r[m]$  decays “fast enough”, i.e.

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{m=-N}^N |m| |r[m]| = 0 \quad (2.21)$$

then Equation 2.19 and Equation 2.20 are equal. [Stoica1997]

For measured data however neither equations can be used directly. For Equation 2.19 the auto covariance sequence  $r[m]$  is unknown. But it could be estimated with Equation 2.16. In case of Equation 2.20 it is not possible to evaluate the limit, because only finite length data can be sampled and also the expectation can not be computed since in general there is only one realization available. Both operations can be neglected when doing an estimation.

With these practical changes in place, Equation 2.19 and Equation 2.20 become

$$\hat{\Phi}_{c,xx}[k] = \sum_{m=-(N-1)}^{N-1} \hat{r}[m] e^{-j\frac{2\pi}{N}km} \quad (\text{Correlogram}) \quad (2.22)$$

$$\hat{\Phi}_{p,xx}[k] = \frac{1}{N} \left| \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn} \right|^2 \quad (\text{Periodogram}). \quad (2.23)$$

Both methods yield equal results, if  $r[m]$  is estimated with the biased estimator  $\hat{r}[m]$  in Equation 2.16 in contrast to the unbiased estimator (compare [Stoica1997])

$$\hat{r}_{\text{unbiased}}[m] = \frac{1}{N-m} \sum_{n=m+1}^N x[n] x^*[n-m] \quad m \in [0, N-1]. \quad (2.24)$$

[Rowell2008] shows one key weakness of the unmodified periodogram method in Equation 2.23: The variance does not decrease significantly with more samples  $N$ . Instead, the variance of the periodogram for each frequency approaches the square of the actual PSD:

$$\lim_{N \rightarrow \infty} \text{Var} \left\{ \hat{\Phi}_{p,xx}[k] \right\} = \Phi_{xx}^2[k] \quad (2.25)$$

Furthermore, the periodogram/correlogram suffer from the smearing and leakage effects because the limited length of the data samples always causes an implicit windowing, thus reducing frequency resolution.

There are several popular methods that improve on the periodogram/correlogram concepts:

**Blackman-Tukey:** Because of the poor accuracy of  $\hat{r}[m]$  for  $k \approx N$  in the definition of  $\hat{\Phi}_{c,xx}[k]$  and the bigger the  $N$ , the more small errors in  $\hat{r}[m]$  sum up, truncating/windowing of  $\hat{r}[m]$  with  $w[k]$  (length  $M$ ) can be beneficial for the accuracy of the estimation.

$$\hat{\Phi}_{BT,xx}[k] = \sum_{m=-(M-1)}^{M-1} w[k] \hat{r}[m] e^{-j \frac{2\pi}{N} k m} \quad (2.26)$$

The choice of the window  $w[k]$  trades frequency resolution for variance and smearing for leakage reduction. [Stoica1997]

**Barlett:** The Barlett method reduces the variance of the periodogram by splitting the  $N$  data samples in  $Q = N/M$  blocks and averaging together the sub-periodograms:

$$\hat{\Phi}_{q,xx}[k] = \frac{1}{M} \left| \sum_{n=0}^{M-1} x_q[n] e^{-j \frac{2\pi}{M} k n} \right|^2 \quad (2.27)$$

$$\hat{\Phi}_{B,xx}[k] = \frac{1}{Q} \sum_{q=1}^Q \hat{\Phi}_{q,xx}[k] \quad (2.28)$$

The variance of the estimation scales with  $Q$  (see [Rowell2008]):

$$\text{Var} \left\{ \hat{\Phi}_{B,xx}[k] \right\} = \frac{1}{Q} \Phi_{xx}^2[k] \quad (2.29)$$

**Welch:** The Welch method combines splitting the data into  $Q$  segments with windowing each segment and allowing the segments to overlap. With  $P = 1/M \sum_{n=0}^{M-1} |w[n]|^2$  being the “power” of the window, the Welch method is computed as

$$\hat{\Phi}_{s,xx}[k] = \frac{1}{MP} \left| \sum_{n=0}^{M-1} x_s[n] e^{-j \frac{2\pi}{M} k n} \right|^2 \quad (2.30)$$

$$\hat{\Phi}_{W,xx}[k] = \frac{1}{Q} \sum_{s=1}^Q \hat{\Phi}_{s,xx}[k]. \quad (2.31)$$

Compared to the Barlett method, the overlapping of up to 50 % (see [Welch1967]) allows increasing  $Q$ , thus reducing the variance.

$$\text{Var} \left\{ \hat{\Phi}_{W,xx}[k] \right\} = \frac{1}{Q} \Phi_{xx}^2[k] \quad (2.32)$$

## Spectrogram

If a stochastic process or a signal as a realization of that process  $x[n]$  is not WSS, one possibility to analyze and display the spectral content is the use of the Short Time Fourier Transform (STFT) and the spectrogram, which is a two dimensional power spectral density function mapping frequency and time to a third coordinate like height, intensity or color.

To calculate the spectrogram, the signal is split into segments with the sliding window  $w[n - m]$  for which duration the signal is assumed to be stationary. For each segment at time index  $m$ , the periodogram is calculated according to

$$\hat{\Phi}_{xx}[k, m] = \frac{1}{N} \left| \sum_{n=0}^{N-1} w[n - m] x[n] e^{-j \frac{2\pi}{N} k n} \right|^2. \quad (2.33)$$

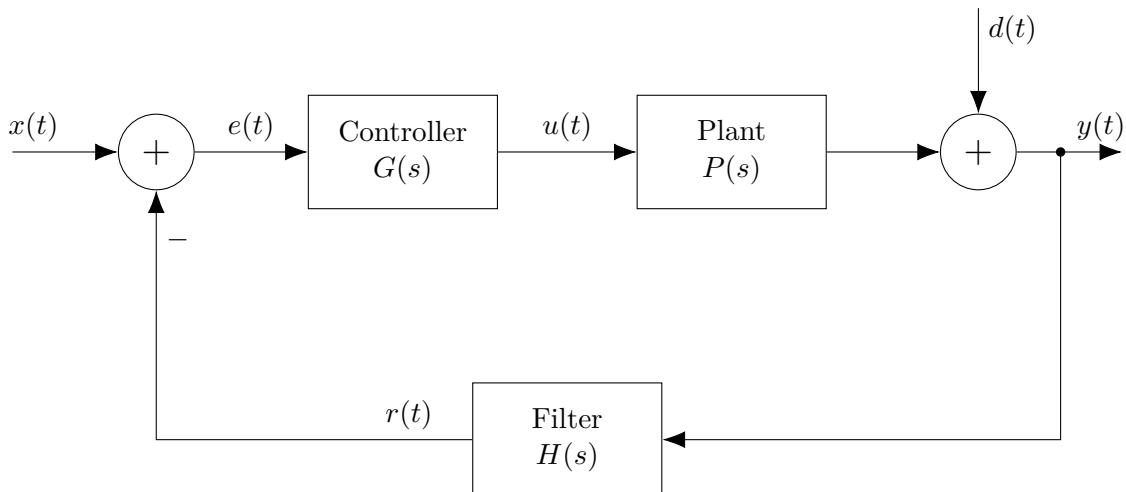
## 2.3 Feedback Control Systems

Feedback control systems are used to control a dynamic system (also called a plant) in such a way that its output  $y(t)$  follows a certain input  $x(t)$  and disturbances on the output  $d(t)$  are rejected. The general structure of a closed-loop control system is shown in Figure 2.1. To achieve sufficient tracking of the input and stabilization of the output, a controller  $G(s)$  uses the error  $e(t)$  to control the plant  $P(s)$  accordingly. The error is defined as

$$e(t) = x(t) - r(t) = x(t) - [y(t) * h(t)] \quad (2.34)$$

with  $h(t)$  being the inverse Laplace transform of the filters transfer function  $H(s)$ . The signal  $r(t)$  is the output of the measurement filter  $h(t)$  or  $H(s)$ . It is most commonly a lowpass filter used to reject high-frequency noise on the system output  $y(t)$ .

Feedback control systems, or closed-loop systems, are to be differentiated from open-loop systems, in which there is no return path, so they cannot compensate for *unknown* disturbances. If  $d(t)$  is known  $\forall t$ , then an open loop system would be possible and any errors could simply be compensated. But for real world application this approach is only usable for crude control tasks or if the system is very well understood or isolated from its surroundings.



**Figure 2.1:** General structure of a time-continuous feedback control system

### 2.3.1 Disturbance Rejection and Input Tracking

Disturbance rejection and input tracking are two important characteristics to evaluate a stable controller. To calculate them, the block diagram in Figure 2.1 and the Laplace transform of the inputs/outputs are used.<sup>2</sup>

To calculate how the output  $y(t)$  depends on the input  $x(t)$ , the input tracking transfer function can be used. [Foellinger2016] It is calculated as the transfer function  $F_T = Y(s)/X(s)$  by setting  $d(t) = 0$ :

$$Y(s) = G(s)P(s)E(s) \quad \text{with: } E(s) = X(s) - H(s)Y(s) \quad (2.35)$$

$$\Leftrightarrow Y(s)[1 + G(s)P(s)H(s)] = G(s)P(s)X(s) \quad (2.36)$$

$$\Leftrightarrow F_T := \frac{Y(s)}{X(s)} = \frac{G(s)P(s)}{1 + G(s)P(s)H(s)}. \quad (2.37)$$

On the other hand, the transfer function  $F_{DR} = Y(s)/D(s)$  can be used to describe the systems response to a disturbance. [Foellinger2016] It is defined by setting  $x(t) = 0$  and calculating

$$Y(s) = G(s)P(s)E(s) + D(s) \quad \text{with: } E(s) = -H(s)Y(s) \quad (2.38)$$

$$\Leftrightarrow Y(s)[1 + G(s)P(s)H(s)] = D(s) \quad (2.39)$$

$$\Leftrightarrow F_{DR} := \frac{Y(s)}{D(s)} = \frac{1}{1 + G(s)P(s)H(s)}. \quad (2.40)$$

### 2.3.2 Stability

The application of a controller to a system is only useful if the resulting system has a stable behavior. One possible definition of stability is the Bounded Input Bounded Output (BIBO) criterion (see [Foellinger2016]):

**Definition 1.** (*BIBO stability*) A Linear Time Invariant (LTI) system is said to be BIBO stable if for some  $M, N \in \mathbb{R}^+$ , the response to a bounded input  $|u(t)| \leq M$  results in a bounded output  $|y(t)| \leq N$ .

For a given control system, one way to analyze its stability is to plot the locus  $z = F_o(s = j2\pi f)$  of the open loop frequency response

$$F_o(s) = G(s)P(s)H(s) \quad (2.41)$$

from  $f = 0$  to  $f = \infty$  and using the Nyquist stability criterion. For the special case of a stable open loop  $F_o(s)$ <sup>3</sup> the Nyquist stability criterion can be stated as (see [Foellinger2016])

**Definition 2.** (*Nyquist stability criterion*) If the open loop  $F_o(s)$  is stable, then the closed loop is stable if  $z = F_o(s = j2\pi f)$  does not go through or encircles  $z = (-1, 0j)$ .

---

<sup>2</sup>The Laplace transform of a function in time  $f(t)$  is written as  $F(s) = \mathcal{L}\{f(t)\}$ .

<sup>3</sup>The stability of  $F_o(s)$  can often easily be determined from the block diagram.

## 2.4 Metrics to Quantify the Stability of a Signal

“Stability” can have different meanings depending on the context. In case of signal processing, a signal is usually said to be *stable* if it has only little variation around its mean or some target value, i.e. the mean has to be constant and the variance stays below some threshold. Stability is not to be confused with stationarity, which requires the mean, the variance and the autocorrelation to stay constant over time. [Guthrie2020] To express stability as a single numerical value, there are several possibilities, some are described in the following.

### Relative Standard Deviation

This measures the stability as the standard deviation but related to the mean value to make it comparable to other quantities with different scaling or units.

The relative, or percentual, standard deviation of the stationary stochastic process  $X$  is defined using the mean  $\mu_X$  and the standard deviation  $\sigma_X$  as

$$\%STD_X := \frac{\sigma_X}{\mu_X}. \quad (2.42)$$

In general, especially if  $X$  is non-stationary (see Equation 2.11),  $\%STD_X$  depends on the absolute time  $t$  and the window size  $T$  for which the process is assumed to be stationary:

$$\%STD_X = \%STD_X(t, T) \quad (2.43)$$

In that case, for a fixed window size  $T = T_0$ , a mean percentual standard deviation can be computed with

$$\%STD_X(T = T_0) = \frac{1}{N} \sum_{n=0}^{N-1} \%STD_X(t_n, T_0). \quad (2.44)$$

This assumes discrete time steps  $t_n$ ,  $n \in [0, N - 1]$ .

### Mean Squared Error

A similar measure to the percentual standard deviation is the Mean Squared Error (MSE), if, instead of the mean  $\mu_X$ , a fixed target  $x_t$  is used.

The mean squared error sums up the squared errors  $(x[n] - x_t[n])^2$  of  $x[n]$  from a set value  $x_t$ . To remove the effect of the length of the data sequence, the sum is divided by the length of the sequence  $N$ :

$$MSE_x := \frac{1}{N} \sum_{n=0}^{N-1} (x[n] - x_t)^2 \quad (2.45)$$

### Relative Power of Most Prominent Noise

This novel approach compares the power of the most prominent noise power source  $P_{\text{noise, max}}$  of the signal  $x$  with the total power  $P_x$ :

$$MPN_x := \frac{P_{\text{noise, max}}}{P_x} \quad (2.46)$$

Compared to the relative standard deviation  $\%STD(t, T)$ , this method has the advantage of being time-independent and keeping periodic noise structures:

### Comparison

To make  $\%STD(t, T)$  independent of time, it could be tempting to choose  $T = T_{\text{total}}$  with  $T_{\text{total}}$  being the time span of the data set. But because of the averaging effect of the standard deviation estimator, this washes out the maxima and thus could lead to an under-estimation of  $\%STD$ , so an over-estimation of the stability.

The MSE uses the whole data set, but due to normalizing maxima can also be hidden.

Periodogram-based method have the advantage that for a fixed target value  $x_t$ , they are very easy to interpret: All power, that is not in the  $f = 0$  frequency bin, is definitely noise. Integrating over these noise bins yields the error power comparable to the MSE. The described MPN metric has the advantage of showing the largest contributor to the noise and allows for an easy validation of a controllers success on this noise component.

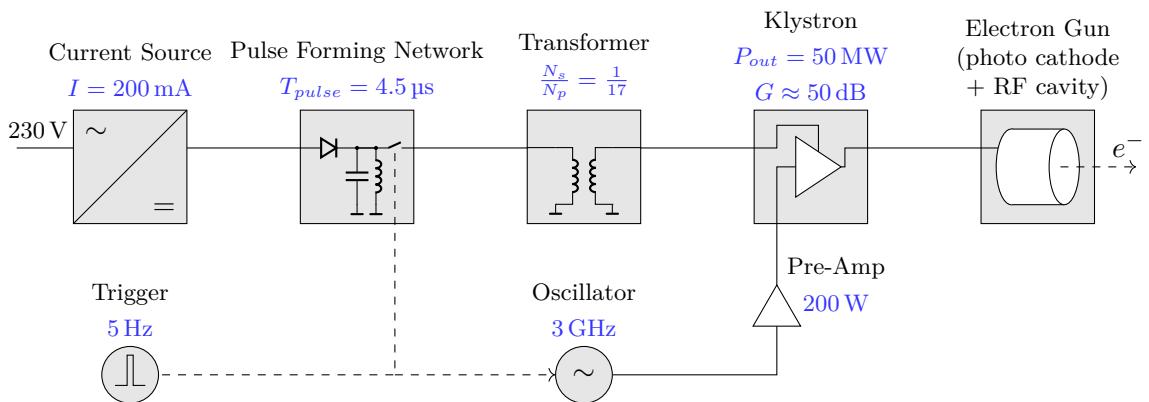
### 3. The Stability of FLUTE, the Electron Gun and Proposed Stabilizing Solution

This chapter deals with the electron gun of FLUTE and its power supply. Then, based on fundamental equations of electron gun's microwave cavity, the dependence of the electron energy from the RF supply is derived, which motivates why the RF supply should be stable. Finally, a solution to stabilize the RF is proposed.

The electron gun is powered by a 50 MW klystron, a high-power vacuum tube RF amplifier. The input signal for the klystron is a  $2.998\text{55 GHz} \approx 3\text{GHz}$  harmonic oscillator pre-amplified to 200 W. The supply input is generated by a pulse forming network and a transformer. The pulse forming network mainly consists of capacitors to store electrical energy and is charged with a constant current source. The connection of these devices is shown in Figure 3.1.

A 5 Hz master clock (“trigger”) is used to switch on the output of the pulse forming network to the klystron and the oscillator every 0.2 s for 4.5  $\mu\text{s}$ . During this time, the laser could also be triggered causing a stimulated emission of an electron bunch from the cathode. But even without the laser being active, powering the electron gun with the klystron generates an electron beam through thermionic emission of electrons. This undesired effect is called *dark current*.

The current source to charge the pulse forming network is powered by mains voltage. This makes it susceptible to noise on the mains and also causes slowly time varying drifts of the klystron power due the pulse forming network being triggered at different relations to the mains 50 Hz. This issue has been remedied in [Nasse2019] by adding synchronization to the mains phase.



**Figure 3.1:** Schematic of the FLUTE RF system

### 3.1 The Electron Gun

The electron gun of FLUTE was originally designed and operated in CTF II at Conseil Européen pour la Recherche Nucléaire (CERN). [Schuh2014] It is of the “BNL type” (see [Batchelor1988], based on the original design by [fraser1987]) and was developed at CERN. [Bossart:288412]

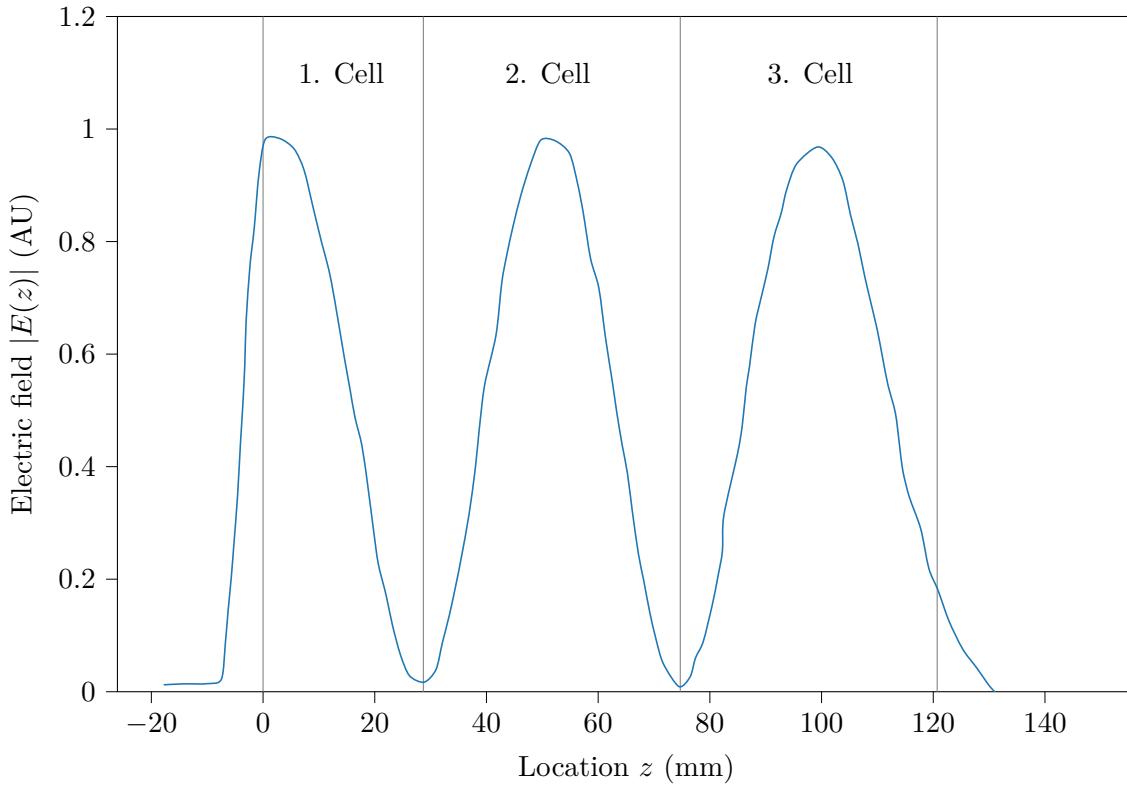
The gun is made up of a 2.5 cell microwave cavity with a removable copper cathode embedded in the cone shaped back at the end of the half cell (see Figure 3.3). Cooling is achieved with a two-stage water cooling system: A temperature control unit uses a short water circuit to cool the gun while itself uses a heat exchanger to a bigger outside climate unit.

Applying RF power to the cavity through the hole-coupled wave guide causes a standing wave inside the cavity. Because of the cavity’s dimensions, only the fundamental mode  $\text{TM}_{010}$  is excited, for which the relation between resonance frequency  $f_{010}$  and radius  $a$  of the cavity is given by

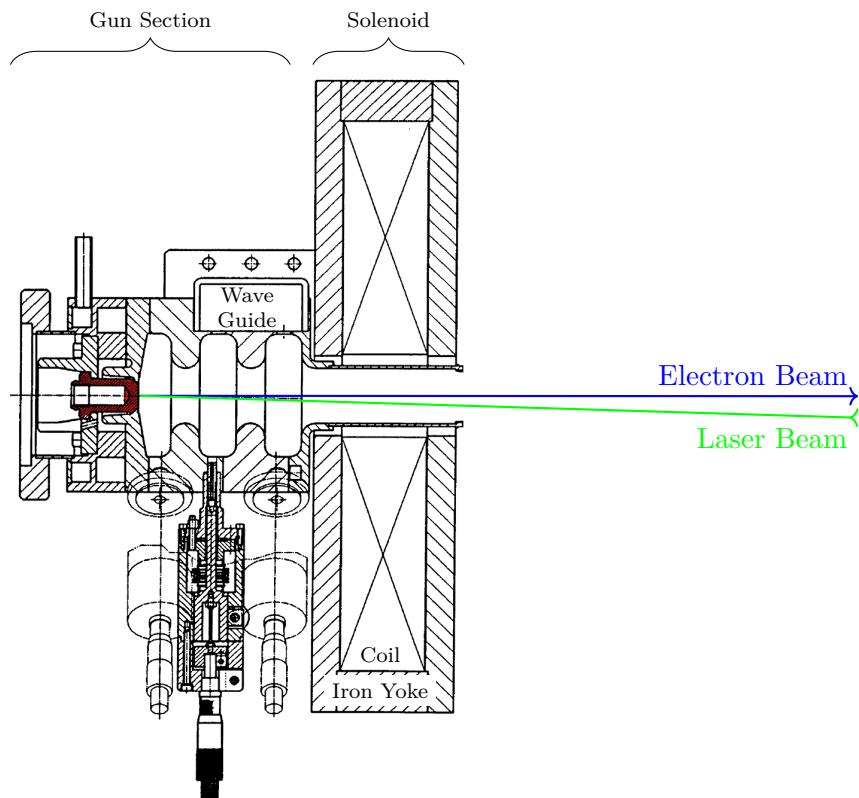
$$\frac{f_{010}}{2\pi} = \frac{2.405 \cdot c}{a}. \quad (3.1)$$

For the  $\text{TM}_{010}$  mode there is only an electrical field in the  $z$  direction, i.e. along the beam axis. This  $E_z(z)$  field is used to accelerate the electrons. For the FLUTE gun,  $E_z(z)$  has been measured in [Bossart:clic], see Figure 3.2. These measurements are also verified in [Schuh2014].

To tune the resonance frequency  $f_{010}$ , which depends on the cavity’s radius  $a$ , to the target design frequency of 2.998 55 GHz, two methods are used. The cavity is equipped with piston tuners that allow changing the geometries of each cell slightly. Because of the expansion and contraction of the copper body due to temperature changes, the set-point of the water cooling system can also be changed to alter the cavity geometry.



**Figure 3.2:** Plot of the electrical field in  $z$  direction over the length of the gun cavity (redrawn from [Bossart:clic] using geometrical measurements from [Hoeninger2014])



**Figure 3.3:** Cross section drawing of the electron gun together with the solenoid (which is used for focusing the electron beam) showing the photo-cathode (red) and the electron and laser beam trajectories (modified version from [Bossart:clic] and [Bossart:288412])

### 3.2 Relation between RF power and Electron Energy

A standing wave inside a RF cavity for a  $TM_{010}$  mode can be written as

$$E_z(z, t) = E(z) \cos(\omega t + \phi). \quad (3.2)$$

The time  $t$  has to be expressed in terms of the electron velocity  $v(z)$  as

$$t = t(z) = \int_0^z \frac{dz}{v(z)}, \quad (3.3)$$

which is the arrival time of the electron at location  $z$ .

If moving through an accelerating gap of length  $L$  inside a cavity, an electron with charge  $q$  gains the energy

$$\Delta W = q \int_{-L/2}^{L/2} E(z) \cos(\omega t(z) + \phi) dz \quad (3.4)$$

This can be rewritten as

$$\Delta W = qV_0T \cos(\phi) \quad (3.5)$$

using the axial RF voltage

$$V_o := \int_{-L/2}^{L/2} E(z) dz \quad (3.6)$$

and the travel time factor  $T$ . [Wangler2008]

With the *shunt impedance*  $R_s$ , the axial RF voltage can be brought into relation with the RF power which needs to be induced into the cavity to compensate losses in the non-perfect conducting walls and power lost to the electron beam. [burtRF]

The shunt impedance is defined as

$$R_s = \frac{V_0^2}{P_{RF}} \quad (3.7)$$

Equation 3.5 and Equation 3.7 show that the RF supply has a great impact on the electron energy, so it needs to be stable.

Additionally, there is the so called *R over Q*, defined as

$$\frac{R}{Q} = \frac{(V_0T)^2}{\omega U} \quad \text{with: } R = R_s T^2 \text{ (effective shunt impedance)} \quad (3.8)$$

using the total stored electromagnetic energy  $U$  and the quality factor  $Q = \omega U / P_{RF}$ .

This shows the gained energy also depends on the properties of the cavity.

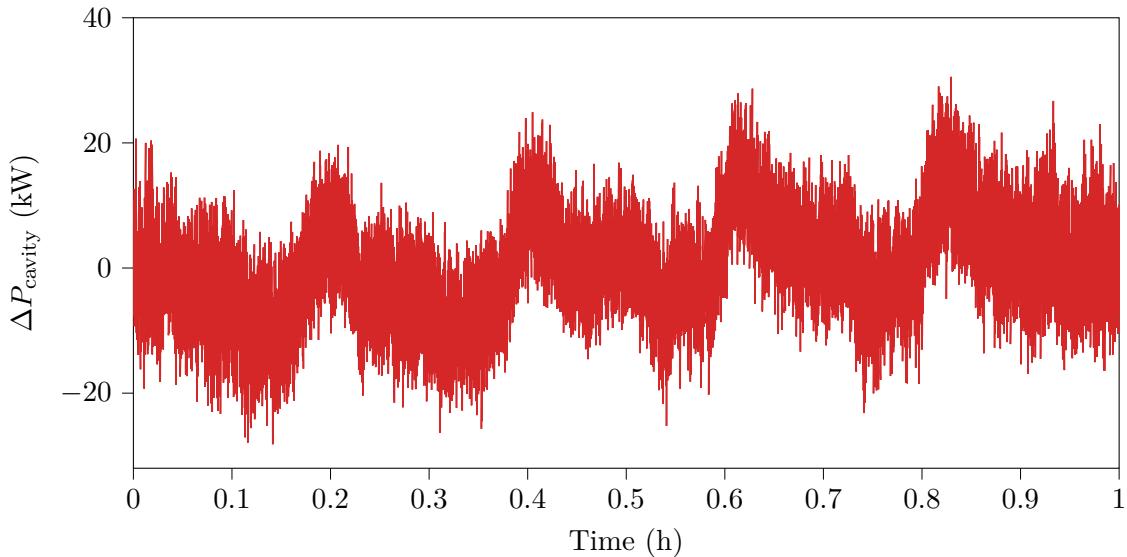
### 3.3 Current RF Stability and Proposed Solution

To get an overview of the current stability of the cavity RF power, the deviation of the cavity power process value (Experimental Physics and Industrial Control System (EPICS)<sup>1</sup> Process Variable (PV) name F:RF:LLRF:01:GunCav1:Power:Out Value =:  $P_{cavity}$ ) from its mean is plotted over one hour, see Figure 3.4.

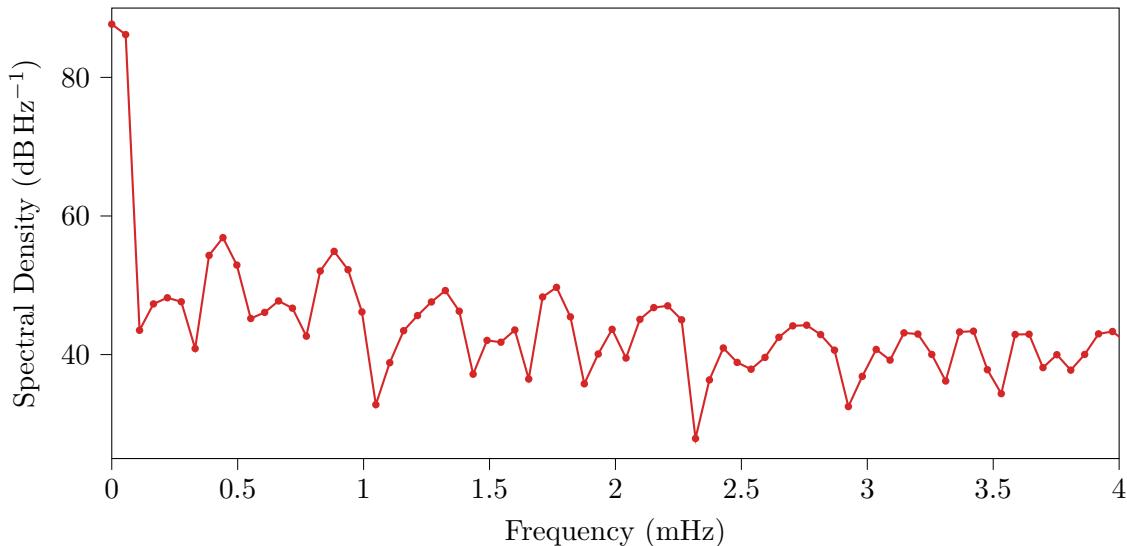
With the metrics defined in section 2.4, over a time interval of  $T = 5$  h, the metrics  $\%STD_{P_{cavity}} = 0.15\%$ ,  $MSE_{P_{cavity}} = 38.54$  and  $MPN_{P_{cavity}} = 0.01677$  are calculated.

---

<sup>1</sup>See section 4.1



**Figure 3.4:** Deviation of the cavity RF power over the course of one hour



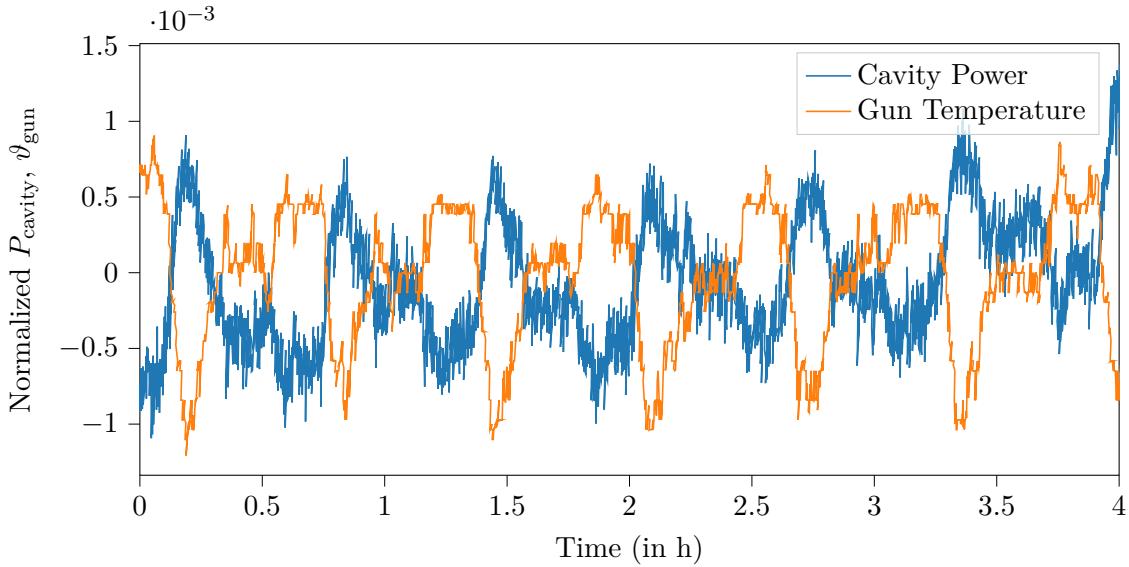
**Figure 3.5:** Periodogram of Figure 3.4; calculated using the Welch method

By utilizing simulation data and approximate analytical calculations, it is known that for THz generation using the chicane at the end of the compressor (see chapter 1) and to generate chirped THz pulses for cSTART, a stability of  $\%STD_{P_{\text{cavity}}} = 0.10\%$  would suffice. For time resolved THz experiments, a higher stability of  $\%STD_{P_{\text{cavity}}} = 0.01\%$  is necessary. [Nigel2021]

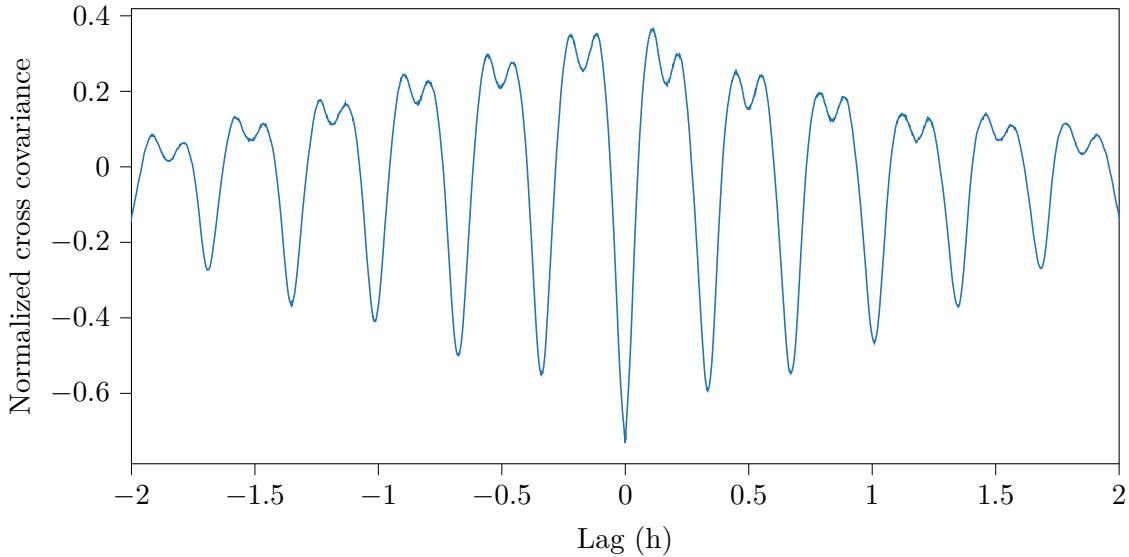
As this is a very demanding task, the goal at the moment is

$$\%STD_{P_{\text{cavity}}} \stackrel{!}{\in} [0.01\%, 0.10\%]. \quad (3.9)$$

From the time plot in Figure 3.4 and the periodogram in Figure 3.5 it becomes clear that there is random white noise, but also a periodic part and a slow drift in the signal. While it is not possible to counteract the random fluctuations by any means of the control system developed in this thesis, it is possible to compensate for the slower disturbances that have effects over several pulses.



**Figure 3.6:** Cavity RF power and electron gun temperature in a normalized plot



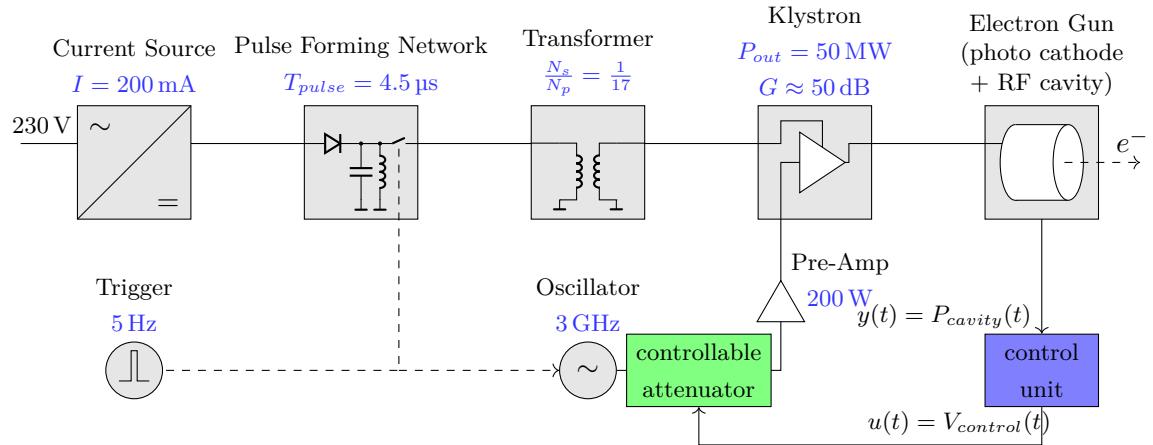
**Figure 3.7:** Normalized cross covariance of the signals in Figure 3.6

To get more insight into these slowly changing disturbances, the time signals of several available sensors in EPICS are compared to the cavity RF power. This shows similar trends in both the cavity RF power  $P_{\text{cavity}}$  and the electron gun temperature  $\vartheta_{\text{gun}}$ . To analyze them in more detail, the signals are both normalized to zero mean and unity standard deviation as they have different units and would be otherwise difficult to compare (see Figure 3.6).

To quantify their relation, the normalized cross covariance is used. It is calculated by using Equation 2.10 with the normalized cavity RF power and the normalized electron gun temperature and the result is shown in Figure 3.7. This shows the two signals are anti-similar ( $\rho = r_{\text{norm}}(0) = -0.7$ ) for zero lag, i.e. no shift in time. This suggests a strong relation between them.

Hence in the next chapters, the control system is developed to counteract these noise components, especially the changes in the electron gun temperature.

To be autonomous from existing components, the control system should be made up with



**Figure 3.8:** Schematic of the FLUTE RF system with the proposed control unit and the controllable RF attenuator added

a controllable RF attenuator added to the existing RF system. This way there is no modification to the proprietary Low Level RF (LLRF)<sup>2</sup> necessary. With the addition of the control unit (see Figure 3.8), which is designed in later chapters and contains the controller  $G(s)$  and the filter  $H(s)$ , a closed-loop for feedback control is formed.

<sup>2</sup>The LLRF is visualized as only the oscillator in Figure 3.8 and Figure 3.1 but contains also its own feedback system and a vector modulator



# 4. Interfacing FLUTE

This chapter covers methods on interfacing the FLUTE accelerator, that is how to read diagnostic measurements into the control system from FLUTE and how to influence the electron acceleration appropriately to achieve stabilization.

In this chapter *input* and *output* refer to the view from the control system.

## 4.1 Inputs

FLUTE uses the *EPICS*[**Dalesio1991**] for control of various parts of the accelerator, to archive time series data and to build user interfaces via *Control System Studio (CSS)*[**CSS2021**], a development studio based on the Java Integrated Development Environment (IDE) Eclipse. [**Mexner2018**]

EPICS offers client/server and publish/subscribe paradigms to access data in so called PV through channels. Modules are usually written in the C programming language. To ease the access to EPICS channels in programs written with the Python language, the package *PyEpics*[**Newville2019**] can be used. Since all data of interest as input for the control system can be extracted through an EPICS channel, the next section deals with using PyEpics to obtain the data.

### 4.1.1 Accessing EPICS channels in Python with PyEpics

Before usage, PyEpics needs to be installed, e.g. with `pip3 install pyepics` from the *PiPy* repository. If the computer running the Python code can reach the EPICS Channel Access (CA) repeater on the machine network, the connection is established automatically in the background. To get a channel value asynchronously, i.e. at an arbitrary time, the function `caget(pvname)` can be used with the name of the desired process value, see Listing 4.1.

---

**Listing 4.1:** Using `caget()` to get the value of an EPICS process value

```
1 from epics import caget
2 print(f"Cavity RF power: {caget('F:RF:LLRF:01:GunCav1:Power:Out')}")
```

---

Another way is to setup a channel object and create a subscription with an user defined callback function that is executed each time the process variable changes. This implements synchronous access to the PV and can be compared to an interrupt rather than polling the variable as in Listing 4.1.

For a non trivial example see Listing 4.2. In this program, the time differences between new values and their statistics are printed to the console.

---

**Listing 4.2:** Using a user defined callback function to access an EPICS process value

```
1 from epics import ca
2 import time
```

---

---

```

3 import numpy as np
4
5 dts=np.array([])
6 lastCalled=time.time()
7
8 def call(pvname, value, **kwargs):
9     global lastCalled, dts
10    now=time.time()
11    dt=now-lastCalled
12    lastCalled=now
13    dts=np.append(dts, dt)
14
15 chid=ca.create_channel("F:RF:LLRF:01:GunCav1:Power:Out")
16 _ , _ , eventID=ca.create_subscription(chid, callback=call, use_time=True)
17
18 while(True):
19     time.sleep(2)
20     print(f"N:{len(dts)},mean:{np.mean(dts)},min:{np.min(dts)},max:{np.max(dts)},std:{np.std(dts)}")
```

---

#### 4.1.2 Properties of the Available Process Variables

In this section some process variables that may be used as inputs for the control system are analyzed. These are:

- **F:RF:LLRF:01:GunCav1:Power:Out Value:** The RF power measured in the half cell of the gun cavity
- **F:AX:DAQDT:01:1:Wave:05:Sample Value:** The charge measured with a Faraday cup (RadiaBeam Technologies FARC-04 [**radiabeamFaradayCups**]) and amplified with a charge sensitive amplifier (PCB 421A25 [**pcbsynotechPCB421A25Charge**])
- **F:INJ-1:Gun:01:Temperature:Body Value:** The body temperature of the electron gun

In Figure 4.1 all three are plotted for a duration of 15 min without any user interaction and the system being in steady state operation. Based on these plots, the quantization steps of these signals and their sample distances are analyzed.

The quantization steps analysis in Table 4.1 shows that the three PVs not have comparable numbers of unique values  $N_{\text{unique}}$  and quantization steps height  $q_{\text{avg}}$ . If the quantization steps are normalized to the PV means, it is obvious, the cavity RF power is quantized about four to five times finer than the gun temperature or the charge signal from the Faraday cup. This shows especially in plots where different PVs are plotted together.

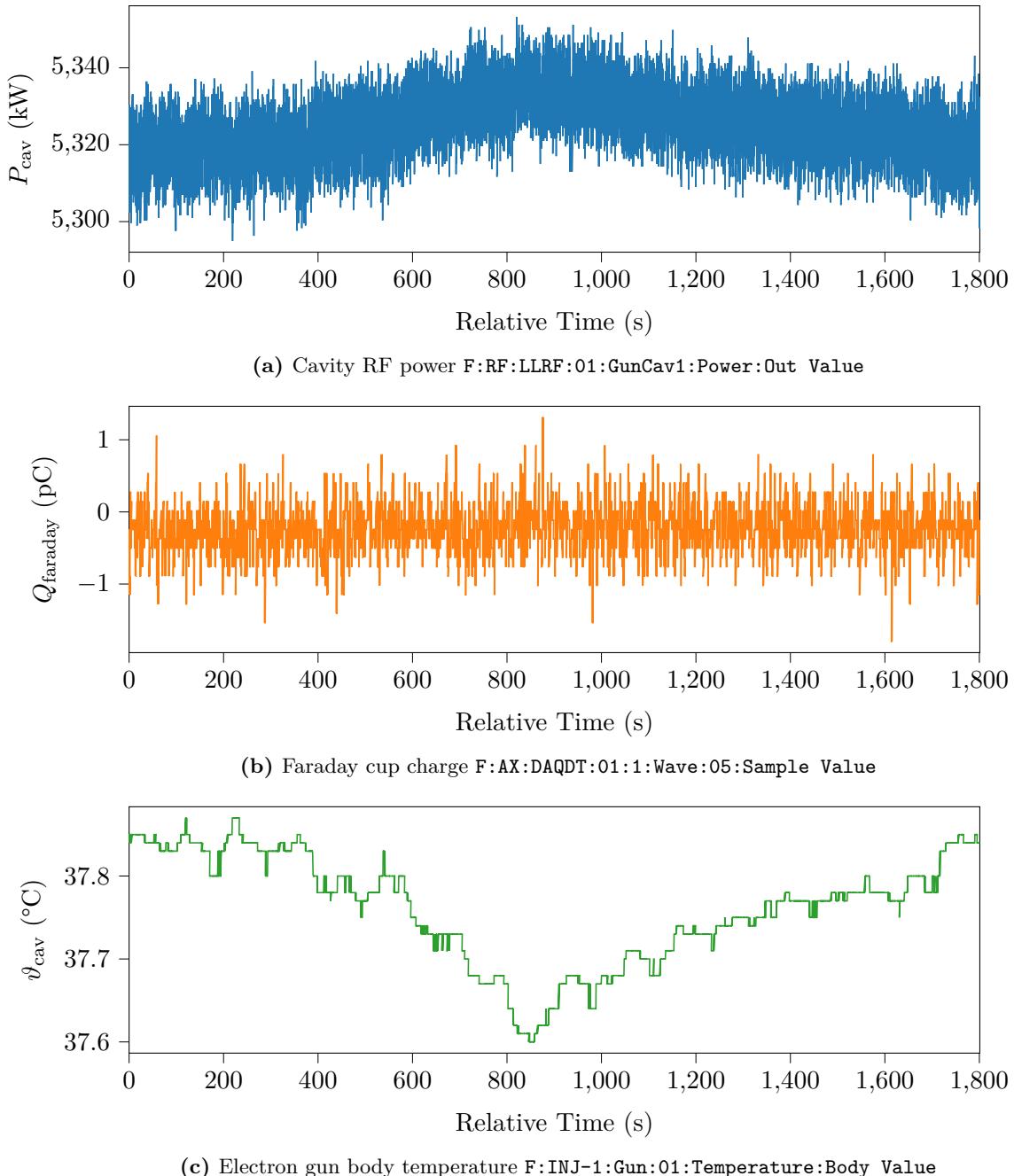
Next the sample times intervals of the process variables are examined if the values are extracted from the EPICS archive using the CSS exporting feature. The differences between the sample times are calculated according to

$$\Delta = t_{n+1} - t_n. \quad (4.1)$$

Then, a histogram with the relative frequency on the y axis is used as an estimator for the probability density function of the sample time differences (see Figure 4.2). The histogram shows that the time series resulting from the export feature are highly unevenly spaced. Thus, the data needs to be converted to have evenly spaced sample times to use common signal processing methods like the DFT or digital LTI filters. For offline analysis of prerecorded data, it can easily be re-sampled to a fixed time grid, e.g. back to the sample time of 0.2 s matching the 5 Hz pulse repetition frequency.

Input data for the control system is obtained with a dedicated clock/scheduler, e.g. an

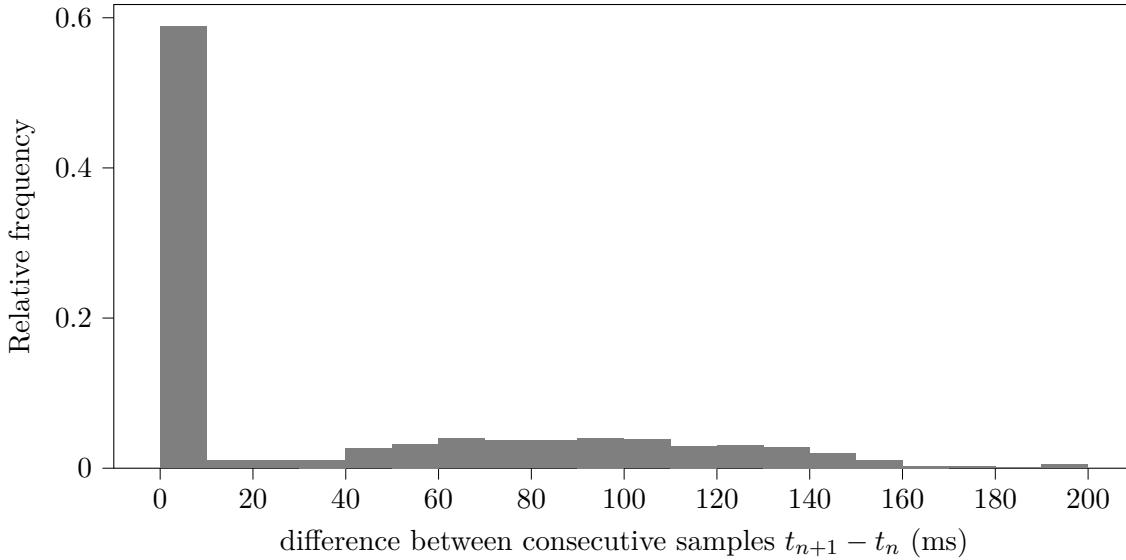
APScheduler or QTimer using `caget()` so the sample time distance depends on the clock or scheduler used.



**Figure 4.1:** Comparing three common FLUTE process variables available in EPICS which shows the difference in quantization step size

**Table 4.1:** Comparing quantization steps

PV	$N_{\text{unique}}$	$q_{\text{avg}}$	$q_{\text{norm}}$
F:RF:LLRF:01:GunCav1:Power:Out Value	84	0.6935	0.011 904
F:AX:DAQDT:01:1:Wave:05:Sample Value	22	0.015	0.045 45
F:INJ-1:Gun:01:Temperature:Body Value	18	0.1294	0.055 55



**Figure 4.2:** Histogram of the sample time intervals  $\Delta$  of the plots in Figure 4.1

#### 4.1.3 Filtering the RF power Signals

In case of the klystron output RF power or the cavity RF power, filtering of the signals is needed to remove zero outliers. These outliers occur if a high voltage arc occurs inside the cavity. This is detected by an optical sensor and the LLRF control system shuts off the RF power for the current pulse (this is called a “breakdown”). These outliers are not representative of the average RF power inside the cavity over multiple pulses and thus would greatly impair the controller performance. For that reason, before any further filtering to remove noise etc., a breakdown removal filter is used (Listing 4.3). In principle, the new power value is checked to be inside a band, the size of which is determined by the mean deviation of the  $N_{\text{filt}}$  previous values and a scaling  $m$ . The percentile differences are used here, as they are robust against outliers (i.e. other breakdowns) in the  $N_{\text{filt}}$  previous values, as opposed to a normal standard deviation. The scaling with  $(2 * 1.2815)^{-1}$  is used to make the mean deviation comparable to a standard deviation.

**Listing 4.3:** Breakdown removal filtering

---

```

1  if(abs(P[i]-np.median(P[i-3*Nfilt:i-Nfilt])) <
2    m*(np.percentile(P[i-3*Nfilt:i-Nfilt],90)-np.percentile(P[i-3*Nfilt:i-Nfilt],10)/(2*1.2815)):
3      P_filt = np.append(P_filt,P[i])
4  else:
5      breakdown_locations_predicted=np.append(breakdown_locations_predicted,i)
6      P_filt = np.append(P_filt,np.median(P[i-3*Nfilt:i-Nfilt]))

```

---

## 4.2 Output: Controllable RF Attenuator

The output signal computed by the control systems has to have a way of influencing the RF power sent to the cavity. This could be done over an EPICS channel (e.g. with the PyEpics function `caput()` to set the value of a process variable via a channel access). However to make it possible to move the control system from a general purpose personal computer to a dedicated digital signal processor, Field-Programmable Gate Array (FPGA) or similar in the future, using a physical device in the signal path is preferred.

### 4.2.1 Defining Requirements

The controllable attenuator should allow to vary the attenuation in a span big enough to counteract typical instabilities on the FLUTE RF power. The set attenuation should be stable. This is especially needed in cases where the control system is not enabled. Then the attenuator should not add noise or drift to stay “transparent” for other systems. To allow for other attenuators or amplifiers in the signal path to compensate the attenuation around its operating point, the nominal attenuation should be as low as possible. The attenuation resolution should be low enough to allow for fine control and not to add noticeable quantization noise. Also the setup time for a new value should be small enough to not be visible to the control system.

Other factors like the control/supply voltages and the operating temperature range are limited by the available hardware or governed by the mounting location.

All requirements are summarized in Table 4.2.

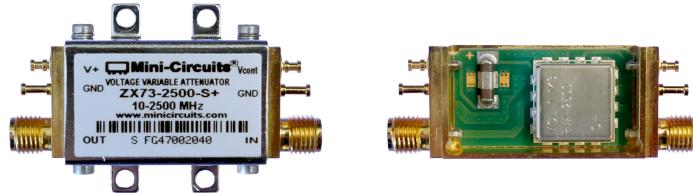
**Table 4.2:** Requirements for the controllable RF attenuator

Requirement	Value/Range
attenuation adjustment	$\pm 0.2$ dB
attenuation stability	$\pm 0.001$ dB
nominal attenuation at operating point	< 10 dB
attenuation resolution	0.001 dB
setup time	10 ms
operating temperature range	$(25.0 \pm 0.1)$ °C
supply voltage	3 V to 12 V
control voltage	3 V to 12 V

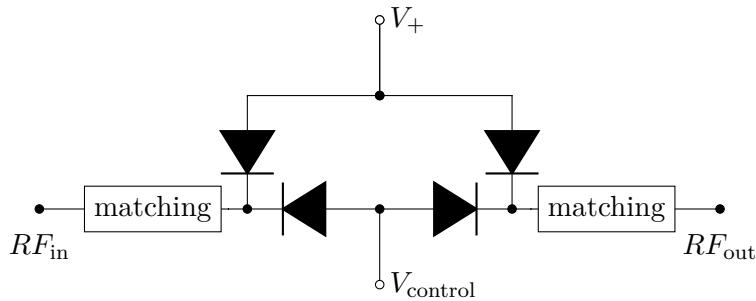
### 4.2.2 Evaluation of the ZX73-2500-S+ Controllable RF Attenuator

The *ZX73-2500-S+* is a voltage controllable RF attenuator with coaxial SMA connectors by the company Mini-Circuits. As there is no alternative model from Mini-Circuits and devices from other manufacturers are offered with similar specifications, only the *ZX73-2500-S+* is evaluated in detail in this section.

The *ZX73-2500-S+* attenuator consists of a brass housing/shielding containing the Mini-Circuits *RVA-2500+*, a variable SMD attenuator in the DV874 case form factor. The RF input and output are connected with female SMA screw connectors. The power supply and control voltage are connected with solder pins. In order to use shielded cables and a reliable connection, for all measurements SMA connectors are soldered to the supply and control pins. According to the equivalent circuit in the data sheet [[mini-circuitsZX732500VoltageVariable](#)], it can be assumed it is based on the common quad- $\pi$  pin diode design introduced in [[waughLowCostSurfaceMount1992](#)]. In this



(a) left: ZX73-2500-S+ with cover; right: cover removed showing RVA-2500+ and a buffer capacitor



(b) equivalent circuit from the manufacturer (redrawn from [mini-circuitsZX732500VoltageVariable])

**Figure 4.3:** The ZX73-2500-S+ controllable RF attenuator by Mini-Circuits

attenuator design, the resistors in a fixed  $\pi$ -configuration attenuator are switched out for RF PIN diodes with the appropriate biasing and matching resistors and capacitors, see Figure 4.4. At high frequencies, PIN diodes behave like resistors with the differential resistance

$$r_d = f(I_F) \quad (4.2)$$

being an inverse function of the forward bias current, which makes the attenuator adjustable.

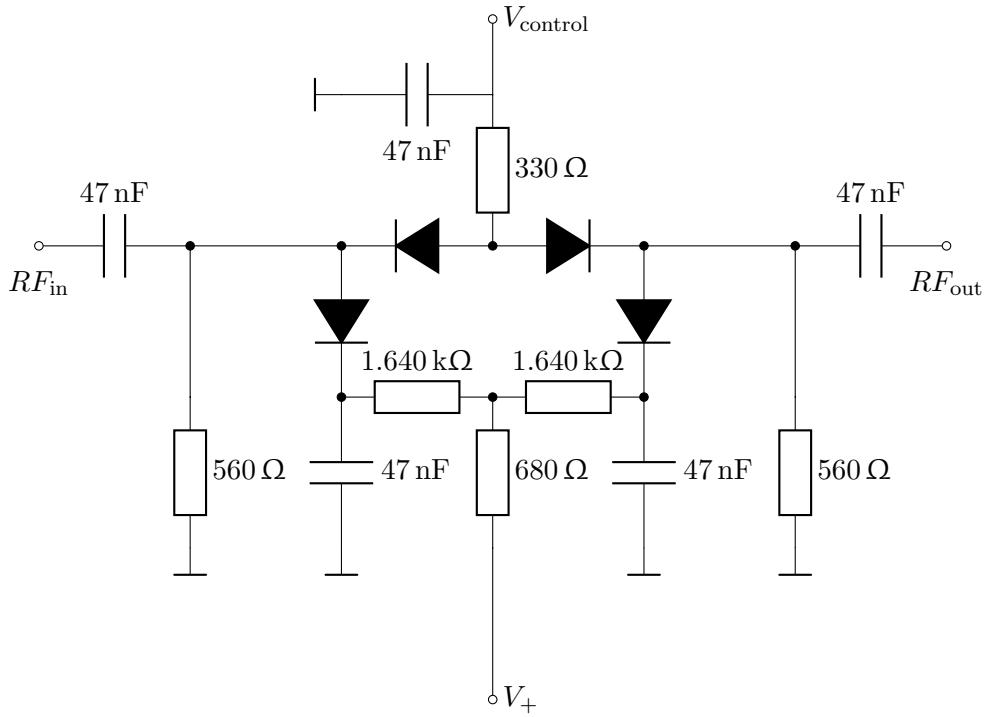
The attenuation versus frequency measurements from the manufacturer's data sheet are redone to both get a first impression of the device and verify it is generally operational and also as a sanity check for the used laboratory test equipment. Because the signal used by the FLUTE RF system is a 3 GHz single harmonic, the frequency measurement range is augmented over the maximum of 2.5 GHz in the data sheet to 4 GHz. Between the measurement of each network analyzer trace, the control voltage  $V_{control}$  of the attenuator is set to 0 V, 2 V, 4 V, 6 V or 12 V. The result is shown in Figure 4.5<sup>1</sup>. When comparing the measured plots to the plots in the data sheet, there are obvious discrepancies. For  $V_{control} = 0$  V the attenuator is very susceptible to noise on the control input, which could explain the differences for this curve. In the case of 2 V and 4 V, the almost constant offset scales with a similar logarithmic fashion as the attenuation does, which suggests device tolerances causing the deviations.

From this quick examination it is not possible to predict how the attenuator behaves for small changes in  $V_{control}$  and how changes in the environment, such as the body temperature or the supply voltage, cause unwanted variations in the attenuation.

For this reason, with different measurement setups, the ZX73-2500-S+ is examined in greater detail in the next sections.

To compare desired and spurious influences on the attenuation, the following model is used. The attenuation of the ZX73-2500-S+  $A$  depends on the control voltage  $V_{control}$  but also

<sup>1</sup>A thru calibration of the network analyzer reduces the influence of the cables and connectors on the measurement. The change in attenuation due to play in the connectors and slight bend changes in the cables exceeds the trace noise (0.004 dBrms) and causes an uncertainty of about 0.5 dB



**Figure 4.4:** Controllable attenuator design around the HP HSMP-3814 RF PIN diodes (redrawn and simplified from [waughLowCostSurfaceMount1992])

on the supply voltage  $V+$ , the case temperature  $\vartheta_{\text{case}}$  and the RF frequency  $f$ :

$$A = A(\underbrace{V_{\text{control}}, V_+, \vartheta_{\text{case}}, f}_{\vec{x}}) = A(\vec{x}) \quad (4.3)$$

Other influences are not modeled as they are difficult to control, such as the manufacturing tolerances between different devices, or assumed to be negligible, such as component degradation or humidity.

In the next sections, the influence of each component of the parameter vector  $\vec{x}$  on  $A$  is measured. As a coarse approximation, all influences are assumed to have linear effect. Then the total derivative of  $A$ ,  $\Delta A$  can be written as

$$\underbrace{A(\vec{x} - \vec{x}_o) - A(\vec{x}_o)}_{\Delta A} = \sum_{j=0}^3 \frac{\Delta A}{\Delta x_j} \Delta x_j \quad (4.4)$$

where the  $\Delta A / \Delta x_j$  approximate the partial derivatives  $dA/dx_j$ .

With Equation 4.4, the maximum error on the attenuation, that is the variation around the operating point with a fixed  $V_{\text{control}}$ , can be approximated with

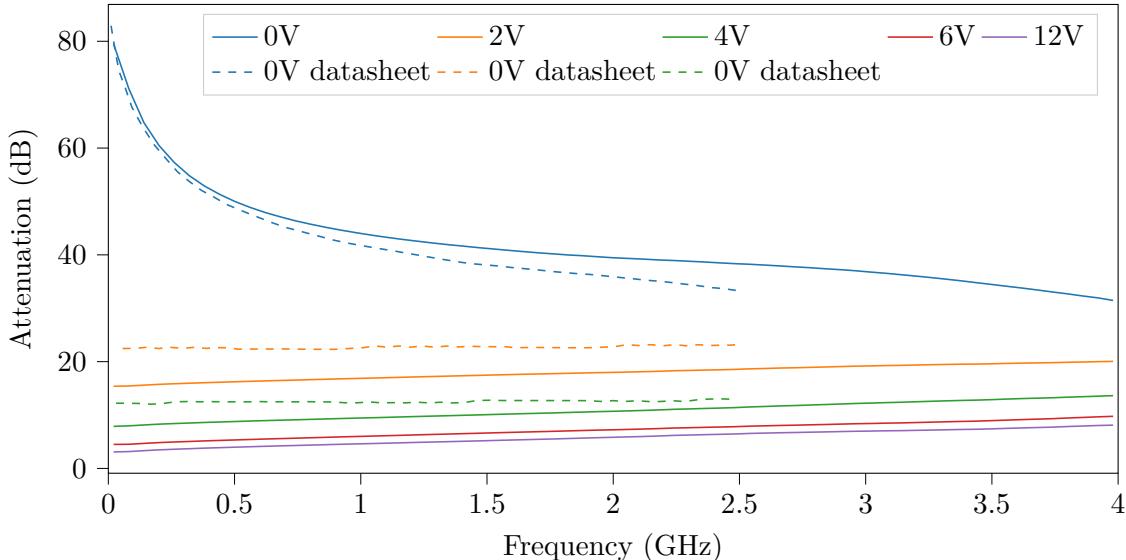
$$\Delta A_{\max} = \sum_{j=0}^3 \left| \frac{\Delta A}{\Delta x_j} \cdot \Delta x_j \right| = \sum_{j=0}^3 \left| \frac{\Delta A}{\Delta x_j} \right| \cdot |\Delta x_j|. \quad (4.5)$$

If not specified otherwise, the operating point

$$A(\vec{x}_o) =: A_o = A(V_{\text{control}}, V_+, \vartheta_{\text{case}}, f) \quad (4.6)$$

$$A_o = A(10 \text{ V}, 3 \text{ V}, 20^\circ \text{C}, 3 \text{ GHz}) \quad (4.7)$$

is used.



**Figure 4.5:** Attenuation vs. frequency over DC control voltage; measured with network analyzer (see Section B.5.1, parameters: #AVG: 16, IF-BW: 10 kHz); plotted in dashed lines are the measurements from the data sheet (see [mini-circuitsZX732500VoltageVariable])

#### 4.2.2.1 Common Measurement Setup

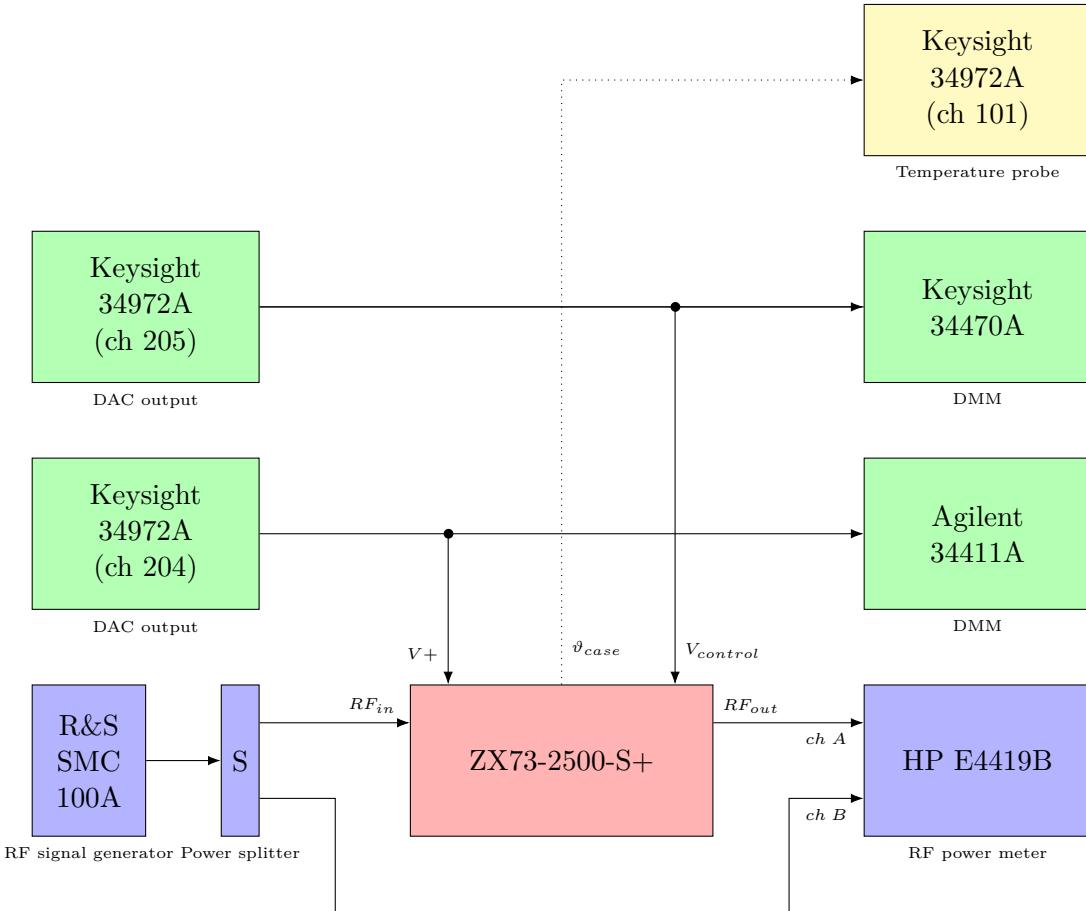
For the following measurements a common setup is used to ease the recording of data and the sequential control for parameter studies. The needed tasks are:

- Supply the attenuator with the supply voltage  $V_+$
- Supply the attenuator with the adjustable control voltage  $V_{\text{control}}$
- Supply the RF input power with variable frequency  $f$
- Measure the RF output power
- Record all data as time series to a computers storage device

To achieve this, the setup in Figure 4.6 is used. The supply and control voltage are generated by a Keysight 34972A with a 34907A module (see Section B.2.1). To get a more accurate measurement of the actual voltages, two digital multimeters (Models 34470A and 34411A, see Section B.1.2) are connected directly at the  $V_{\text{control}}$  and  $V_+$  pins. The RF signal is generated by a Rhode und Schwarz SMC 100A signal generator (see Section B.3.1). With the HP E4419B (see Section B.4.1) and its two inputs and a power splitter, it is possible to directly get the attenuation of the ZX73-2500-S+. The body temperature of the device is monitored with a PT100 temperature sensor connected to the 34972A.

Each of the lab devices used is compatible with VME eXtensions for Instrumentation protocol specification 11 (VXI-11), which is a widely adopted standard that is used to send American Standard Code for Information Interchange (ASCII) Standard Commands for Programmable Instruments (SCPI) commands over an ethernet network (called LAN eXtensions for Instrumentation (LXI)) or General Purpose Interface Bus (GPIB). [lxi2021] This enables remote and programmatically control of the devices over the network. With the library *python-vxi11*, it is now possible to write a custom script, that sets the measurement devices to known initial conditions, drives the inputs of the attenuator and records the generated data. Timing the output of new set-values and recording of data is done with the *Advanced Python Scheduler*. [apscheduler]

The whole setup of the hardware in Figure 4.6 and the Python software, a measurement



**Figure 4.6:** Measurement setup: DUT(red), RF generator/power splitter/meter(blue), DC sources/meters(green), temperature probe(yellow)

frequency of about 0.5 Hz to 1 Hz can be achieved, which is enough because most measurements taken are of a static nature and in the case of the temperature influence measurement, the thermal time constant is in the order of a few seconds. The limiting device is the HP E4419B which takes the longest to perform one measurement. Without it, measurement frequencies of over 4 Hz are possible.

All measurements are performed in the “RF lab” in building 348 (Karlsruhe Research Accelerator (KARA) hall on KIT Campus North) which is air conditioned to about  $(20 \pm 2)^\circ\text{C}$ .

#### 4.2.2.2 Relation between Control Voltage and Attenuation

In this section, the relation between the control voltage and the attenuation is examined. All other parameters are kept constant:

$$A(\vec{x}) := A(V_{\text{control}}) \quad (4.8)$$

The measured spectra in Figure 4.5 already suggest that there is a non linear relationship between the control voltage  $V_{\text{control}}$  and the attenuation  $A(V_{\text{control}})$  of the attenuator:

$$A(V_{\text{control}}) \neq \text{const.} \cdot V_{\text{control}} + A_0. \quad (4.9)$$

This also means the sensitivity

$$S(V_{\text{control}}) := \frac{\Delta A(V_{\text{control}})}{\Delta V_{\text{control}}} \quad (4.10)$$

is not a constant. In other words, for a desired relative change in attenuation, the needed adjustment in  $V_{\text{control}}$  depends on the chosen operating point  $A_o = A(V_{\text{control},o})$ .

With all other variables being at the operating point in Equation 4.6,  $A(V_{\text{control}})$  is measured by stepping  $V_{\text{control}}$  in 0.1 V increments up and down between 0 V and 12 V several times with each value held constant for 30 s.  $A(V_{\text{control}})$  is then calculated as the mean of all measurements  $A_j(V_{\text{control}})$  with the same  $V_{\text{control}}$  with

$$A(V_{\text{control}}) = \frac{1}{N} \sum_{j=0}^{N-1} A_j(V_{\text{control}}). \quad (4.11)$$

With the averaging done in Equation 4.11 and  $N = 120$ , the resulting mean standard deviation is  $\sigma_A = 0.00574$  dB. Figure 4.7 shows the resulting plot.

The plot shows the attenuation can be varied over more than 30 dB and the magnitude of the sensitivity being large for small control voltages (1 V to 3 V). Since the required change in attenuation of less than 1 dB is much smaller, it is only necessary to vary  $V_{\text{control}}$  around the operating point.

The optimal operating point for  $V_{\text{control}}$  is selected by considering the following three aspects. First, the absolute attenuation at the operating point should be as low as possible to not worsen the Signal-to-Noise Ratio (SNR) of the signal path. Second the control voltage has to fit in the possible output voltage range of the voltage source and still allow for adjustment towards both lower and higher voltages. In case of the Keysight 34972A/34907A the maximum possible output voltage is 12 V. Third, the sensitivity should be as low as feasible to make the attenuation less susceptible to noise on the  $V_{\text{control}}$  input.

For these reasons,  $V_{\text{control},o} = 10$  V (as already used in Equation 4.6) is chosen as the operating point, which allows  $V_{\text{control}}$  to be varied in 8 V to 12 V. At the operating point  $\vec{x}_o$  (with  $V_{\text{control},o} = 10$  V), the measured attenuation is

$$A(V_{\text{control}}) = 6.4859 \text{ dB} \quad (4.12)$$

Figure 4.8 shows the attenuation and sensitivity around the operating point.

The sensitivity at the operating point is determined with discrete forward differentiation as

$$S(V_{\text{control}}) = S(10 \text{ V}) = \frac{\Delta A}{\Delta V_{\text{control}}} = -0.151 \text{ dB V}^{-1}. \quad (4.13)$$

For further error calculations, the maximum  $\Delta A(V_{\text{control}})/\Delta V_{\text{control}}$  is needed. Figure 4.8 shows the maximum magnitude of  $\Delta A(V_{\text{control}})/\Delta V_{\text{control}}$  to be at the lower edge of the  $V_{\text{control}}$  range of  $S(8 \text{ V})$ :

$$\left[ \frac{\Delta A(V_{\text{control}})}{\Delta V_{\text{control}}} \right]_{\max} = -0.242 \text{ dB V}^{-1} \quad (4.14)$$

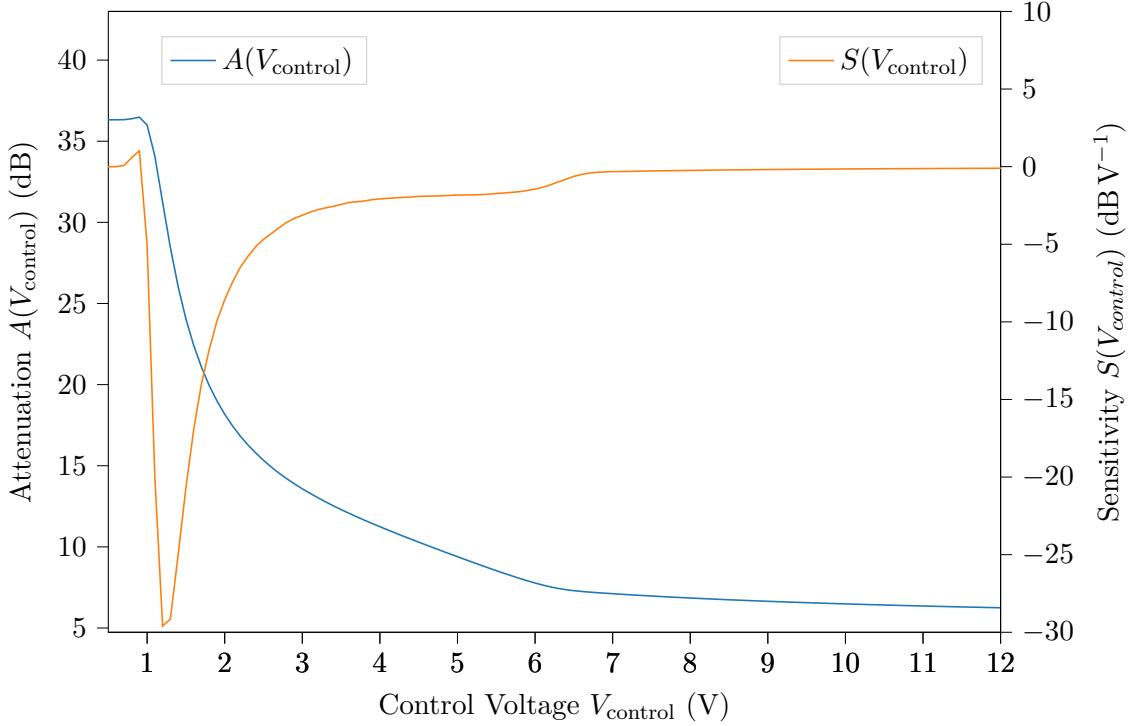
The minimum possible step size of the attenuation can be calculated using  $[\Delta A(V_{\text{control}})/\Delta V_{\text{control}}]_{\max}$  and the DAC output resolution of the Keysight 34907A ( $2^4 \text{ V}/2^{16}-1 = 366.22 \mu\text{V}$ ):

$$\delta A(V_{\text{control}}) = 0.242 \text{ dB V}^{-1} \cdot 366.22 \mu\text{V} = 0.00008862524 \text{ dB} \quad (4.15)$$

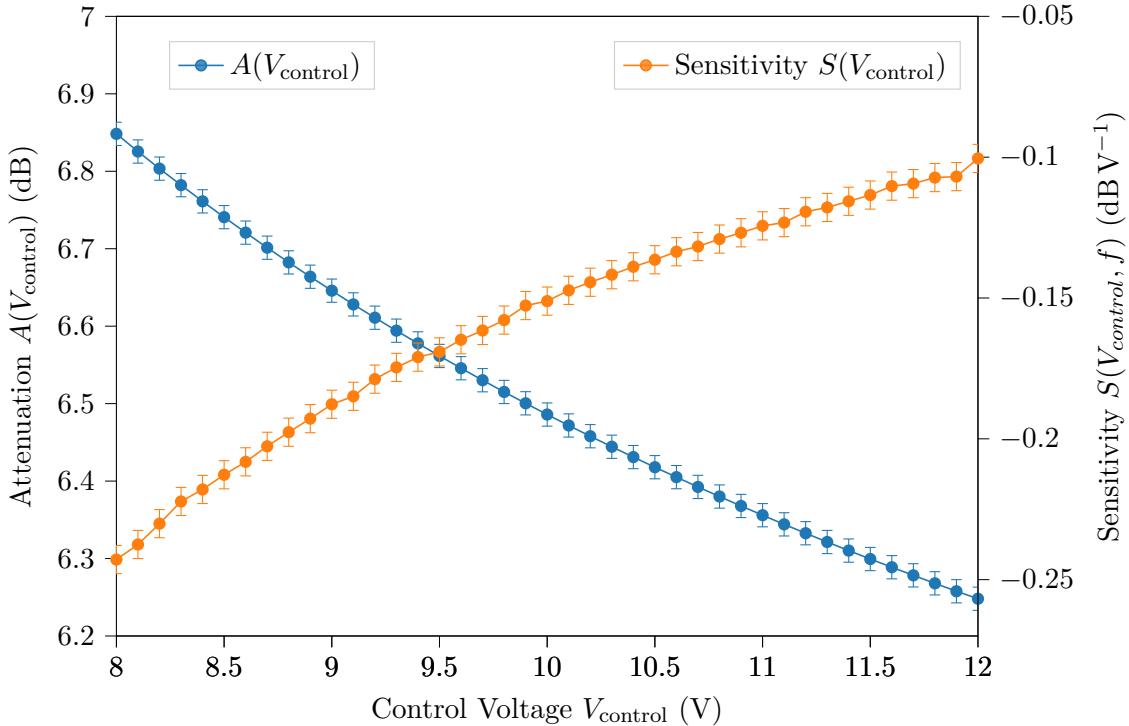
To assess the stability of  $V_{\text{control}}$ , delivered from the Keysight 34972A (see subsubsection B.2.1) Digital-to-Analog Converter (DAC), its stability over the course of one day is measured. For that the voltage is taken once every 2 seconds with a Keysight 34470A multimeter (see subsubsection B.1.2). The result is shown in Figure 4.9.

This measurement shows the stability of  $V_{\text{control}}$  to be

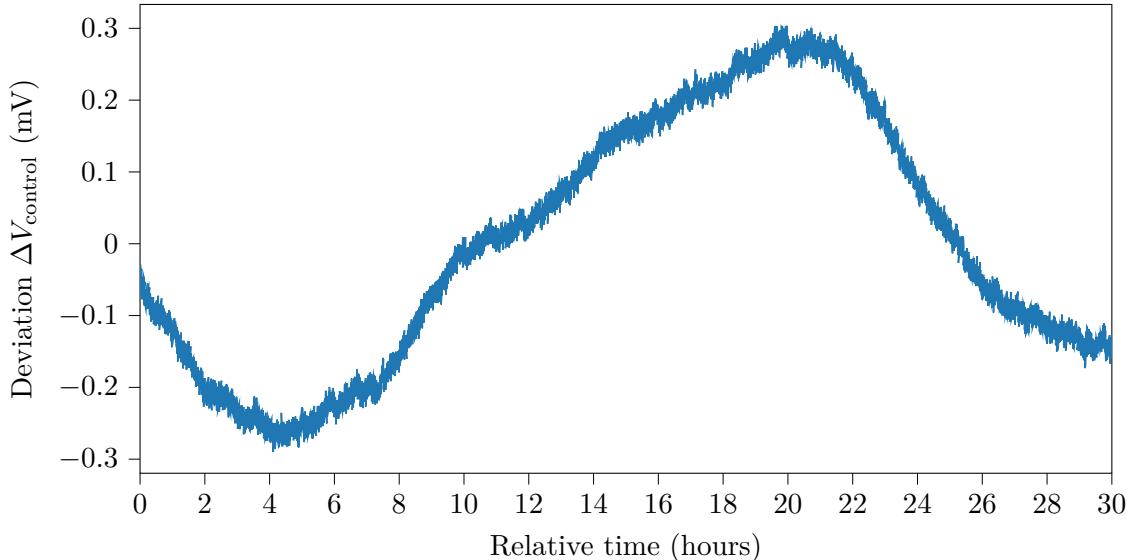
$$\sigma_{V_{\text{control}}} = 0.173 \text{ mV} \quad (4.16)$$



**Figure 4.7:** Measured attenuation  $A(V_{\text{control}}, f = 3 \text{ GHz})$  of the ZX73-2500-S+ as a function of the control voltage input  $V_{\text{control}}$  along with the sensitivity  $S(V_{\text{control}}, f = 3 \text{ GHz})$



**Figure 4.8:** Zoomed in version of Figure 4.7 shows the attenuation and sensitivity around the operating point  $V_{\text{control},o} = 10 \text{ V}$



**Figure 4.9:** Long term stability of  $V_{\text{control}}$  as delivered by the Keysight 34972A DAC (ch. 205); measured with Keysight 34470A;  
room temperature during measurement:  $\mu_\vartheta = 19.12^\circ\text{C}$ ,  $\sigma_\vartheta = 0.28^\circ\text{C}$

#### 4.2.2.3 Influence of Supply Voltage Noise on Attenuation

To get the required stability for the power supply voltage, the effect of the supply voltage  $V_+$  on the attenuation  $\Delta A/\Delta V_+(V_+)$  has to be examined first. To do that  $V_+$  is varied  $\pm 0.2\text{ V}$  around the nominal supply voltage at the operating point  $V_{+o} = 3\text{ V}$ , all other parameters are kept constant and the attenuation is measured. To make the measurement more robust against fluctuations of the room temperature and drift of the devices, the procedure of stepping through the voltages is repeated in a similar fashion as for the influence of  $V_{\text{control}}$  and the means for each set  $V_+$  are computed. The result is shown in Figure 4.10.

The plot shows  $A(V_+)$  to be of linear nature over the measured range. Therefore using a linear regression of the measured data points, the influence on the attenuation can be estimated to

$$\frac{\Delta A(V_+)}{\Delta V_+}(V_+) \approx \frac{\Delta A(V_+)}{V_+} = \left[ \frac{\Delta A(V_+)}{V_+} \right]_{\max} = 0.003\,559\,2 \text{ dB V}^{-1}. \quad (4.17)$$

Next the stability of the supply voltage is measured. for that the stability over the course of one day is measured. The voltage is taken once every two seconds with a Keysight 34470A multimeter (see subsubsection B.1.2). The result is shown in Figure 4.11.

Comparing Figure 4.11 with Figure 4.9 suggests a relation between the deviations in  $V_{\text{control}}$  and  $V_+$ . Since they are both generated by the same Keysight 34907A module, this is plausible. The slightly changing room temperature is assumed to be the common cause. By calculating the correlation coefficients between  $\Delta V_+$  and  $\Delta V_{\text{control}}$  and also between  $\Delta V_+$  and  $\Delta \vartheta_{\text{ambient}}$ , this can be verified:

$$\text{Corr}(\Delta V_+, \Delta V_{\text{control}}) = 0.992\,04 \quad (4.18)$$

$$\text{Corr}(\Delta V_+, \Delta \vartheta_{\text{ambient}}) = -0.922\,42 \quad (4.19)$$

$$(4.20)$$

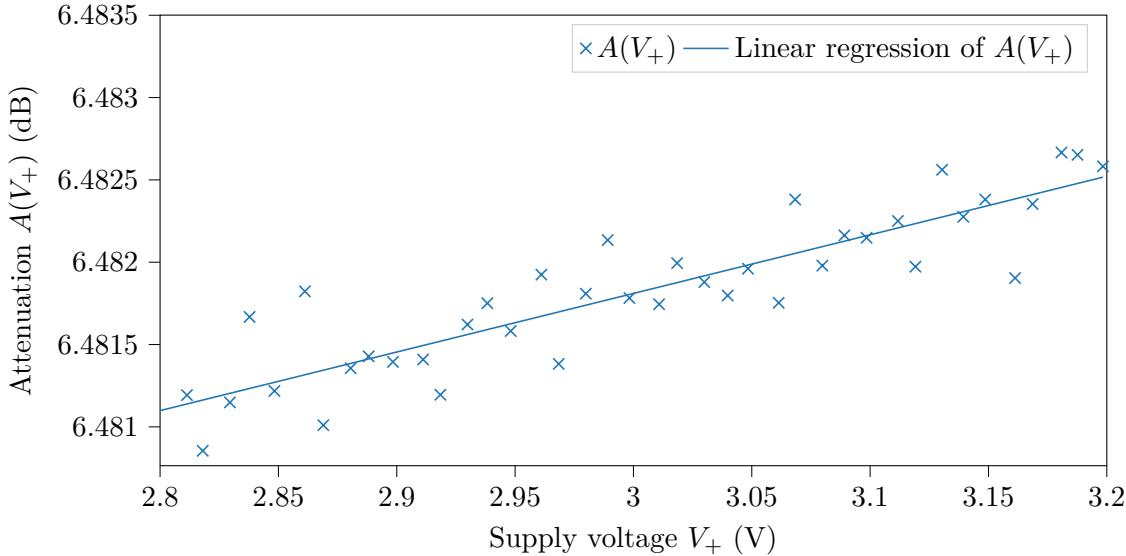
The long term measurement yields a standard deviation of

$$\sigma_{V_+, \text{longterm}} = 0.154 \text{ mV}, \quad (4.21)$$

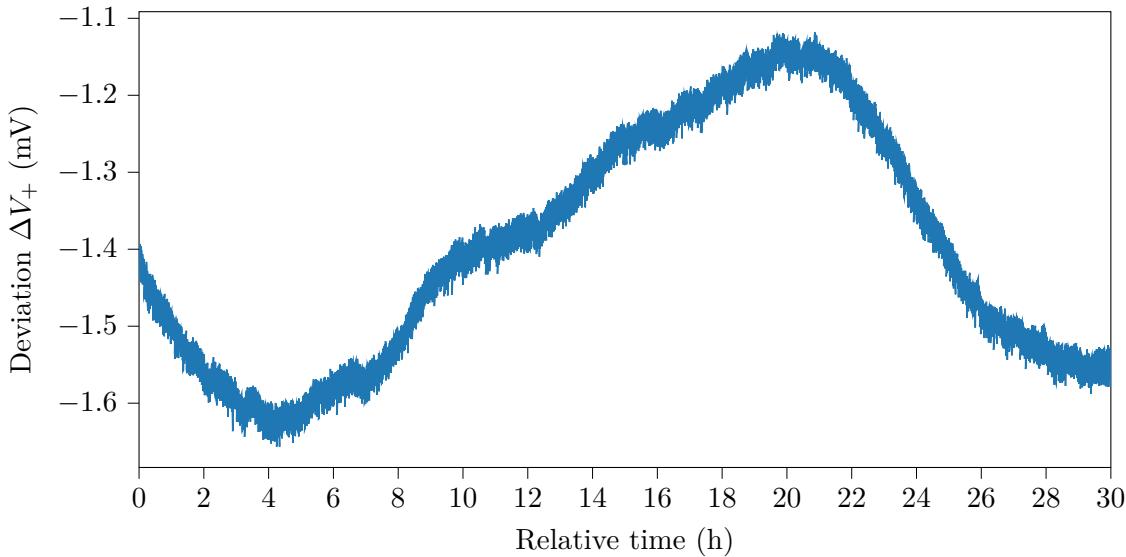
which is also used as the worst case stability

$$\sigma_{V_+} = 0.154 \text{ mV} \quad (4.22)$$

There is also a constant offset of  $\mu_{V_+, \text{longterm}} = -1.35 \text{ mV}$ , but it is disregarded since it can easily be compensated by slightly increasing the supply voltage.



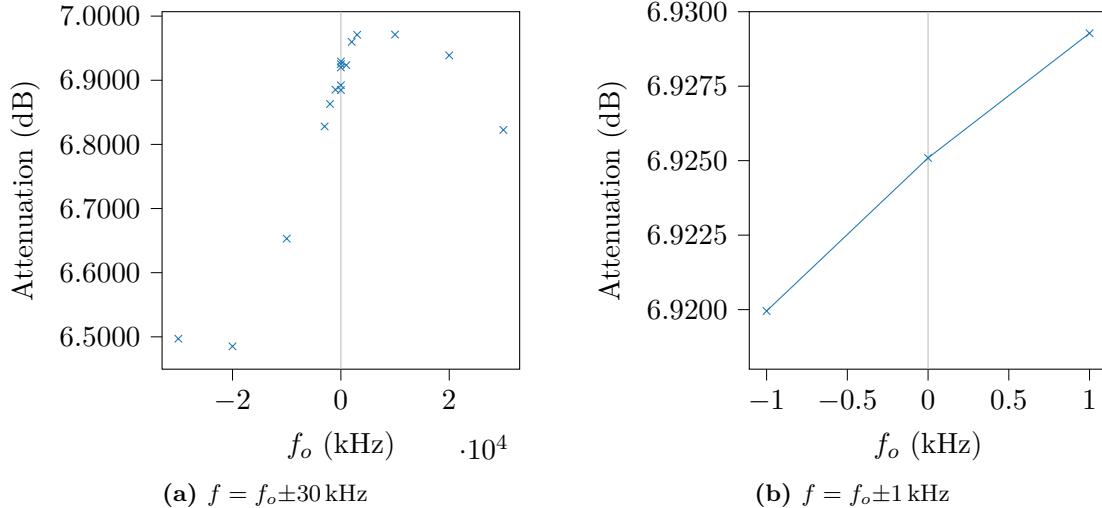
**Figure 4.10:** Influence of the supply voltage  $V_+$  on the attenuation



**Figure 4.11:** Long term stability of  $V_+$  as delivered by the Keysight 34972A DAC (ch. 204); measured with Keysight 34470A;  
room temperature during measurement:  $\mu_\vartheta = 19.12^\circ\text{C}$ ,  $\sigma_\vartheta = 0.28^\circ\text{C}$

#### 4.2.2.4 Influence of RF Frequency on Attenuation

In this section the influence of a varying RF frequency on attenuation  $\Delta A/\Delta f(f)$  is examined. For that the set frequency of the R&S SMC100 signal generator is varied while the attenuation is measured. The result is shown in Figure 4.12.



**Figure 4.12:** Influence of an offset frequency  $f - 3 \text{ GHz}$  on the attenuation

Phase noise measurements of the FLUTE main oscillator<sup>2</sup> yield the phase noises

$$\mathcal{L}(f_o = 10 \text{ Hz}) = -78.63 \text{ dBc} \quad (4.23)$$

$$\mathcal{L}(f_o = 1 \text{ kHz}) = -116.26 \text{ dBc} \quad (4.24)$$

$$\mathcal{L}(f_o = 10 \text{ kHz}) = -139.29 \text{ dBc} \quad (4.25)$$

$$\mathcal{L}(f_o = 1 \text{ MHz}) = -142.29 \text{ dBc}, \quad (4.26)$$

with  $f_o$  being the offset frequency from the carrier and the unit dBc measuring the offset power from the carrier.

Since this shows the oscillator to be much more stable than  $f_o = 10 \text{ kHz}$ , is it is sufficient to consider only Figure 4.12 (b). In the range of  $\pm 1 \text{ kHz}$  around the operating point frequency of  $3 \text{ GHz}$ ,  $\Delta A/\Delta f(f)$  is almost constant. From the data points, the value can be calculated to

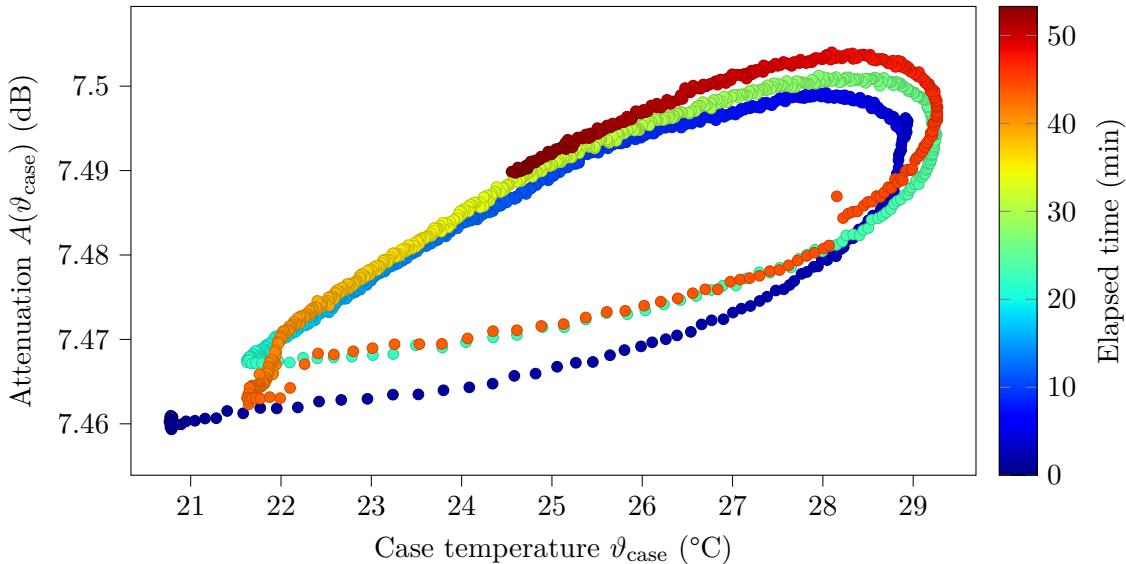
$$\frac{\Delta A(f)}{\Delta f}(f) \approx \frac{\Delta A(f)}{\Delta f} = \left[ \frac{\Delta A(f)}{\Delta f} \right]_{\max} = 0.000\,004\,661\,9 \text{ dB Hz}^{-1}. \quad (4.27)$$

#### 4.2.2.5 Influence of Case Temperature Variations on Attenuation

To get insight into the importance of a stable case temperature, its influence on the attenuation  $\Delta A/\Delta \vartheta_{\text{case}}(\vartheta_{\text{case}})$  and the temperature stability both in the “RF lab” and the final mounting position inside the FLUTE LLRF cabinet are measured.

To measure  $\Delta A/\Delta \vartheta_{\text{case}}(\vartheta_{\text{case}})$ , the following experimental setup is used.

The bottom of the attenuator is fixed to a rectangular iron profile with zip ties. Then the iron profile is heated with the tip of a soldering iron (set to  $150^\circ\text{C}$ ) for 1 min and then allowed to cool for 20 min. This cycle is repeated three times and the device temperature and the attenuation are measured once every 2 s. Due to heat flowing from the soldering



**Figure 4.13:** Raw measurement result of the influence of the case temperature  $\vartheta_{\text{case}}$  on the attenuation; color coded is the relative elapsed time of the measurement

iron to the iron profile and through device itself, a strong hysteresis between the heating and cooling cycle can be observed in Figure 4.13.

To get  $\Delta A / \Delta \vartheta_{\text{case}}(\vartheta_{\text{case}})$  from the plot in Figure 4.13, an approximately linear relationship  $A(\vartheta_{\text{case}})$  is assumed. To get the slope, the  $\vartheta_{\text{case}}$  component of the measured data points ( $\vartheta_{\text{case}} | A(\vartheta_{\text{case}})$ ) are rounded to one decimal place. Then, data points with the same  $A(\vartheta_{\text{case}})$  are averaged together and a linear regression is applied to the result. See Figure 4.14 for the resulting plot. The linear regression estimator yields

$$\frac{\Delta A}{\Delta \vartheta_{\text{case}}}(\vartheta_{\text{case}}) \approx \frac{\Delta A}{\Delta \vartheta_{\text{case}}} = \left[ \frac{\Delta A}{\Delta \vartheta_{\text{case}}} \right]_{\max} = 0.00432449 \text{ dB } ^\circ\text{C}^{-1}. \quad (4.28)$$

Next, the temperature stability in both the “RF lab” and in the FLUTE LLRF cabinet is measured. For that in each environment the temperature is recorded over night with a thermo element connected to the Keysight 34907A inside the Keysight 34972A every two seconds.

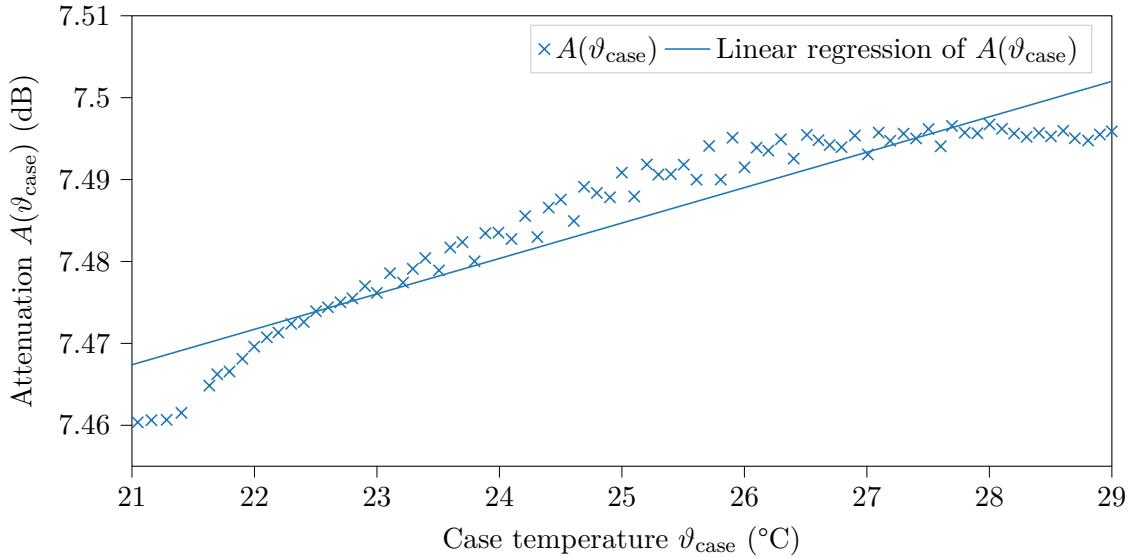
From the data in Figure 4.15, the corresponding stabilities are calculated to be

$$\sigma_{\vartheta_{\text{ambient, RF lab}}} = 0.1067 \text{ } ^\circ\text{C} \quad (4.29)$$

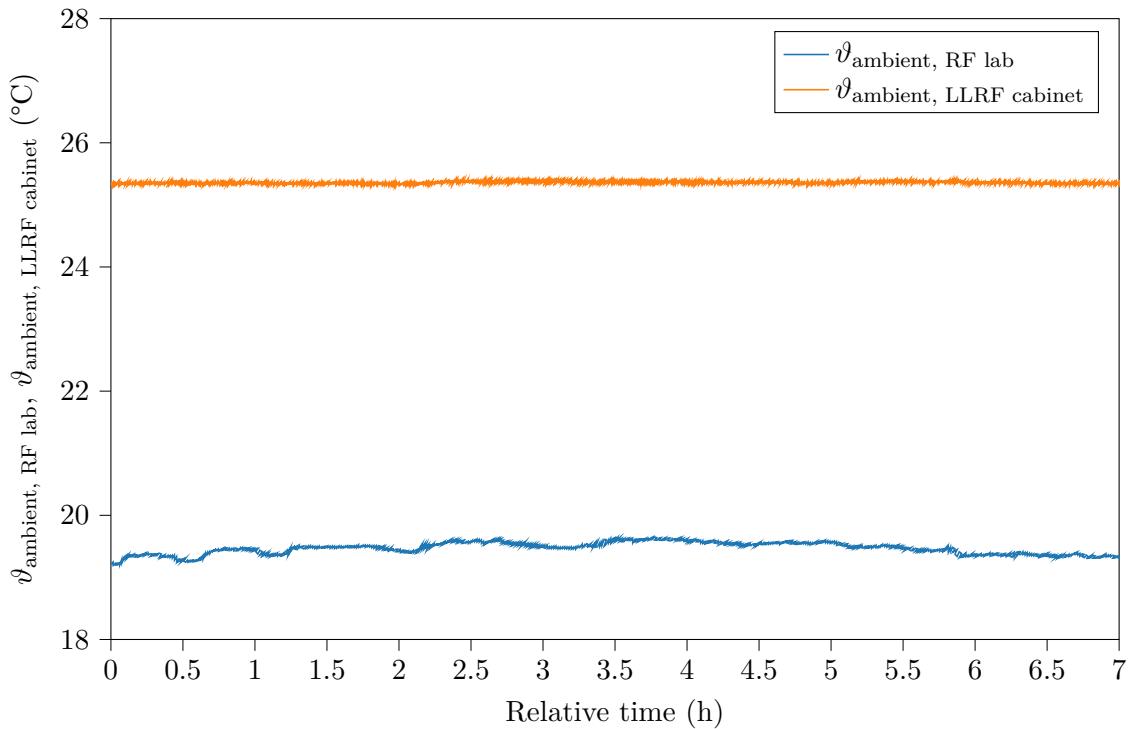
$$\sigma_{\vartheta_{\text{ambient, LLRF cabinet}}} = 0.0288 \text{ } ^\circ\text{C}. \quad (4.30)$$

This assumes the case of the attenuator and the ambient around it being in thermal equilibrium. With the thick brass case and its only weakly mechanically mounting to the metal support, the attenuator posses a significant thermal time constant  $\tau_{th} = R_{th}C_{th}$ . Therefore, thermally the attenuator is a lowpass filter and cannot follow fast changes in the ambient temperature. This means only slow changes of the ambient temperature have an influence on  $A$ , so short changes, like opening a door, have only a minor effect.

<sup>2</sup>Using the phase noise analyzer HA7062C by Holzworth, see Section B.6.1



**Figure 4.14:** Influence of the case temperature  $\vartheta_{\text{case}}$  on the attenuation



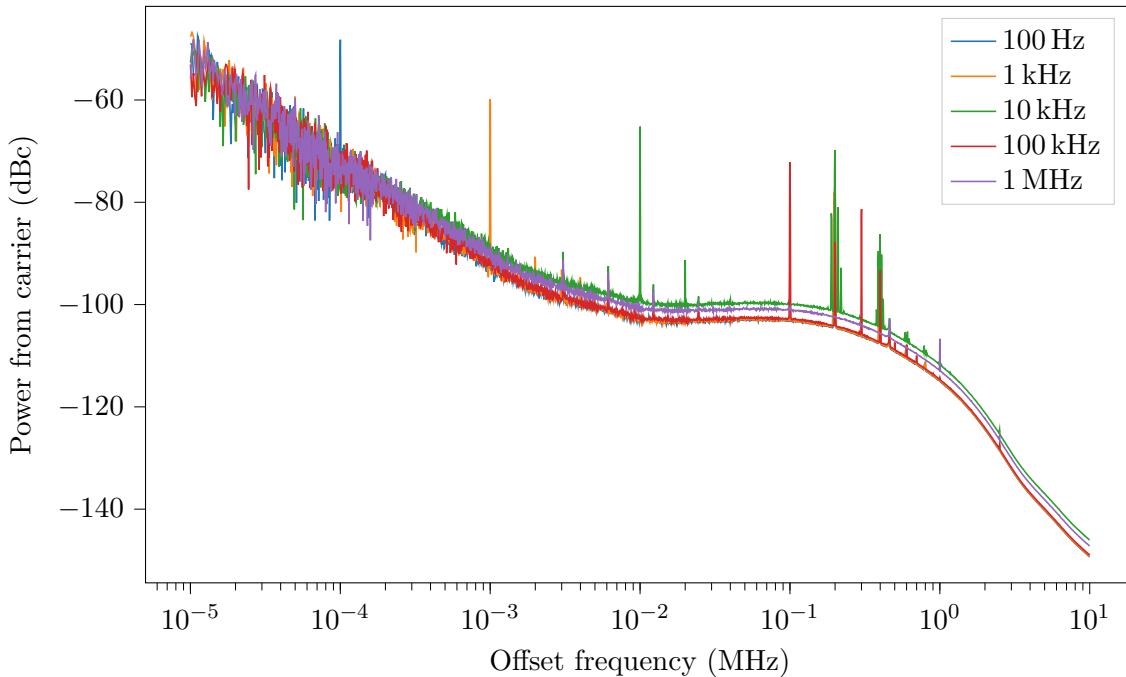
**Figure 4.15:** Comparison between the temperature stability in the “RF lab” and inside the FLUTE LLRF cabinet

#### 4.2.2.6 $V_{\{ \} < control}$ Frequency response

Using a non inverting adder with a *TS912IN* operational amplifier, a sine wave with constant offset is made, which is then used to drive the control voltage  $V_{control}$ . The result for different sine wave frequencies is shown in Figure 4.16.

This result is to be interpreted purely qualitatively, as neither measurement device and software is designed, nor is the circuit used optimal. But the traces in Figure 4.16 verify that the ZX73-2500-S+ attenuator is able to follow changes of  $V_{control}$  at least up to a few ten kHz. If the attenuator should be set to a new value, this is equivalent to applying a step at the attenuators  $V_{control}$  input which, according to the Fourier transform of a  $\text{rect}(t)$  pulse, a  $\text{sinc}(f)$ , contains high frequencies.

This is several orders of magnitude larger than the control system can input, compute and output new values to the attenuator.



**Figure 4.16:** Spectrum (measured with Holzworth HA7062A (see Section B.6.1)) showing the effect of modulating  $V_{control}$  with different frequencies (Modulation amplitude: 1 V)

#### 4.2.2.7 Combined Maximum Error of the Attenuation and Conclusion

Using Equation 4.5 and setting  $\Delta x_j = \sigma_{x_j}$  and  $\Delta A/\Delta x_j = [\Delta A/\Delta x_j]_{\max}$ , the upper bound on the deviation of the attenuation, that is the error  $\Delta A$  in the worst case, can now be calculated:

$$\Delta A_{\max} = \sum_{j=0}^3 \left| \frac{\Delta A}{\Delta x_j} \cdot \Delta x_j \right| \quad (4.31)$$

$$= \left| \frac{\Delta A}{\Delta V_{\text{control}}} \cdot \Delta V_{\text{control}} \right| + \left| \frac{\Delta A}{\Delta V_+} \cdot \Delta V_+ \right| + \left| \frac{\Delta A}{\Delta f} \cdot \Delta f \right| + \left| \frac{\Delta A}{\Delta \vartheta_{\text{case}}} \cdot \Delta \vartheta_{\text{case}} \right| \quad (4.32)$$

$$= \underbrace{0.242 \text{ dB V}^{-1} \cdot 0.173 \text{ mV}}_{41.9 \mu\text{dB}} \quad (4.33)$$

$$+ \underbrace{0.003 559 2 \text{ dB V}^{-1} \cdot 0.154 \text{ mV}}_{548 \text{ ndB}} \quad (4.34)$$

$$+ \underbrace{0.000 004 66 \text{ dB Hz}^{-1} \cdot 0 \text{ Hz}}_{0 \text{ dB}} \quad (4.35)$$

$$+ \underbrace{0.004 325 \text{ dB } ^\circ\text{C}^{-1} \cdot 0.11 \text{ } ^\circ\text{C}}_{475.75 \mu\text{dB}} \quad (4.36)$$

$$= 518 \mu\text{dB} = 0.000 518 \text{ dB} \quad (4.37)$$

This shows that even in the worst case, the ZX73-2500-S+ attenuator is stable within the requirement.

The other requirements from Table 4.2 are also fulfilled, see Table 4.3.

**Table 4.3:** ZX73-2500-S+ controllable RF attenuator evaluation test results

Requirement	Value/Range set	Value/Range actual	Verdict
attenuation adjustment	$\geq \pm 0.2 \text{ dB}$	$\pm 0.3 \text{ dB}$	pass
attenuation stability	$\leq \pm 0.001 \text{ dB}$	0.000 518 dB	pass
nominal attenuation at operating point	$\leq 10 \text{ dB}$	6.6 dB	pass
attenuation resolution	$\leq 0.001 \text{ dB}$	0.000 089 dB	pass
setup time	$\leq 10 \text{ ms}$	1 ms	pass
operating temperature range	$(25.0 \pm 0.1) \text{ } ^\circ\text{C}$	$(25.0 \pm 0.1) \text{ } ^\circ\text{C}$	pass
supply voltage	3 V to 12 V	3 V	pass
control voltage	3 V to 12 V	8 V to 12 V	pass

#### 4.2.3 Test of the Attenuator with FLUTE

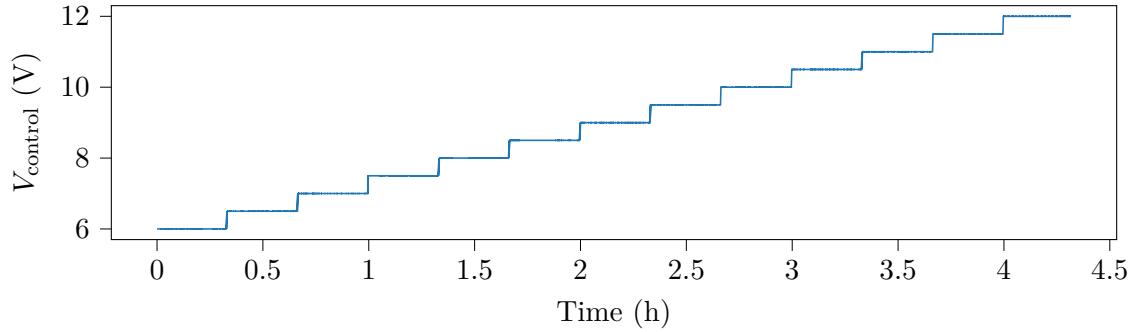
In this section the attenuator is mounted at FLUTE and its function is verified against the data gathered in the lab in earlier sections.

The attenuator is installed at its final mounting location inside the LLRF cabinet in the FLUTE bunker basement and is connected between the output of the vector modulator (after the oscillator, not shown in Figure 3.1) and the input of the pre-amplifier. With the controllable attenuator at its operating point, the signal path now contains an attenuation of about 6 dB at all times.

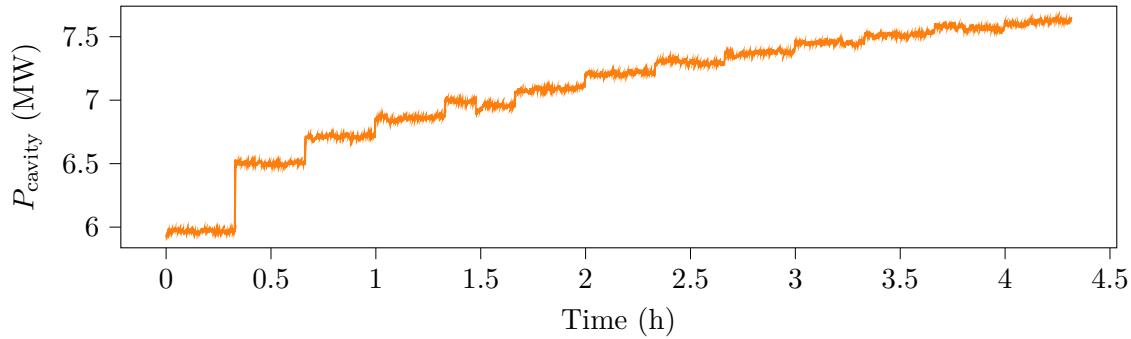
After all components of FLUTE are warmed up to operational temperatures, the attenuation versus control voltage curve  $A(V_{\text{control}})$  is measured again (compare the lab measurement Figure 4.7).

To do so, the control voltage  $V_{\text{control}}$  is varied in 0.5 V steps (see Figure 4.17), with each step kept for 20 min. Synchronous to that, the RF power of the cavity is recorded (see

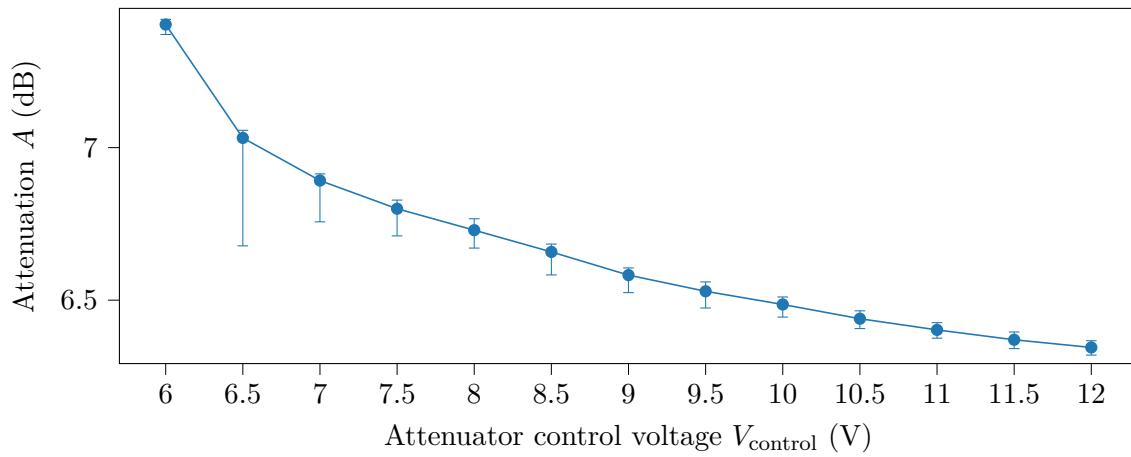
Figure 4.18), from which the attenuation  $A$  can be calculated. Computing the averaging over each step yields  $A(V_{\text{control}})$ . It is depicted in Figure 4.19 together with its uncertainty.



**Figure 4.17:** Time signal  $V_{\text{control}}$  used to calculate Figure 4.19



**Figure 4.18:** Time signal  $A(V_{\text{control}})$  used to calculate Figure 4.19



**Figure 4.19:** Measured  $A(V_{\text{control}})$  with FLUTE; error bars show the span between minimum and maximum values in each  $V_{\text{control}}$  step



# 5. Controller Design and Evaluation

In this chapter, a control system is designed and evaluated to stabilize FLUTEs RF system. Referring back to the block diagram of a generic control system in Figure 2.1, there are three building blocks of a control system to be determined. First the plant transfer function, which describes the system that is to be controlled. Second, based on the plant, an appropriate controller type is chosen and its parameters are calculated. Third, a measurement filter is used to improve the quality of the feedback signal path. As the choice of the measurement filter influences the controller design, its design is treated before the controller.

## 5.1 Plant Identification

### 5.1.1 Principle

Before choosing an appropriate controller, some insight of the system response has to be obtained. Therefore in this section the plant's transfer function is estimated. In the context of this chapter "plant" refers to everything from the attenuation set at the controllable attenuator to the system output, e.g. the cavity RF power.

In the time domain, a LTI system is described by its impulse response  $h(t)$ , that is the reaction of the plant to an impulse at the input. Using this definition directly, the plant's impulse response  $p(t)$  could be measured by applying a short peak in the attenuation setting on the attenuator. The effect on the output is not easy to measure and a single measurement of this kind is very susceptible to noise. Therefore it is more common to measure the step response, which is the output of a system, when a step function is applied to its input. As the step function is the time integral of the impulse function, the step response can be converted to the impulse response by differentiation in time. [Wang2000]

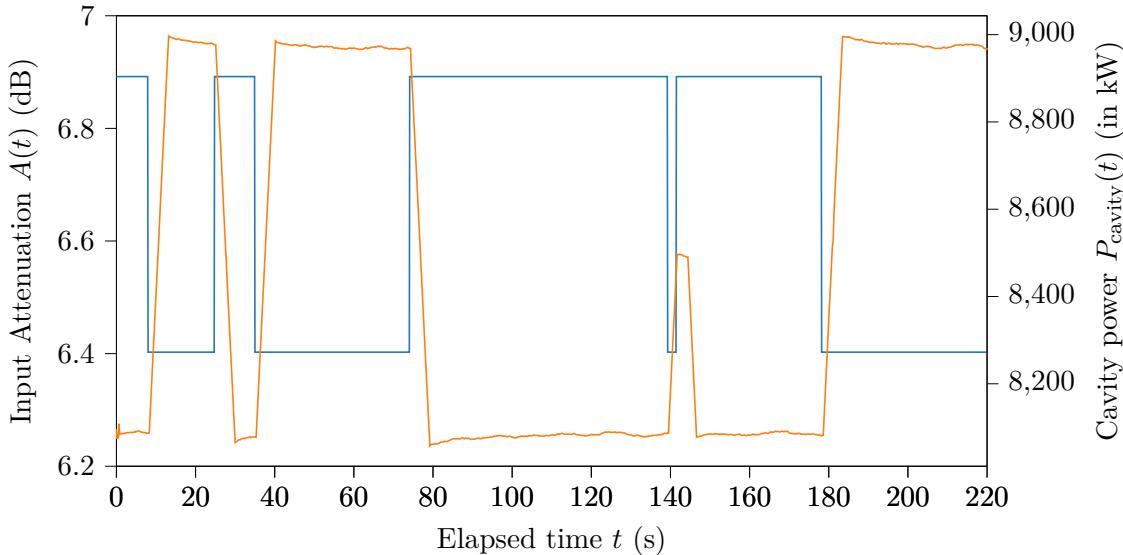
Instead of measuring a single step response, often several step responses are measured and their average is computed to reduce the variance of the estimation. When measuring a step response, the minimum needed measuring time depends on the systems time constants, but they are often not known beforehand.

That is why when there is no prior knowledge of the system, the identification is sometimes done with a Pseudo Random Binary Sequence (PRBS) to excite the system with step functions of different lengths. The PRBS is chosen in a way that some of the steps will probably last longer than a few dominant time constants of the system.

To get the transfer function  $P(s) = \mathcal{L}\{p(t)\}$  of the plant from its step response(s), several methods are common, including correlation based and frequency response based algorithms.

### 5.1.2 Identifying the Plant Attenuator+RF

The input PRBS signal is generated with the Python script in Listing 5.1. Based on the value of a pseudo random number generator, the sequence toggles the attenuator between



**Figure 5.1:** Section of the input sequence (blue, Attenuation) and the system response (orange, cavity RF power); Note the inverse relation: A higher attenuation  $A$  causes a lower cavity power  $P_{\text{cavity}}$

$V_{\text{control}} = 7 \text{ V}$  and  $V_{\text{control}} = 11 \text{ V}$ . Using Figure 4.7, this equals a span in attenuation of  $6.892 \text{ dB}$  to  $6.4026 \text{ dB}$ . With the parameter `toggleP`, the average length of one constant voltage level can be controlled.

**Listing 5.1:** Function to get a pseudo random binary sequence

---

```

1 def randomBinarySequence(N,toggleP):
2     u=[False]*N
3     for i in range(1,len(u)):
4         if(np.random.binomial(1, toggleP, 1)[0]):
5             u[i]=not u[i-1]
6         else:
7             u[i]=u[i-1]
8     return list(map(lambda x: 7 if x==False else 11,u))

```

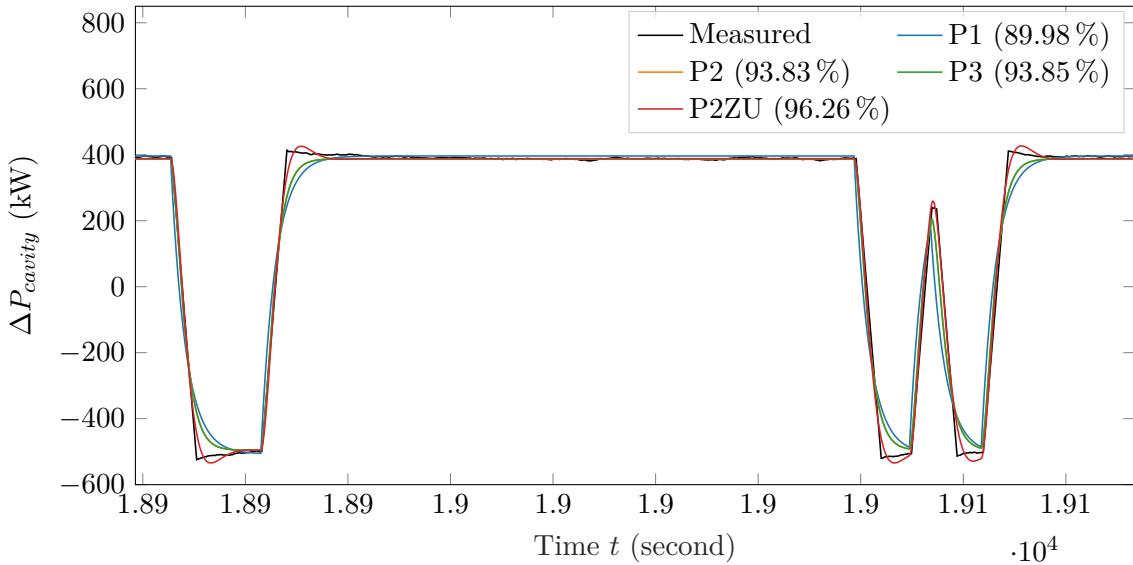
---

In a test run over six hours (after all FLUTE subsystems had stabilized), the attenuator was driven with such a PRBS. The result is shown in Figure 5.1.

The time signals  $A(t)$  and  $P_{\text{cavity}}(t)$  are then split into an *estimation* data set (about 80 % of the samples) and a *validation* data set (the remaining  $\approx 20 \%$ ). This is done so the bulk of the available information is used to estimate the model, but there is data left that the model hasn't seen before. This smaller portion is used to validate the model's performance, hence it is called the validation data set.

The two data sets are then loaded into the `{ }<MATLAB System Identification Toolbox`. With the toolboxes pre-processing tools, first the means of both the input and output are removed, which is required by the estimators used. Then, using the “process model” estimator, linear, continuous time transfer functions with different numbers of zeros and poles are estimated (See screenshot of the Graphical User Interface (GUI) in Figure A.1). After that, to check the accuracy of the estimations, the System Identification Toolbox is used to simulate the output of the estimated systems (see Figure 5.2). For that the aforementioned validation data set is used. The measured output is compared with the outputs predicted by the models.

Using the Matlab function `zpk()` the models are converted into the zero-pole-gain representation. In Table 5.1 the estimated models are listed with their zeros, poles, gains and



**Figure 5.2:** Validation of the estimated process models for the plant; the legend also shows the model fits in percent

**Table 5.1:** Process models of the plant as estimated by the Matlab System Identification Toolboxes process model estimator

Model	Zeros	Poles	Gain	Model fit
P1		$(s + 0.3505)$	-646.94	89.98 %
P2		$(s + 0.6875)(s + 0.7039)$	-873.18	93.83 %
P3		$(s + 1 \times 10^6)(s + 0.7376)(s + 0.669)$	$-8.9019 \times 10^8$	93.85 %
P2ZU	$(s - 2846)$	$(s^2 + 0.8014s + 0.3195)$	0.20296	96.26 %

the model fit, as it is computed by the System Identification Toolbox. P1, P2 and P3 are models with one, two and three poles and a gain as free parameters. P2ZU consists of a complex pole pair, a zero and a gain.

Figure 5.2 and Table 5.1 show the P2ZU model to have the best fit. Therefore it is accepted as the plants transfer function. Using the Matlab function `tf()`, its time continuous transfer function can be stated as

$$P(s) = \frac{0.6352s - 1808}{3.13s^2 + 2.508s + 1}. \quad (5.1)$$

Using the Matlab function `c2d()`<sup>1</sup> and the sample time  $T_s = 0.2$  s,  $G(s)$  can be converted to the time discrete transfer function

$$P[z] = \frac{-10.91z^{-1} - 10.41z^{-2}}{1 - 1.84z^{-1} + 0.8519z^{-2}}. \quad (5.2)$$

The high model fit percentage of 96.26 % justifies the choice of a linear model with few parameters and further estimation attempts using other (non-) linear models are not needed. Considering the accuracy of the model, it is important to note this estimation is only valid at the time the measurements are taken. With FLUTE being a large experimental setup still under commission, it is always possible that small changes to certain (sub-) systems can lead to minor or major influences to others. For the time being, the estimation is

<sup>1</sup>Without specifying a different method, `c2d()` discretizes the continuous-time model zero-order hold on the input.

redone a day later and for both measurements days, different parts of the about six hour measurements each are used as the estimation and validation data sets. Doing so shows no significant change in the estimated coefficients and the resulting model fits, indicating the estimated models are at least plausible.

To account for errors in the estimated model (and possible other errors), when designing the controller, sufficient gain and phase margins are set to ensure stable operation.

## 5.2 Measurement Filter

Like all measurements of physical quantities, the measuring of the system output of the control system is subjected to noise. In addition to these disturbances of thermal or electrical origins, also high frequency variations of the system output have detrimental effects on the control systems performance. For example, the magnitudes of the bunch-by-bunch changes of the measured cavity power are often in the same order as the long-term drifts. Trying to correct for them, instead of the long term drifts, often leads to overcompensating and can even make the system unstable.

To remove the high frequency components, a low pass filter is used as the measurement block  $H(s)$ .

In pre-tests, the incoming signal was simply filtered with a moving average filter. Commonly, the moving average is defined as the mean of a signal  $x$  inside a window of length  $L$ , centered around the current time or sample index  $n$ , that is shifted along the signal. This smooths out small variations, thus the moving average acts as a low pass filter. This *non-causal* version of the moving average can only be used with already measured data as to compute the moving average at  $n$ , future values at  $n + i$  are needed:

$$\text{MA}_{x,\text{non-causal},L}[n] = \frac{1}{L} \sum_{i=n-\frac{L-1}{2}}^{n+\frac{L-1}{2}} x[i] \quad (5.3)$$

When filtering real-time data, future values are not available and a shifted, *causal* version, of the moving average

$$\text{MA}_{x,\text{causal},L}[n] = \frac{1}{L} \sum_{i=n-(L-1)}^n x[i] \quad (5.4)$$

is used.

In case of the cavity RF power, experiments show a window length of about  $L = 100$  or more is necessary to sufficiently smooth the measured power signal. When comparing the original signal with the filtered one, it is apparent that in addition to the desired smoothing effect, the filtered signal also is delayed in time, with the delay being dependent on the window size. To quantify the delay, the alternative definition of the moving average as a digital Finite Impulse Response (FIR) filter is used. One possibility to describe a FIR filter is by giving its impulse response, i.e. the output signal when the input of the filter is an impulse with unity height. In case of the moving average filter, the coefficient sequence of the corresponding FIR filter has the length  $N = L$  and is equal to the the impulse response  $h[n]$ :

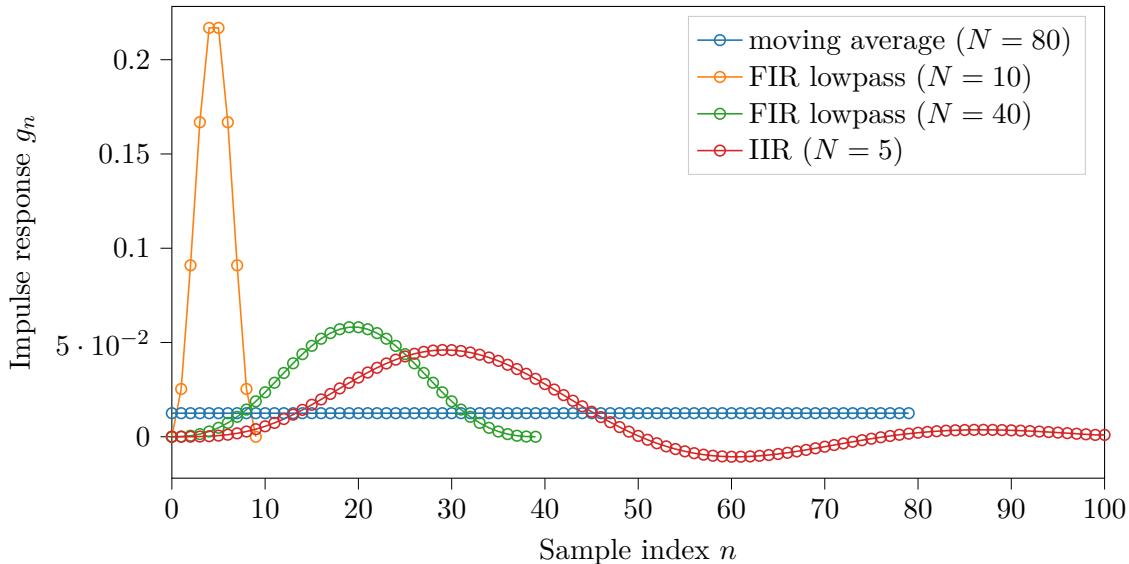
$$h[n] = \frac{1}{N} \underbrace{[1, 1, \dots, 1]}_N \quad (5.5)$$

The delay introduced by a digital filter can be quantified with the filter's group delay

$$\frac{\tau_g(f)}{T_s} = \frac{d\phi(f)}{df} \quad (5.6)$$

which is given normalized to the sampling time  $T_s$ . [Kammeyer2002] In case of a FIR filter with linear phase (with a symmetrical impulse response), the group delay is always constant over all frequencies and is only dependent on the filter length  $N$  (see [Kammeyer2002]):

$$\frac{\tau_g(f)}{T_s} = \frac{d\phi(f)}{df} = \frac{d\phi}{df} = \frac{N}{2} \quad (5.7)$$



**Figure 5.3:** Impulse responses of a moving average filter ( $N = 100$ ), a FIR lowpass ( $N = 50$ ,  $f_c = 0.1$  Hz) and a IIR Butterworth lowpass ( $N = 50$ ,  $f_c = 0.1$  Hz)

With a sampling time of  $T_s = 1/5\text{Hz} = 200\text{ ms}$  and  $N = 100$  the group delay is 10 s. In case of a steady operation this is acceptable, as the disturbances to compensate happen on a timescale in the order of several minutes. But in case of ongoing transients due to user changes to the control system parameters or short error bursts on the measured signal, this long delay causes problems and therefore should be reduced.

Therefore a more sophisticated digital filter is designed to replace the simple moving average.

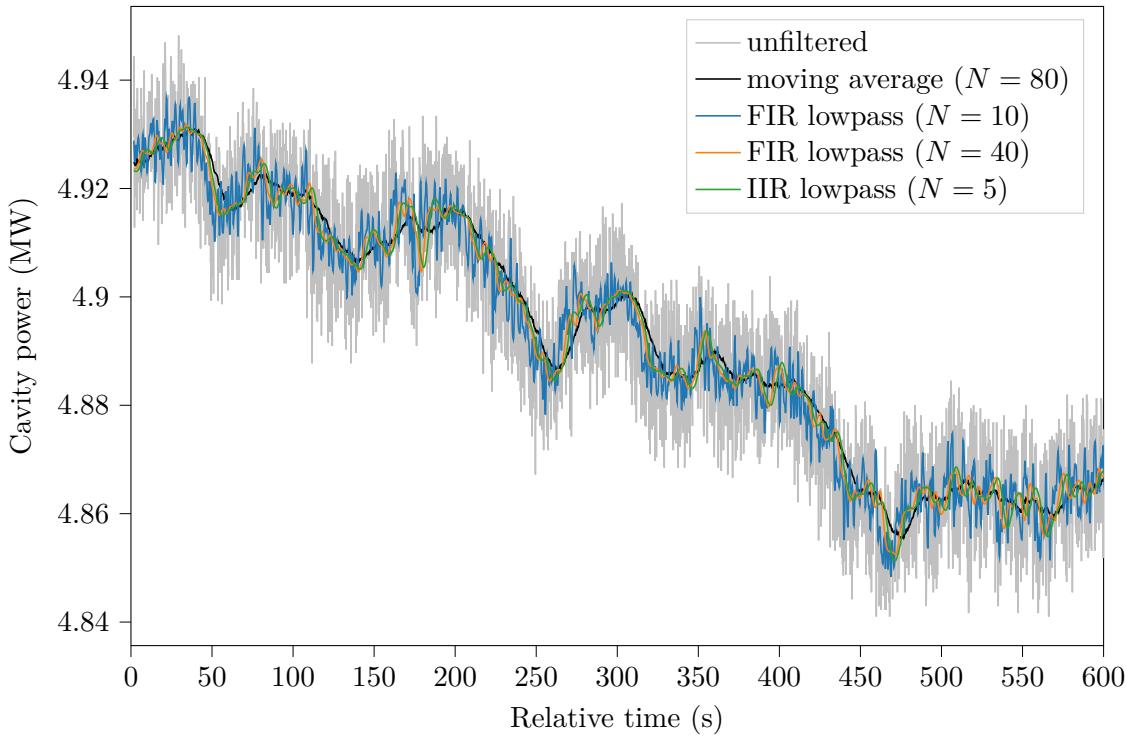
On the one hand, a FIR filter is designed with the Kaiser window method. This method starts with the desired frequency response, which is usually given piece-wise. In case of the low pass filter it is a step function at a cutoff frequency  $f_c$ . Then the IDFT is used to compute the corresponding impulse response  $h_{\text{IIR}}[n]$ , which is in general infinitely long. Windowing with e.g. a Kaiser window and then truncating the impulse response yields the impulse response of the desired FIR filter  $h_{\text{FIR}}[n]$ . [Oppenheim2010] With SciPy using `b=signal.firwin(N, fc, fs)`, the coefficients of a FIR with this method can be calculated.

On the other hand, an Infinite Impulse Response (IIR) filter is designed with the impulse invariance method and an analog Butterworth filter. This method could be interpreted as sampling the infinitely long analog impulse response. [Oppenheim2010] In SciPy using `b,a=signal.butter(N,fc,'lowpass',fs,)`, the coefficients of an IIR can be calculated with this method. For an IIR filter, the group delay cannot be calculated with Equation 5.7 and it is in general frequency-dependent.

Figure 5.3 shows the impulse responses of the moving average, the FIR lowpass and the IIR lowpass (truncated to  $N = 100$ ).

In Figure 5.4, the three filter types described above are compared by filtering a ten minute long segment of pre-recorded data. The filtering is done with the SciPy function `signal.lfilter()` which does causal filtering and does not compensate group delay<sup>2</sup>, so the results are the same as they would be for real-time data.

<sup>2</sup>In contrast to `signalfiltfilt()`, which applies the filter both forward and backward achieving zero phase/group delay, but this cannot be done for incoming real-time data.



**Figure 5.4:** Effects of the three different lowpass filters in Figure 5.3 on noisy data

The plot shows the FIR lowpass filter requiring ten times the number of coefficients to achieve about the same result as the IIR lowpass filter. Also the moving average filter has double the number of coefficients as the FIR lowpass filter, but there is still high frequency noise in the output (caused by the  $\text{sinc}(\cdot)$  shape of its frequency response  $H[f] = \text{DFT}\{h[n]\}$ ).

Compared to the FIR lowpass, the moving average offers no benefit besides its easy implementation. When comparing the FIR with the IIR approach, the IIR has the advantage of needing less coefficients, thus occupying less memory, which is not really an advantage when the control system is implemented on a personal computer, which typically has enough free memory to hold millions of floating point numbers. Also the IIR filter has a non-constant group delay and is not guaranteed to be stable like all FIR filters are.

For these reasons, in the following a FIR lowpass filter is used.

One example filter generated with `signal.firwin()` with a cutoff frequency  $f_c = 10 \text{ mHz}$  and order  $N = 10$  has the transfer function

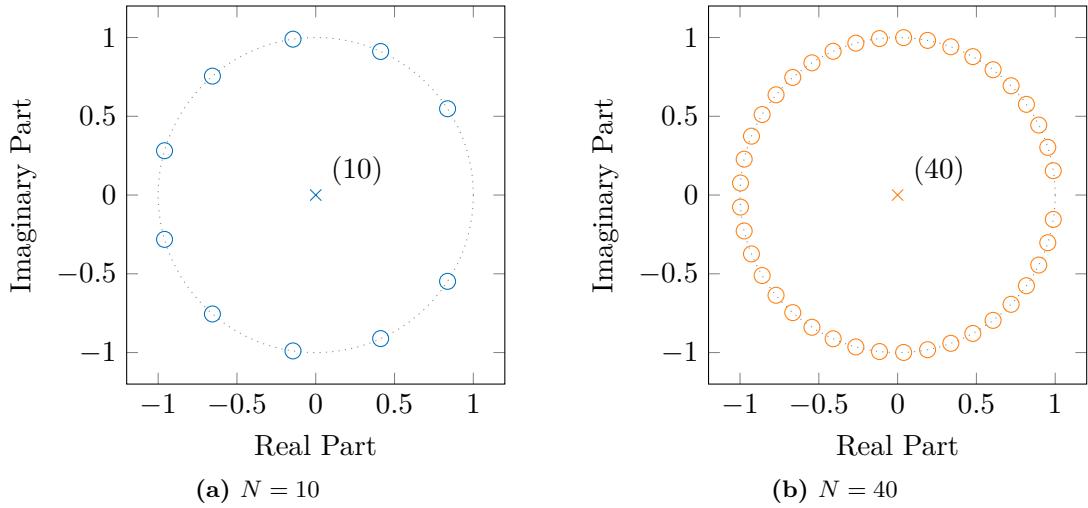
$$H[z] = \frac{1}{b_{10}z^{10} + b_9z^9 + b_8z^8 + b_7z^7 + b_6z^6 + b_5z^5 + b_4z^4 + b_3z^3 + b_2z^2 + b_1z + b_0} \quad (5.8)$$

with the coefficient vector  $\vec{b} = [b_{10}, \dots, b_0]$ , with

$$\vec{b} = [0.0876, 0.0896, 0.0911, 0.0922, 0.0929, 0.0931, 0.0929, 0.0922, 0.0911, 0.0896, 0.0876]. \quad (5.9)$$

Instead of stating the transfer function, a plot of the poles and zeros is a more intuitive representation. In Figure 5.5, the poles and zeros of the FIR filter with  $N = 10$  are compared to one with  $N = 40$ . In addition, Figure 5.6 shows the magnitude responses of such filters.

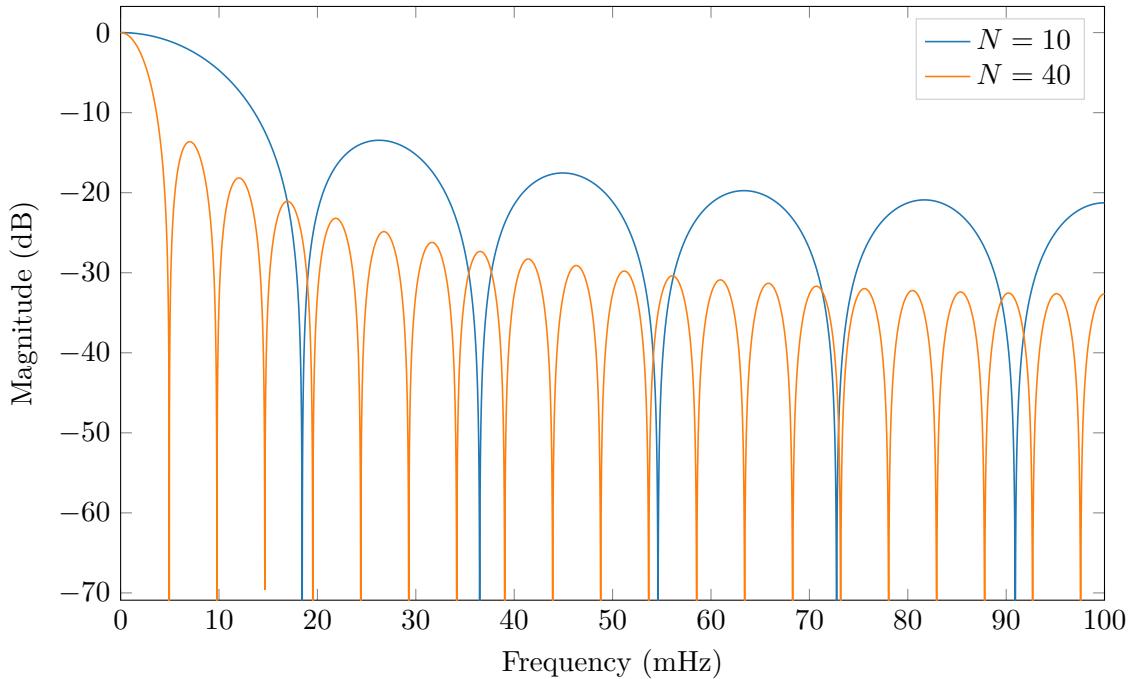
For a fixed cutoff frequency and variable order  $N$ , the trade-off for choosing  $N$  is between the group delay introduced by the filter (see Equation 5.7) and the width of the transition



**Figure 5.5:** Pole-Zero maps for two FIR filters with a common cutoff frequency  $f_c = 10$  mHz but different filter orders  $N$ ;  
○ denotes zeros, × denotes poles, ( $k$ ) is a  $k$ -times pole

band or in other words the sharpness of the filter. This can be seen by comparing the cases  $N = 10$  and  $N = 40$  for the FIR filters in Figure 5.4: While the smoothing-effect of the longer filter is obviously better, it also causes a higher group delay, which results in the filter result of the  $N = 40$  filter being shifted in time by  $\frac{40}{2}T_s = 4$  s compared to  $\frac{10}{2}T_s = 1$  s in case of  $N = 10$ . While a high smoothness is desired for the control system to reject high frequency noise, using a filter with a too high order can introduce a group delay high enough to shift the closed-loop from negative to positive feedback, thus making it unstable.

For the controller design in the next section and the later implemented real-time system, the cutoff frequency and the filter order are kept variable to leave room for improvement.



**Figure 5.6:** Magnitude response of two FIR filters with a common cutoff frequency  $f_c = 10$  mHz but different filter orders  $N$

### Real-time implementation of a FIR Filter in Python

Similar to Equation 5.4, a FIR filter designed with the SciPy function `sigal.firwin()` can be used in a causal manner to filter real-time data.

Applying the filter on pre-recorded data, like in Figure 5.4 can be done with `signal.lfilter()`. To use `signal.lfilter()` on sample-wise incoming real-time data, the “initial in” input and “final out” output of `signal.lfilter()` can be used to keep the filters state. This is demonstrated in Listing 5.2 by looping through pre-recorded data point-by-point.

**Listing 5.2:** Demonstration of the `zi` and `zf` variables when using `signal.lfilter()`

---

```

1 x=df2["F:RF:LLRF:01:GunCav1:Power:Out Value"].to_numpy()
2 y=np.array([])
3 zf=signal.lfilter_zi(b, 1)
4 for i in range(len(x)):
5     y0,zf=signal.lfilter(b, 1, [x[i]], zi=zf)
6     y=np.append(y,y0[0])

```

---

### 5.3 Controller Design

Based on the estimated model of the plant  $P(s)$  (or  $P[z]$ ) in section 5.1 and the type of measurement filter designed in section 5.2, in this section an appropriate controller to stabilize the plant is designed and its performance is evaluated.

Tuning the controller and testing its capabilities is performed both *offline* with simulations using the building blocks  $P[z]$ ,  $H[z]$  and the yet to be defined  $G[z]$  and *online* using a software implementation of the control system with the real hardware, i.e. FLUTE and the controllable attenuator.

#### 5.3.1 Choosing a Controller Type

The plant has been identified using a linear model in section 5.1 resulting in the LTI system  $P(s)$  (or  $P[z]$ ). The matching controller does not necessarily have to be a LTI system, but choosing a LTI system simplifies design and analysis and is a justifiable choice here, as there are no good reasons against it, yet.

The class of LTI controllers is dominated by the Proportional Integral Derivative (PID) controller and its variants. PID stands for “proportional”, “integral” and “derivative”, which are LTI systems themselves, performing scaling, integration or differentiation. Depending on the application, variants, such as the PI controller, or the PD controller are used as well.

According to [Aastroem1995] PID control is applicable for plants with order two or less. Compared to other types of controllers, using a PID controller has many advantages. It is quick to design and does not depend on an accurate plant model. Also after the designing step, the few free parameters can easily be presented to an operator and online fine tuning is possible.<sup>3</sup>

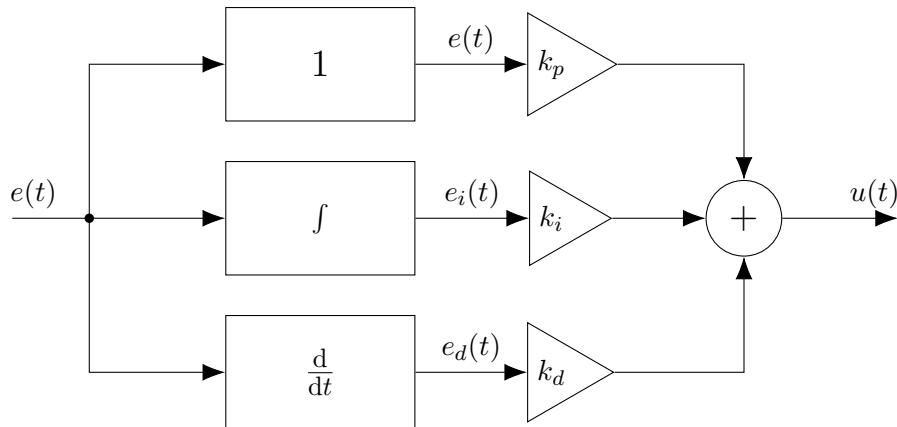
In the next section such a PID controller is designed. As it is to be implemented in software later, the design is done in discrete time.

#### 5.3.2 Designing a Discrete Time PID Controller

The output signal  $u(t)$  of a generic time continuous PID controller (see Figure 5.7) consists of the weighted sum of three error signals.

---

<sup>3</sup>In contrast a compensating controller is based on an accurate model of the plant and changes on the fly are difficult. [SergeZacher2010]



**Figure 5.7:** Block diagram of a generic PID controller

The first is simply the controller input error signal  $e(t)$  scaled by the gain  $k_p$ . This is the proportional part of the PID controller. The integral part is calculated by computing the running time integral

$$e_i(t) = \int_0^t e(\tau) d\tau. \quad (5.10)$$

This is then scaled by the constant  $k_i$ . To get the derivative part, the derivative

$$e_d(t) = \frac{d}{dt} e(t) \quad (5.11)$$

is calculated and weighted with  $k_d$ . All three are then summed to get  $u(t)$  in the so called parallel form (see [Dodds2015]):

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{d}{dt} e(t) \quad (5.12)$$

Often instead of the parallel form, the PID controller is stated in *standard form*. Instead of using the gains  $k_{p,i,d}$ , the parameters proportional gain  $K$ , integral time  $T_i$  and derivative time  $T_d$  are used. [Aastrom1995] With the conversions

$$K = k_i \quad (5.13)$$

$$T_i = \frac{k_p}{k_i} = \frac{K}{k_i} \quad (5.14)$$

$$T_d = \frac{k_d}{k_p} = \frac{k_d}{K} \quad (5.15)$$

the PID controller in standard form is

$$u(t) = K \left[ e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{d}{dt} e(t) \right]. \quad (5.16)$$

The transfer function  $G(s)$  is given by the Laplace of  $u(t)$  using the computation rules (see [Leon2015])

$$\mathcal{L} \left\{ \int_0^t y(\tau) d\tau \right\} = \frac{1}{s} Y(s) \quad (5.17)$$

$$\mathcal{L} \left\{ \frac{d}{dt} y(t) \right\} = sY(s) \quad (5.18)$$

as

$$G(s) = K \left[ 1 + \frac{1}{sT_i} + sT_d \right]. \quad (5.19)$$

To get the discrete transfer function, either the Laplace transform  $G(s) = U(s)/E(s)$ , is discretized or the  $\mathcal{Z}$  transform of  $u[n]$  is calculated. First, to get  $u[n]$ , the derivative in Equation 5.11 is approximated by

$$e_d[n] = \frac{e[n] - e[n-1]}{T_s}. \quad (5.20)$$

This assumes  $e(t)$  is sampled with a sampling rate of  $f_s = 1/T_s$  to get  $e[n]$ . In a similar fashion, the integral in Equation 5.10 is approximated by

$$e_i[n] = e_i[n-1] + e[n]T_s. \quad (5.21)$$

Using the shift rule of the  $\mathcal{Z}$  transform (see [Leon2015])

$$y[n-k] \circ \bullet z^{-k} Y[z], \quad (5.22)$$

the discrete transfer function of the PID controller is

$$G[z] = K \left[ 1 + \frac{T_s}{T_i} \frac{z}{z-1} + \frac{T_d}{T_s} \frac{z-1}{z} \right]. \quad (5.23)$$

In the time domain, this becomes

$$u[n] = k_p e[n] + k_i \left( \underbrace{e_i[n-1] + T_s e[n]}_{e_i[n]} \right) + k_d \left( \underbrace{\frac{1}{T_s} e[n] - e[n-1]}_{e_d[n]} \right) \quad (5.24)$$

### 5.3.2.1 Controller Tuning

Next the three free parameters  $k_{p,i,d}$  or  $K, T_{i,d}$  are to be chosen in such a way that the controller has optimal performance. This process is called *tuning*. Tuning of the parameters can be performed online or offline.

Online tuning is done by using the physical<sup>4</sup> plant. The most popular member of this class is the Ziegler-Nichols tuning. The method relies on experiments and tabulated values. The mostly used variant uses one measured step response. [Ziegler1942]

Using this method at FLUTE produced mixed results. While being a very simple and fast process, the resulting controller is often unstable. This is partially due to errors when extracting the tabulated values from the noisy step response, but also the method intrinsically leads to poor stability margins. [Aastroem1995] Nonetheless the Ziegler-Nichols method yields a usable starting point, if the strategy is to fine tune the controller manually by intuition and experience of the user.

As the Ziegler-Nichols method does not yield an acceptable parameter set and the method combined with fine tuning by hand takes a considerable amount of time, next tuning the parameters offline using only the plants transfer function and the measurement filter is done. The offline tuning can be done analytically or using different numerical optimization strategies. In [DiazRodriguez2019] analytical methods, such as the internal model control design or the pole placement design are discussed. Both require the system's transfer function in a closed form.

Tuning by numerical optimization chooses the parameters by a simulation or measured data. Goal of the optimization is to minimize a cost function  $J$  like

$$J(\theta) = \sum_{n=0}^{\infty} e_{\theta}[n]^2 \quad (5.25)$$

with the parameter vector  $\theta = [k_p, k_i, k_d]$  or  $\theta = [K, T_i, T_d]$ :

$$\theta_{\text{opt}} = \underset{\theta}{\operatorname{argmin}} J(\theta) \quad (5.26)$$

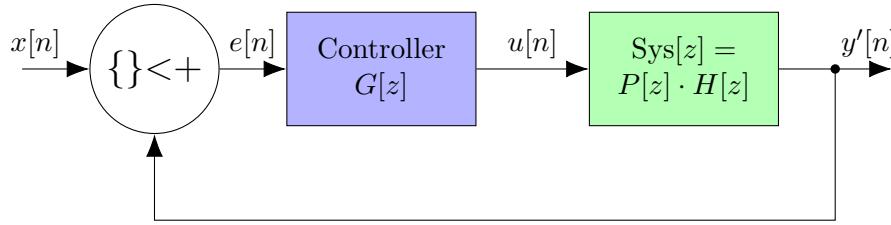
A convenient way to do such an optimization by simulation using the transfer function of the estimated plant is the Matlab *PID Tuner*.

#### Offline Tuning the Controller with the Matlab PID Tuner

The Matlab PID Tuner accepts different kinds of system models used by the Matlab/Simulink ecosystem. It is possible to directly use estimated models, models defined via

---

<sup>4</sup>“Physical” in the sense that one could touch it. (But should one?)



**Figure 5.8:** Required system architecture for the Matlab PID Tuner; PID Tuner input is the system block  $Sys[z]$  (green), generated output is the controller  $G[z]$  (blue)

their transfer function or the zero-pole-gain representation, Simulink models or combinations of those<sup>5</sup>.

The manual of the PID Tuner (see [pidTunerMan]) describes the expected input (see Figure 5.8). The feedback path has to have unity gain, i.e. there is no measurement filter allowed. Therefore the measurement filter is moved before the junction where the output  $y[n]$  would normally be measured. Then the measurement filter  $H[z]$  is combined with the plant  $P[z]$  to form  $Sys[z]$ . For that reason  $y[n]$  becomes an internal signal of  $Sys[z]$ . However the new system output  $y'[n] = h[z] * y[n]$  is not too different from the old one, since  $H[z]$  is designed to remove high frequency noise but retain the rest of  $y[n]$ .

The manufacturer's documentation in [pidTuner] does not disclose any internals of the PID Tuner nor state which optimization technique is used, but three tuning objectives are stated:

- Stability: The closed loop should be stable (that is BIBO stable as defined in 1)
- Performance: The closed loop system tracks the input well and rejects disturbances as rapidly as possible (see subsection 2.3.1)
- Robustness: A gain and phase margin accounts for errors in the system model (see section 5.1)

Using Listing 5.3, the estimated plant  $P[z]$  is loaded, a measurement filter  $H[z]$  is generated and the combination  $Sys[z] = P[z]H[z]$  is fed into the PID Tuner.

**Listing 5.3:** Matlab script to generate an input system for PID Tuner

```

1  {}<ρ design lowpass measurement filter H
2  N=10;
3  lpFilt1 = designfilt('lowpassfir', 'FilterOrder', N, 'CutoffFrequency', ...
4                      0.01, 'SampleRate', 0.2, 'DesignMethod', ...
5                      'window', 'Window', 'kaiser');
6
7  % convert filter to a dynamic system
8  [z1,p1,k1]=zpk(lpFilt1);
9  H1=zpk(z1,p1,k1,0.2);
10
11 % load estimated transfer function and convert it to discrete form
12 load('P2ZU.mat')
13 P=c2d(idtf(P2ZU),0.2);
14
15 Sys1=tf1*H1; %Combine plant P and measurement filter H
16 pidTuner(Sys1); %Launch pidTuner

```

Using the PID Tuner GUI (see Figure A.2), the controller is designed by changing the design parameters **Response Time** (RT) and **Transient Behavior** (TB). Changing the response time (in seconds) influences how fast the controller acts on changes. With the

<sup>5</sup>Series connection can be established by multiplying the models transfer functions.

**Table 5.2:** Parameters of a discrete time PID controller in parallel form calculated with the Matlab PID Tuner;  $N$  is the order of the used measurement filter

Name	$N$	RT (in s)	TB	$k_p$	$k_i$	$k_d$
$G_1[z]$	10	5	0.2	-0.000 577	-0.000 197	-0.000 421
$G_2[z]$	10	10	0.2	-0.000 155	-0.000 111	$-5.43 \times 10^{-5}$
$G_3[z]$	40	5	0.2	-0.000 559	$-2.27 \times 10^{-5}$	-0.000 396
$G_4[z]$	40	10	0.2	-0.000 35	$-9.68 \times 10^{-5}$	-0.000 316

transient behavior setting, the allowed over- and under-shoots, that is, the deviation above and below the final value for  $t \rightarrow \infty$  can be set qualitatively.

With this tool four different controllers are designed, two for each measurement filter's order of  $N = 10$  and  $N = 40$ . The transient behavior with 0.2 for all controllers is chosen as a compromise between speed and overshoot behavior. The optimized PID parameters (for a discrete time PID controller in parallel form) are listed in Table 5.2.

### 5.3.3 Analyzing the Input Tracking and Disturbance Rejection

The PID Tuner defaults to show the input tracking step response (as in Figure A.2). The input tracking is calculated based on Equation 2.35. To calculate it, the Matlab function `step()` is used, which generates the step response plot of a dynamic system, in this case  $F_T$ . As the whole system only consists of linear building blocks, using an arbitrary step height of 1 is possible. The response is then normalized in such a way that the final value is also 1.

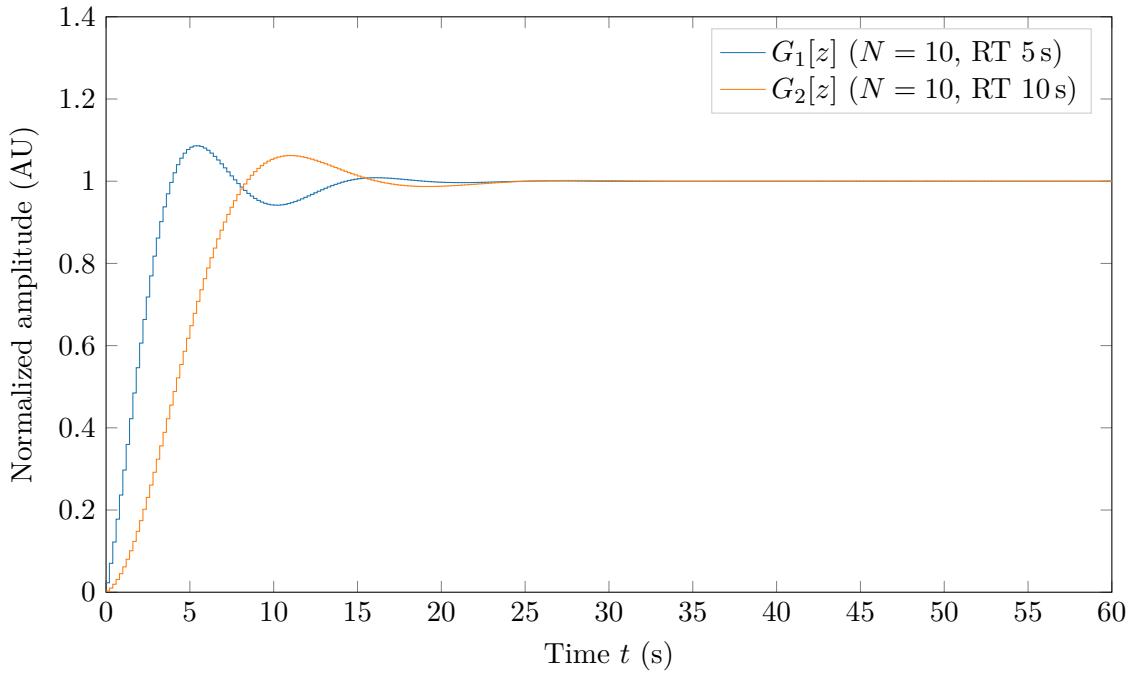
In Figure 5.9 and Figure 5.10 the input tracking step responses for all four controllers are plotted. Note the different  $t$  scales. As expected, for longer measurement filters, i.e. filters of higher orders, the controller has to act less aggressive. This causes longer settling times. Especially Figure 5.10 shows that setting a shorter response time does not automatically cause the controller to set the output faster to its final value. While for the shorter response time, the set value is reached quicker, there is also a strong oscillation, so choosing the longer response time (see  $G_4[z]$ ) can be beneficial.

As the set-point for FLUTE only changes occasionally, basically a fixed set-point controller is needed. Therefore the disturbance rejection is more important than the input tracking when analyzing the controllers performance. The step responses of the disturbance rejection  $F_{DR}$  are calculated in a similar way as the input tracking, using Equation 2.38 and `step()`. The  $y$  axes are normalized so that the initial value is 1 and the final value for  $t \rightarrow \infty$  is 0.

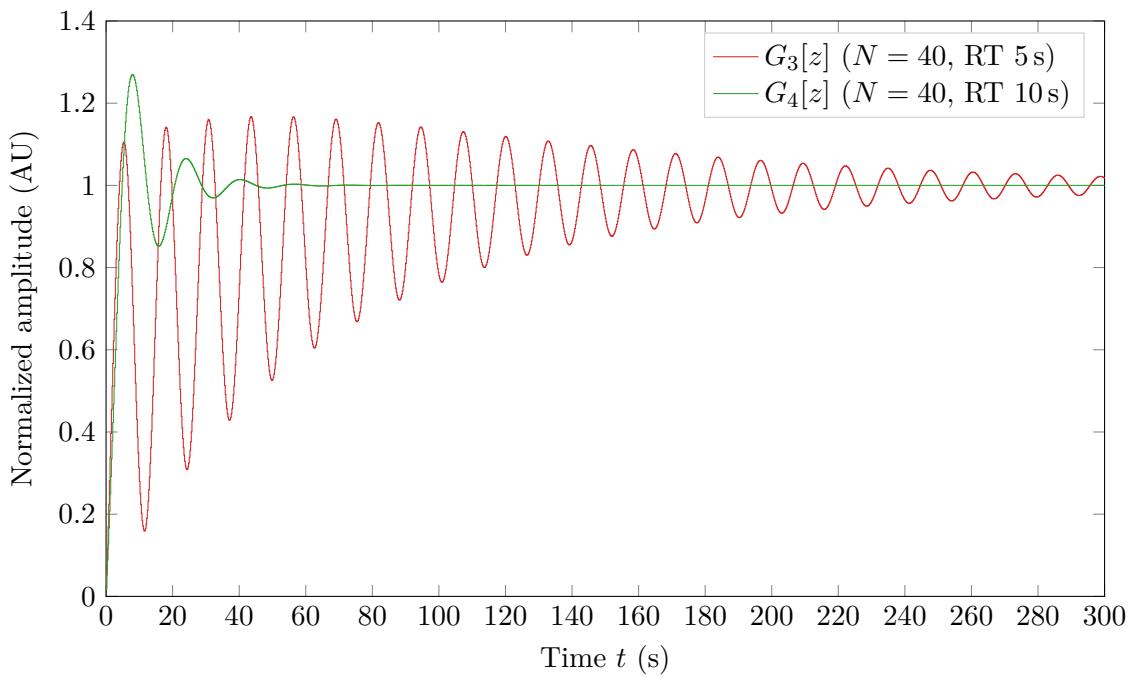
In Figure 5.11 and Figure 5.12 the results for all controllers in Table 5.2 are plotted.

This shows that even for an optimistically low measurement filter order of  $N = 10$ , the settling time is in the order of slightly under a minute. This is sufficient for the application, but has more of a practical drawback: when fine-tuning the PID parameters manually on the machine, it is necessary to wait a few minutes after setting new parameters before assessing the change.

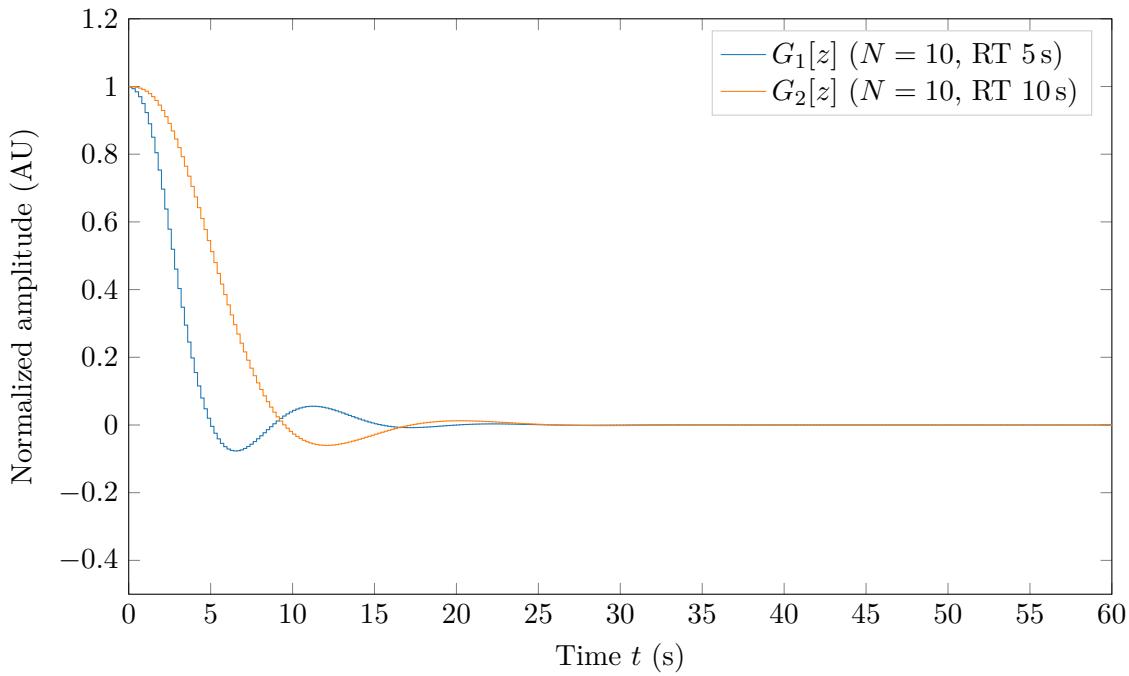
Especially when looking at the plots of  $G_3[z]$ , it is obvious that there can be stability issues when using a too aggressive controller. For that reason in the next section the stability of the four controllers is analyzed.



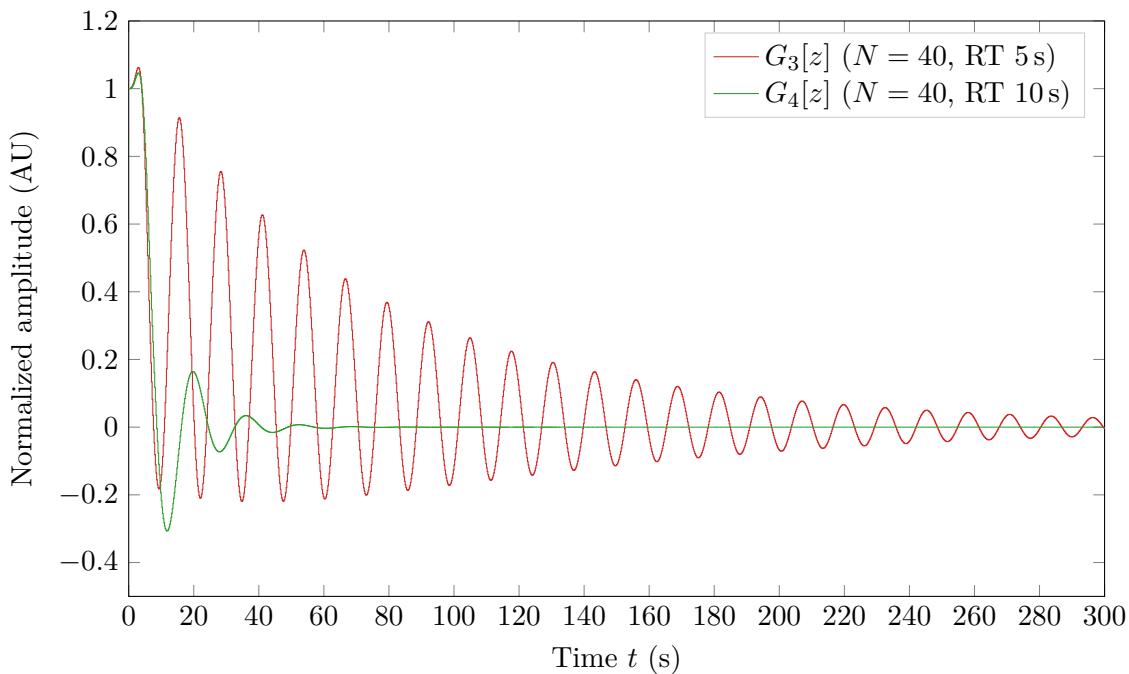
**Figure 5.9:** Step response of the input tracking  $F_T$  for controller  $G_1[z]$  and  $G_2[z]$ , designed with the plant  $P[z]$  and a measurement filter of order  $N = 10$ , response time goal to 5 s and 10 s



**Figure 5.10:** Step response of the input tracking  $F_T$  for controller  $G_3[z]$  and  $G_4[z]$ , designed with the plant  $P[z]$  and a measurement filter of order  $N = 40$ , response time goal to 5 s and 10 s



**Figure 5.11:** Step response of the disturbance rejection  $F_{DR}$  for controller  $G_1[z]$  and  $G_2[z]$ , designed with the plant  $P[z]$  and a measurement filter of order  $N = 10$ , response time goal to 5 s and 10 s



**Figure 5.12:** Step response of the disturbance rejection  $F_{DR}$  for controller  $G_3[z]$  and  $G_4[z]$ , designed with the plant  $P[z]$  and a measurement filter of order  $N = 40$ , response time goal to 5 s and 10 s

### 5.3.4 Analyzing the Controller Stability

In this section the stability of the controllers  $G_i[z]$  from the last section is evaluated.

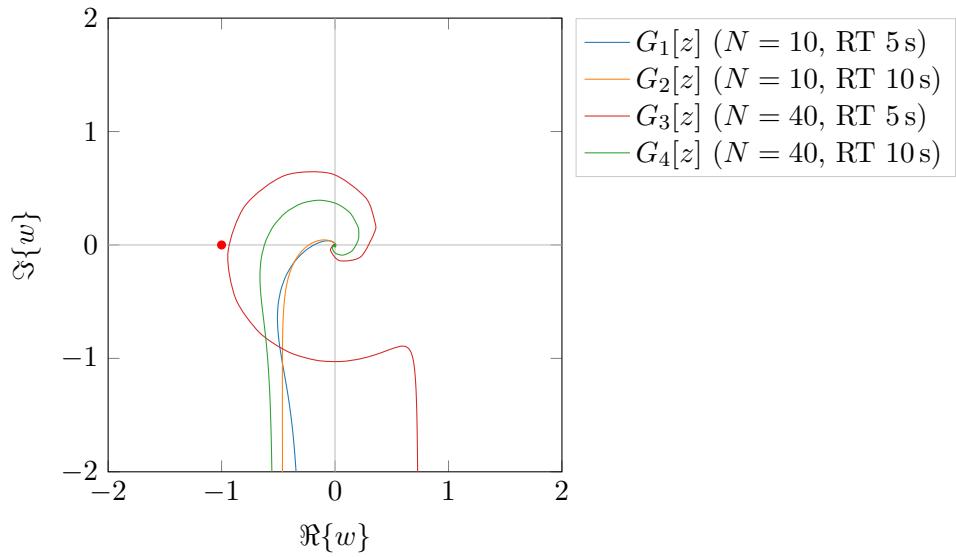
For that the Nyquist criterion according to 2 is used. With the Matlab function `nyquist()`, the locus  $w = F_o[w = j2\pi f]$  of the open loops

$$F_o[z] = G[z]P[z]H[z] \quad (5.27)$$

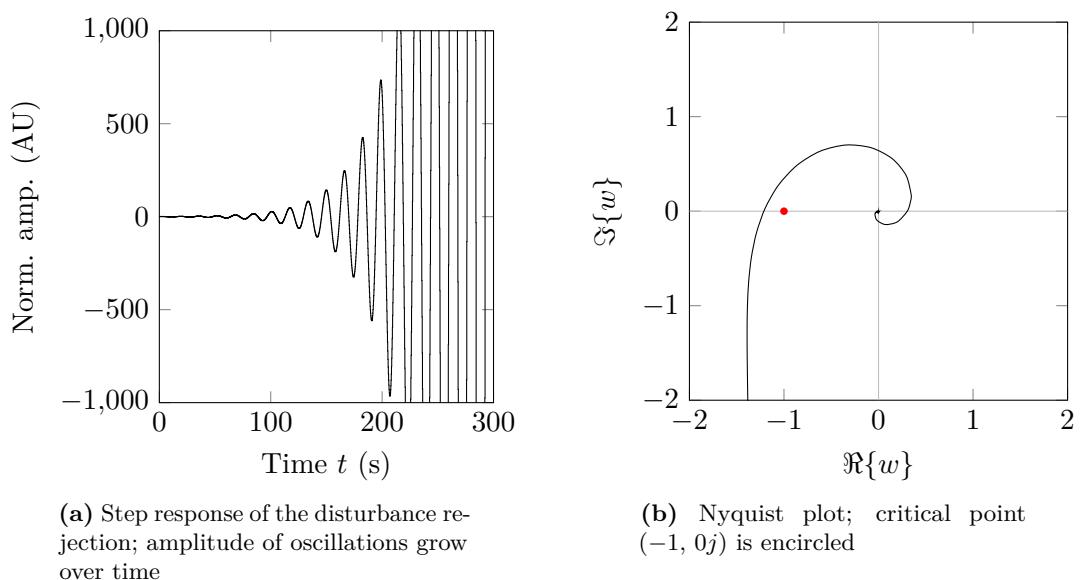
are calculated. For all controllers designed, they are plotted in Figure 5.13.

This shows the gain margin for  $G_3[z]$  (in combination with the  $N = 40$  filter), that is the distance to the critical point  $w = (-1, 0j)$ , is already very small. So in addition to its oscillatory behavior and the longer settling time, there is also the risk that the system can become unstable if there are even small system parameter changes.

In Figure 5.14 another issue is highlighted. If a controller is designed with a certain measurement filter order  $N = N_0$  in mind, but is used together with a filter of order  $N = N_1 > N_0$ , it is very likely the closed loop system becomes unstable. In the figure the controller  $G_1[z]$  is used together with the  $N = 40$  filter. Both the disturbance rejection and the Nyquist plot show the closed loop to be unstable.



**Figure 5.13:** Nyquist plot to analyze the stability of the closed-loop control system based on the locus plot of the open-loop system; Note the critical point  $(-1, 0j)$  is always on “left” of the curve indicating stability



**Figure 5.14:** Two signs of an unstable closed loop system, in this case caused by a measurement filter with too high order and/or a too aggressive controller

### 5.3.5 Offline Evaluation with Measured Data

Now the performance of the controller  $G_1[z]$  together with the filter order  $N = 10$  should be evaluated with measured data before testing it with FLUTE. This simulation is called offline evaluation.

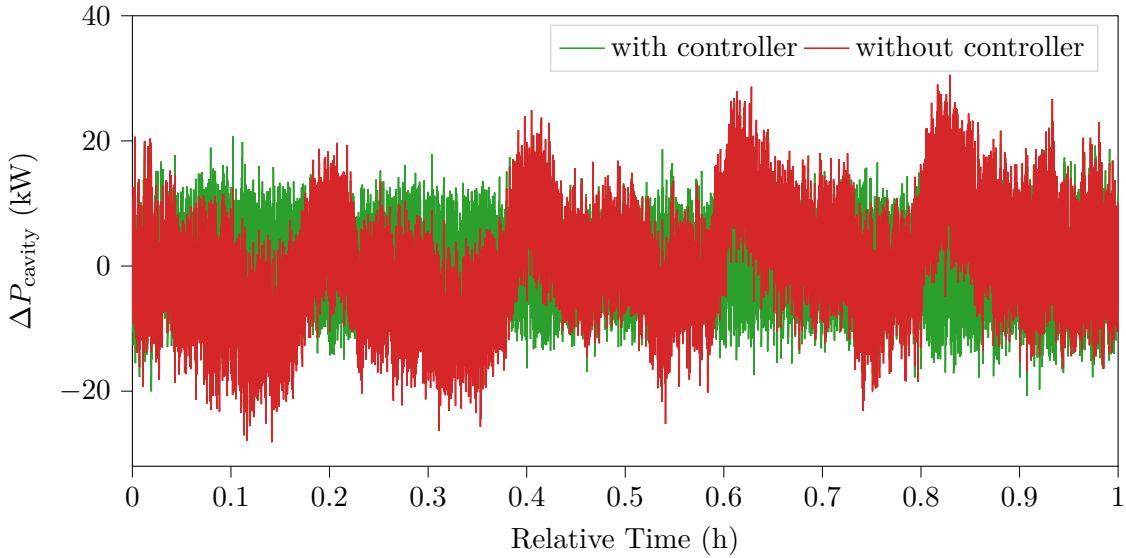
Offline evaluation of the control system is potentially less accurate due to an incomplete or erroneous model. But on the upside it requires no access to FLUTE and no potential downtime of the machine. Also with modern computers the simulation time advances much faster than real-time. Another big advantage is repeatability. With a pre-recorded data-set as the disturbance input, simulations are consistent and do not depend on a changing environment or system parameters.

The offline evaluation is done with Simulink, a block diagram development and simulation environment based on Matlab. The Simulink model (see Figure A.3) uses the disturbance

$$d[n] = \Delta P_{\text{cavity}}[n] = P_{\text{cavity}}[n] - \mu P_{\text{cavity}} \quad (5.28)$$

with  $\mu P_{\text{cavity}}$  being the time average of  $P_{\text{cavity}}[n]$  over one hour.  $P_{\text{cavity}}[n]$  is re-sampled to  $T_s = 0.2$  s. With this calculation of  $d[n]$ , it is assumed that the disturbance is the deviation of the cavity power from a (theoretical) set-point of  $\mu P_{\text{cavity}}$ .

With the simulation the effect of adding  $d[n]$  to a system without any feedback and a system with the controller  $G_1[z]$  and the measurement filter is compared. The result is shown in Figure 5.15



**Figure 5.15:** Output of the Simulink model in Figure A.3; step size  $T = 0.2$  s, end time 3600 s. Simulation time on a computer equipped with an Intel i7-3770: 3.5 s

## 5.4 Implementation of the Control System in Software

In this section the findings from the beginning of this chapter are used to implement a time discrete control system, that is a time discrete PID controller that is used to drive the controllable attenuator based on filtered measurements from the system.

The system should be implemented as software that runs on most personal computers and requires few external dependencies to make it as portable as possible. The other core requirements can be summarized to be:

- Communicate with EPICS (over an Ethernet connection) to read in data such as  $P_{\text{cavity}}[n]$
- Provide means to filter the incoming data with a measurement filter  $H[z]$  with variable order  $N$  and cutoff frequency  $f_c$
- Calculate the control error  $e[n]$  and based on that the controller output  $u'[n]$ . Then convert the controller output, an attenuation, to the matching control voltage  $V_{\text{control}}$
- Communicate with the Keysight 34972A over VXI-11 (over an Ethernet Network) to set the control voltage  $V_{\text{control}}[n]$  of the attenuator
- The control routine of the program needs to be light-weight enough so a scheduler can call it faster than five times a second to achieve a sampling time of 0.2 s
- Show the input, the output and the error signals to the user on a GUI
- Provide graphical input elements to let the user modify the measurement filter and the controller parameters ( $k_p$ ,  $k_i$ ,  $k_d$ )
- Log the input, output and parameters to disk for later reference

Since many other choices depend on it, first the programming language has to be picked. As Python was used in earlier chapters and the communicating abilities to both EPICS and the Keysight 34972A were already proven, it is the obvious choice.

From there on the GUI framework is the next decision to be made. For Python popular choices are *Tkinter*, a built in implementation of the Tk/tcl GUI toolkit, *wxPython*, a

Python binding to the cross-platform GUI library wxWidgets, or *PyQt*(see [**pyqt**]), a set of Python bindings for the Qt GUI framework. [**qt**] While Tkinter has the advantage of being built into the Python language, wxPython offers more functionality and is more widely adopted. PyQt profits from the large Qt ecosystem, so is it for example possible to develop the GUI separately from the code using *Qt Designer*.

For (live-) plotting of data in the GUI, the standard library in Python is *matplotlib*. [**Hunter2007**] It can be used with all three plotting libraries. However the biggest drawback is the possible update speed of the plots. With more than about a hundred points on the screen, the update frequency drops to well below 1 Hz. A much faster alternative is *pyqtgraph* (see [**pyqtgraph**]), a scientific plotting library written in Python and using the Qt GraphicsView. It is only compatible only with PyQt (or the PySide alternatives).

Therefore PyQt and pyqtgraph are used to build handle the GUI and plot the live data. Also Qt Designer is used to model the GUI graphically.

The control algorithm is implemented by directly using the equations derived in sub-section 5.3.2. The time domain representation of the PID controller is (restated from Equation 5.24)

$$u[n] = k_p e[n] + k_i \left( \underbrace{e_i[n-1] + T_s e[n]}_{e_i[n]} \right) + k_d \left( \underbrace{\frac{1}{T_s} e[n] - e[n-1]}_{e_d[n]} \right) \quad (5.29)$$

When translating Equation 5.29 to code, the recursion ( $e_i[n]$  depending on  $e_i[n-1]$ ) is solved by introducing a variable that keeps track of  $e_i[n]$  over time. The last value of  $e[n]$ ,  $e[n-1]$  is kept in memory for the derivative part. An example in pseudo code is listed in Listing 5.4.

**Listing 5.4:** PID controller implemented in pseudo code

---

```

1 while(true) {
2     e := x_set - x_actual
3     e_i := e_i_old + e*T_s
4     e_d := (e-e.old)/T_s
5     e_i.old := e_i
6     e.old := e
7     wait_sec(0.2)
8 }
```

---

To get the timing right and to allow the GUI to be responsive while the control algorithm runs in the background, a scheduler that activates a callback function every 0.2 s should be used. Since in section 4.2 the Advanced Python Scheduler (AP scheduler) is used successfully, it should also be used for the control system. Unfortunately the AP scheduler is not compatible with PyQt and using it interferes with the internal Qt timing systems. For this reason, a timer is constructed instead with the **QTimer** class. [**qtimer**]

Logging the data to disk is done by using the same **QTimer** as is used for the control algorithm, to write one line of CSV data to disk each time the timer is triggered.

With these building blocks ready, the GUI is build with Qt Designer and based on the **.ui** file generated from Qt Designer the GUI Application is implemented. For a screenshot of the program running see Figure A.4.

## 5.5 Evaluation of the Control System (Online)

After evaluating the controller offline, it is now time to evaluate the performance of the control system on the machine. To do so, FLUTE is operated with and without the control system (using the controller  $G_1[z]$  and the FIR filter with order  $N = 40$ ) switched on for 6 h each. Before the test, FLUTE is allowed to run a few hours for all components to reach operating temperatures. The result of the test is shown in Figure 5.16. Note, that in this test about 7 h into the experiment there was an unexpected shutdown of FLUTE and the corresponding block of data is removed before further processing.

The time plot Figure 5.16 and also the spectrogram in Figure 5.17 show the cavity RF power approximately reaches stationarity in the [0, 6] hour interval respectively in the [6, 12] hour interval. This allows the spectrum for these two blocks to be estimated using a periodogram method. In Figure 5.19 the spectra for each case are shown. The corresponding time data is plotted in Figure 5.18

With the periodogram data, the control system success can be measured with the most prominent noise metric.

Next, the cases controller on and controller off should be compared with the relative standard deviation. Measuring and displaying the relative standard deviation %STD is also supported on the LLRF CSS panel at FLUTE, so while the experiments run, the values are checked from time to time. It can be observed that different values are shown over time although the plotted time signal looks stationary by eye. This suggests that the relative standard deviation does not only depend on the window size  $T$  over which it is calculated, but also heavily on the absolute position in time  $t$ . This effect is more visible for small window sizes. The issue is illustrated in Figure 5.20, where  $\%STD = \%STD(t, T)$  is plotted as a function of the time  $t$  and the window size  $T$ .

For that reason, for small window sizes, time averages of %STD should be used instead of single values. For the measured  $\%STD(t, T)$  for the controller off and the controller on cases,  $STD\%(T)$  is plotted in Figure 5.21. This plot shows what is already suggested by Figure 5.20: with the %STD metric, the system seems *less* stable with the controller on, if the %STD is calculated over small window sizes e.g.  $T = 1$  min. For very long windows, the dependency on the window position becomes negligible. So for the comparison with the other metrics,  $\%STD(T = 4$  h) is used.

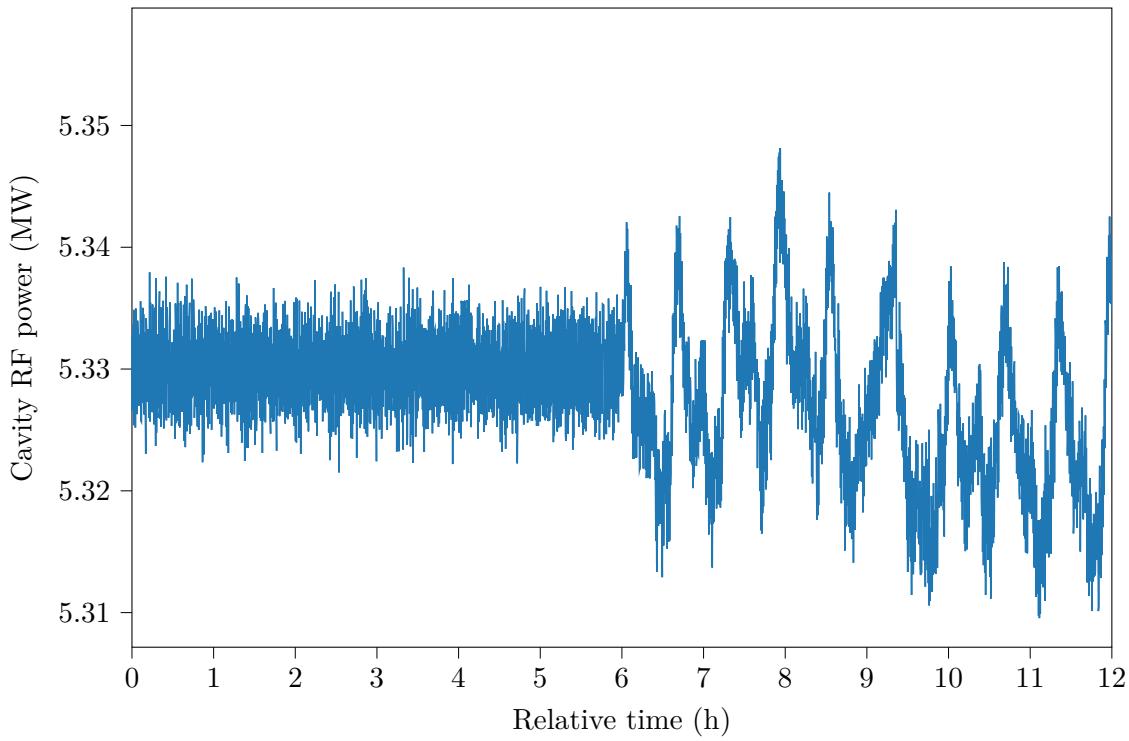
With the metrics from section 2.4, the success of the control system is assessed in Table 5.3.

### Result:

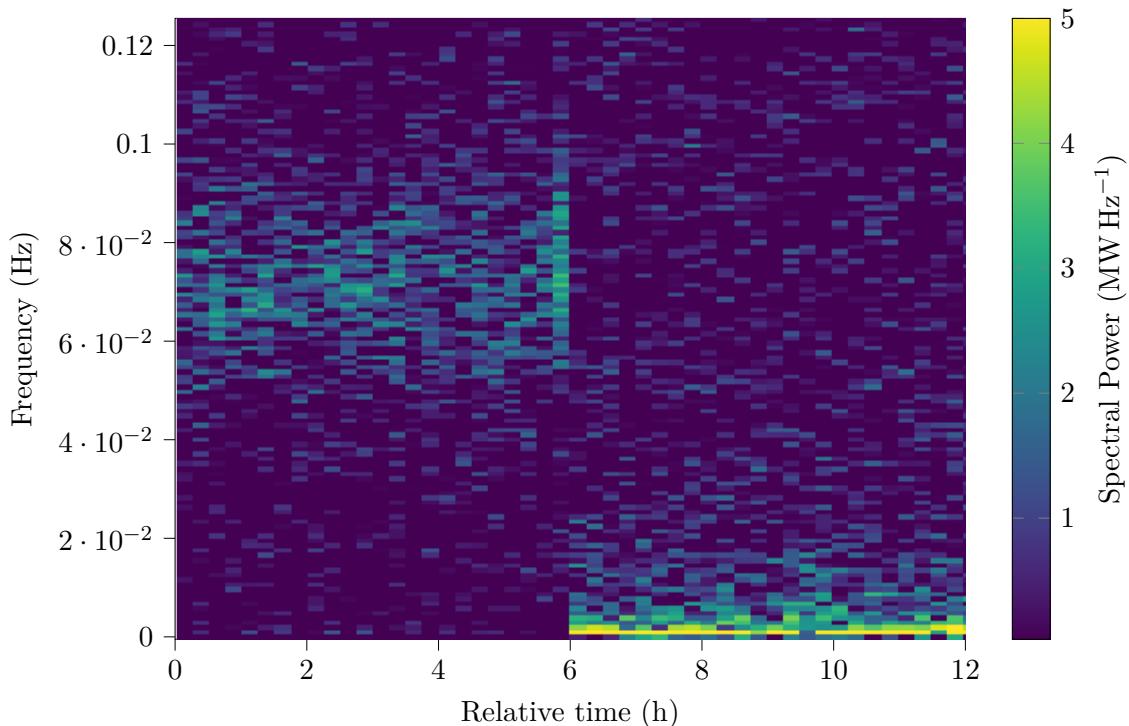
Regarding  $\%STD = 0.0599 \%$ , achieved with the control system, shows the **success** of the control system. With the controller, the variation of the cavity RF power can be brought down to lie inside the 0.01 % to 0.1 % goal, set in Equation 3.9.

**Table 5.3:** Quantitative assessment of the controllers performance

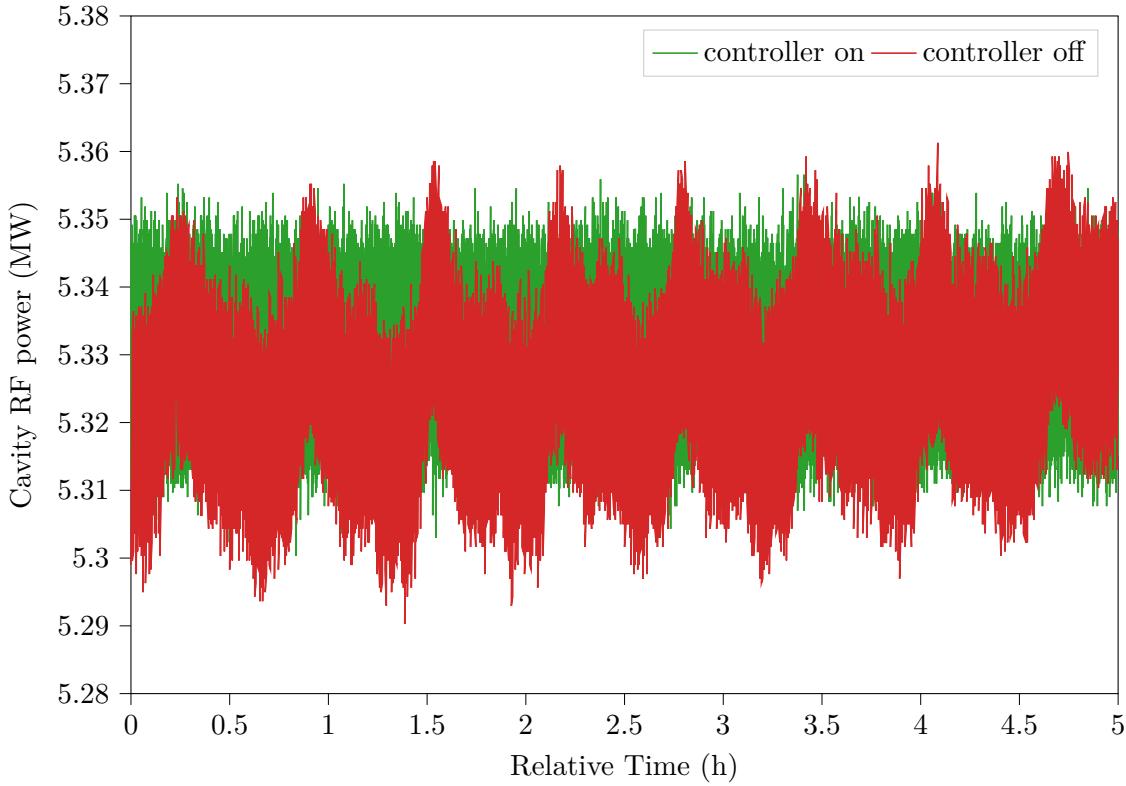
Metric	Controller off	Controller on	Controller off/Controller on
$\%STD(4\text{ h})$	0.115 59	0.059 922 5	1.928 91
$MSE$	37.639	0.101 33	371.44
$MPN$	0.0170	0.0151	1.126



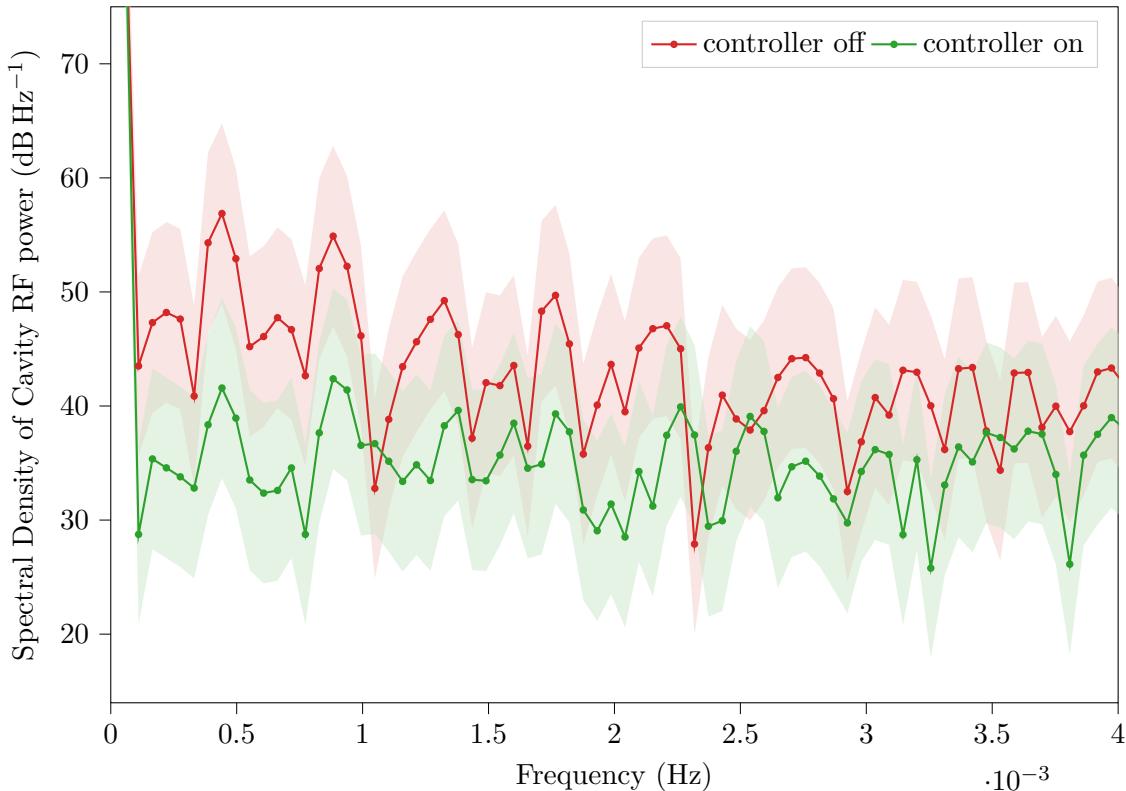
**Figure 5.16:** Cavity power over about 15 h (about three hours of downtime removed for clarity around the 7 h mark); control system switched off at 6 h (recording started 2021/05/01 20:00)



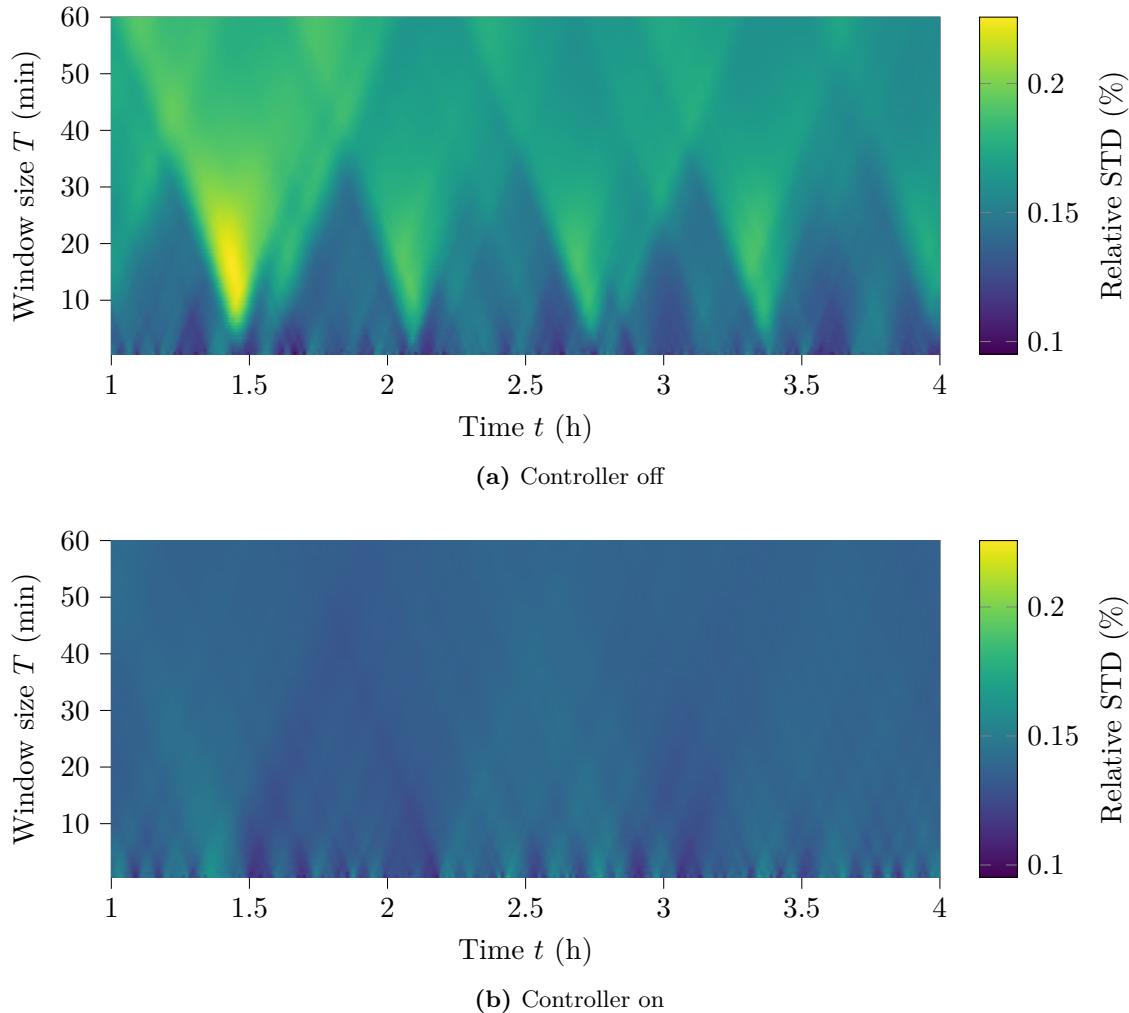
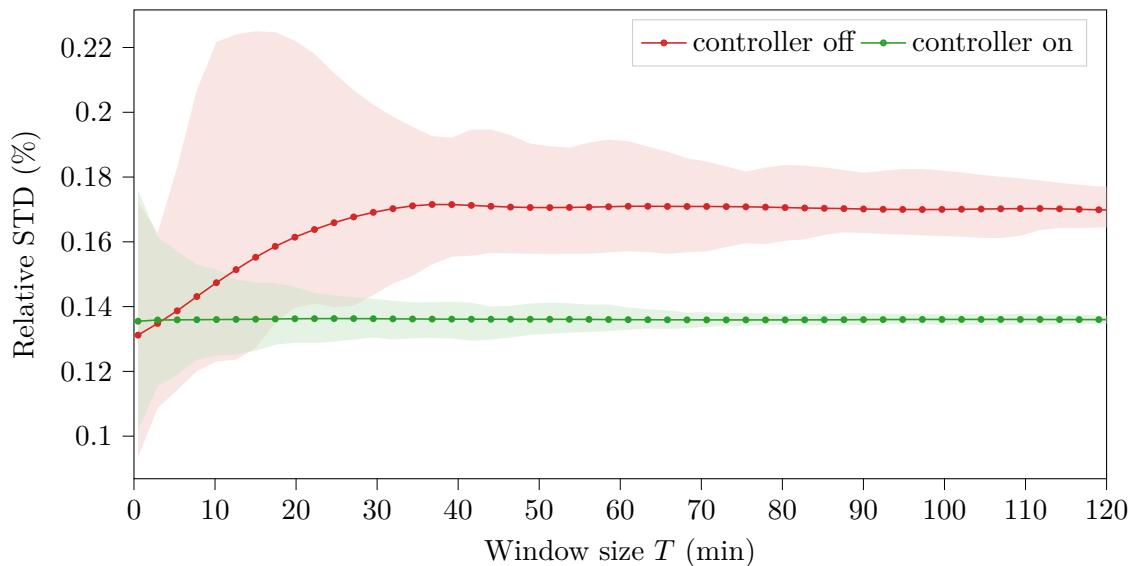
**Figure 5.17:** Spectrogram of the cavity power in Figure 5.16



**Figure 5.18:** The cavity RF power plotted over time for the two cases control system on and control system off (recorded one after the other and overlayed for comparison)



**Figure 5.19:** Power spectrum of the plots in Figure 5.16 computed with Welch's method; shaded areas show the uncertainty according to Equation 2.32

**Figure 5.20:** Relative standard deviation  $\%STD(t, T)$ **Figure 5.21:** Relative standard deviation  $STD\%(T)$ , shaded areas show  $\min\{STD\%(t, T_0)\}$  and  $\max\{STD\%(t, T_0)\}$ , solid lines show  $\text{mean}\{STD\%(t, T_0)\}$  for a fixed window size  $T = T_0$

## 5.6 Further Improvements to the Control System

In this section, two methods for improving the control system's performance are examined. First the closed-feedback loop is supplemented with an additional input for one disturbance source, the gun's body temperature. A different approach tested is switching from the gun cavity RF power to the electron charge, as measured by the Faraday cup at the end of the low-energy section, as the controlled signal.

### 5.6.1 Changing the Controller Architecture: Disturbance Feed-Forward of the Gun Temperature

Instead of changing only the parameter of the PID controller or switching to another type of controller, the strategy presented here relies on changing the architecture over the standard form (see Figure 2.1) to include a feed-forward path<sup>6</sup>.

To use the disturbance  $d(t)$  ( $|d(t)| < \infty$ ) explicitly in the control system, it has to be measurable separately from the control system's output  $y(t)$ . This can be only done practically if either all disturbance sources or the dominant ones can be identified and measured with some physical sensor. If the signal  $d(t)$  is obtainable, it can be fed into the signal path near the controller. In literature, two locations for the disturbance signal to be injected into are described. The method used decides, among other things, which (if any) filtering of  $d(t)$  has to be done.

One technique is to regard the disturbance to be similar to the error signal  $e(t)$ , i.e. the controller input. [Brosilow2002] In this case, the pre-processing, in general, has to include a scaling operation, because the units and magnitudes of  $e(t)$  and  $d(t)$  are different. If  $d(t)$  has non-zero mean, it needs to be subtracted to achieve linear behavior. This means, the needed disturbance filter is

$$D_1(s) = k_{\text{disturbance},1} D'(s), \quad \text{with } D'(s) = \mathcal{L} \left\{ d(t) - \mu_{d(t)} \right\}. \quad (5.30)$$

The second method is to add the filtered disturbance signal to the controller output. [Foellinger2016; electronics7100223] In general, this also requires the same pre-processing as before,

$$D_2(s) = k_{\text{disturbance},2} D'(s), \quad \text{with } D'(s) = \mathcal{L} \left\{ d(t) - \mu_{d(t)} \right\}, \quad (5.31)$$

but with a different scaling factor  $k_{\text{disturbance},2} \neq k_{\text{disturbance},1}$ . This method has the advantage of being potentially faster, because the controller does not occur in the signal path for the disturbance feed-forward, so its dynamics add no additional delay to  $d_2(t)$ .

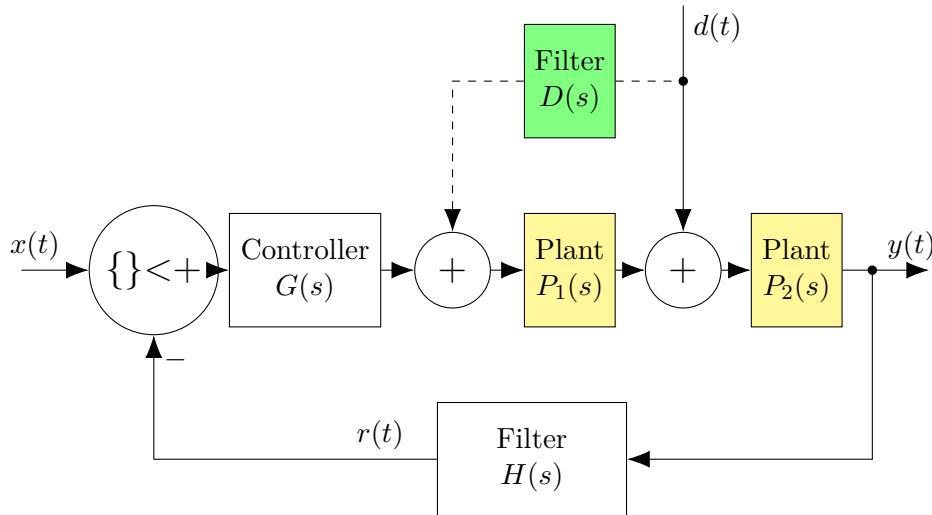
In both cases it is common to add another filter to  $D_{1,2}$ . This is often a lowpass filter  $H_d()$  to remove noise. The disturbance filter than becomes

$$D(s) = k_{\text{disturbance}} \cdot H_d(s). \quad (5.32)$$

The new controller architecture is shown in Figure 5.22. Drawn there is the plant  $P(s)$  being split into  $P_1(s)$  and  $P_2(s)$ . This is neither necessary nor possible under all circumstances, but feeding forward the disturbance that acts earlier on the plant would lessen the delay of  $d(t)$ <sup>7</sup>. Besides its main purpose of quickly reacting to a disturbance change, the feed-forward also has the inherent benefit of having no influence on the systems stability. Since  $d(t)$  is assumed to be finite, the output has to react to  $d(t)$  also with a finite response, as the filtered  $d(t)$  is only added to  $y(t)$ .

<sup>6</sup>The calculations here are done in continuous time to be comparable with the control system model in section 2.3

<sup>7</sup>A disturbance that acts on the middle of the plant is different from the one that acts on the end.



**Figure 5.22:** Schematic for a control system that uses disturbance feed-forward of the measurable disturbance  $d(t)$  in addition to the feed-back loop; changes to the classical control system architecture in Figure 2.1 highlighted in yellow(changed) and green(added) (based on [Foellinger2016])

When testing the newly improved system, there are two observations to be made. First, the stability over a few hours is not improved compared to the system that uses only feedback. But the stability also stays about the same when the control parameters are changed to achieve a slightly “less aggressive” controller. Decreasing  $k_p$  by a factor of 2 and  $k_i$  by a factor of 4 does not change stability (measured with the MSE over 1 h each) over the feedback-only case.

### 5.6.2 Feedback on Faraday-Cup Charge Measurements

In the previous part of this chapter, the control input and the measured output was always the cavity power  $P_{\text{cavity}} = P_{\text{cavity}}(t)$ . This power reading is measured with a RF power probe in the first half-cell of the electron gun.

But as the electrons are accelerated not only by the half-cell but also by the second and third full-cells, regarding only  $P_{\text{cavity}}(t)$  as a representative for the electron energy can be misleading. Also, the conversion from RF power to electron energy depends on the (time-varying) properties of the cavity, so there is no simple linear relation between them fixed in time.

An approach that tries to circumvent these issues uses the dark current. The dark current is normally unwanted, as it is the electron emission from the gun, that is not stimulated by the laser. Without a laser pulse, the electric field, generated by the RF inside the first half-cell, accelerates electrons from the vicinity of the cathode that are not freed by the UV laser pulse and the photo effect, but rather spontaneous by thermionic emission.

For this dark current, the relation

$$P_B = \frac{I \Delta W}{q}, \quad (5.33)$$

holds. It describes the relationship between the power transferred to the beam  $P_B$  by the cavity, the current of the beam  $I$  and the energy gain  $\Delta W$  (with the electron charge  $q$ ). [Wangler2008]

That means by measuring the beam current  $I(t)$  or the charge during one pulse

$$Q_B = \int_0^{T_{\text{pulse}}} I(t) dt = \int_0^{4.5 \mu\text{s}} I(t) dt, \quad (5.34)$$

it is possible to calculate at least  $P_B/\Delta W$ .

This charge reading can then be used for the control system input/output as an alternative to the cavity RF power.

To measure the charge  $Q_B$ , a Faraday cup is used. Faraday cups are hollow metal cups designed to catch charged particles, such as electrons, in vacuum. [radiabeamFaradayCups]

The Faraday cup does not discriminate between the origin of the electrons in the  $[0, T_{\text{pulse}}]$  interval. This is different from the measurements taken with the Integrating Current Transformer (ICT) and the charge output of one of the Beam Position Monitors (BPMs). They are only sensitive to the laser-generated electron beam. [Nasse2019]

The charge is captured as a current over some time. This charge then needs to be transformed into a voltage readable by the FLUTE data acquisition system, which takes voltages as an input. This is done with a Charge Sensitive Amplifier (CSA).

It is similar to an analog integrator (see Figure 5.23). In principle it can be constructed using an operational amplifier as depicted in Figure 5.23. [Horowitz2015]

By using the fundamental rules of the ideal model of an operational amplifier, that is the voltage difference between the + and – inputs vanishes and the input currents flowing in the + and – inputs are zero, the input impedance is

$$Z_{\text{in}} = \frac{V_{\text{in}}}{I_{\text{in}}} = R. \quad (5.35)$$

The transfer function in the Laplace space can be directly stated using Kirchhoff's voltage law<sup>8</sup> and the complex impedance of a capacitor  $Z_C = \frac{1}{sC}$  (with  $s = j\omega$ ) as

$$\frac{V_{\text{out}}(s)}{V_{\text{in}}(s)} = -\frac{R + \frac{1}{sC}}{R}. \quad (5.36)$$

Substituting  $I_{\text{in}}(s) = \frac{V_{\text{in}}(s)}{Z_{\text{in}}} = \frac{V_{\text{in}}(s)}{R}$  yields

$$\frac{V_{\text{out}}(s)}{I_{\text{in}}(s)} = -\left(R + \frac{1}{sC}\right). \quad (5.37)$$

$$\frac{V_{\text{out}}(s)}{I_{\text{in}}(s)} = -\frac{1}{sC}. \quad (5.38)$$

With  $\mathcal{L} \left\{ \int_0^t x(\tau) d\tau \right\} = \frac{1}{s} X(s)$  (see [Leon2015]), the output in the time domain becomes

$$V_{\text{out}}(t) = \mathcal{L}^{-1} \left\{ -I_{\text{in}}(s) \frac{1}{sC} \right\} \quad (5.39)$$

$$= -\frac{1}{C} \frac{1}{s} I_{\text{in}}(s) \quad (5.40)$$

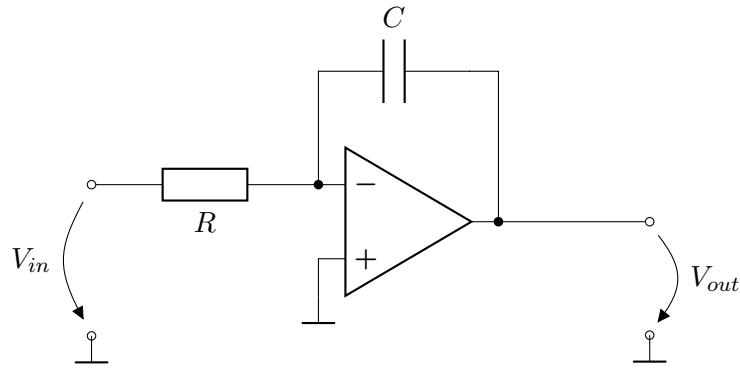
$$= -\frac{1}{C} \int_0^t I_{\text{in}}(\tau) d\tau. \quad (5.41)$$

The CSA uses an adjustable capacity (to set the range of expected charges) and an analog switch to clear the accumulated charge on the capacitor.

At the moment, the Faraday cup is mounted at the end of the low energy section. The capacity in the CSA is resetted at the start of every pulse and outputs a voltage proportional

---

<sup>8</sup>The sum of all signed voltages around a closed loop is zero.

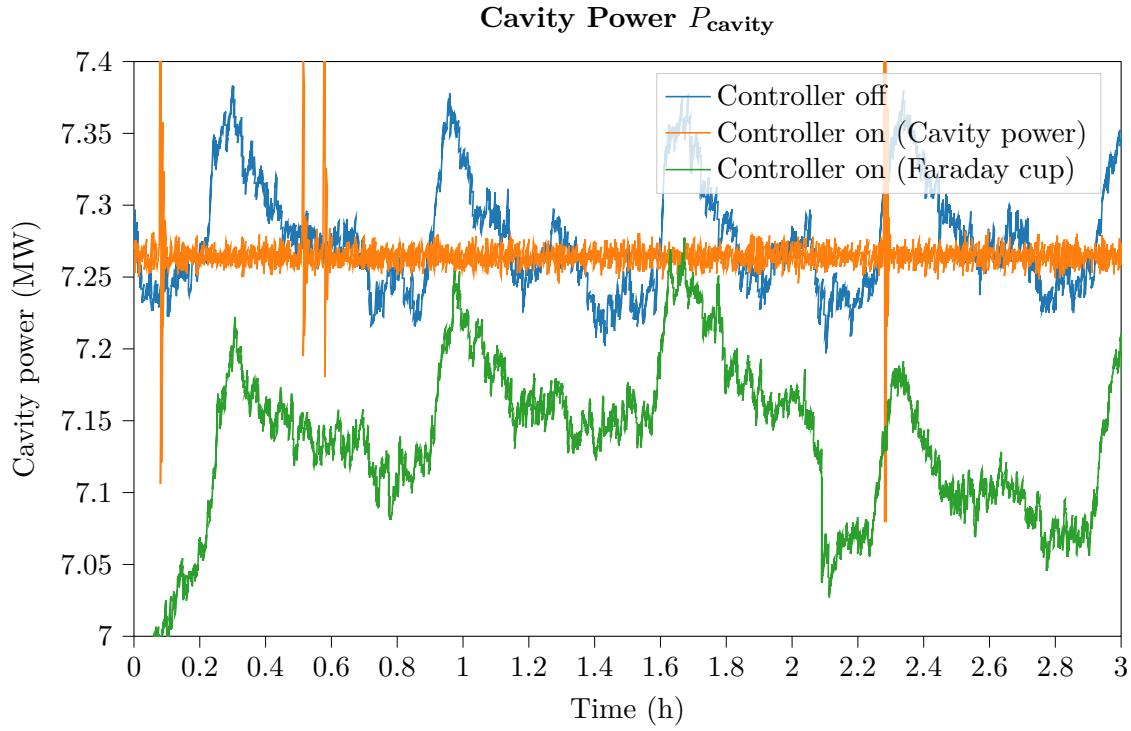


**Figure 5.23:** Schematic of an inverting integrator using an integrated operational amplifier

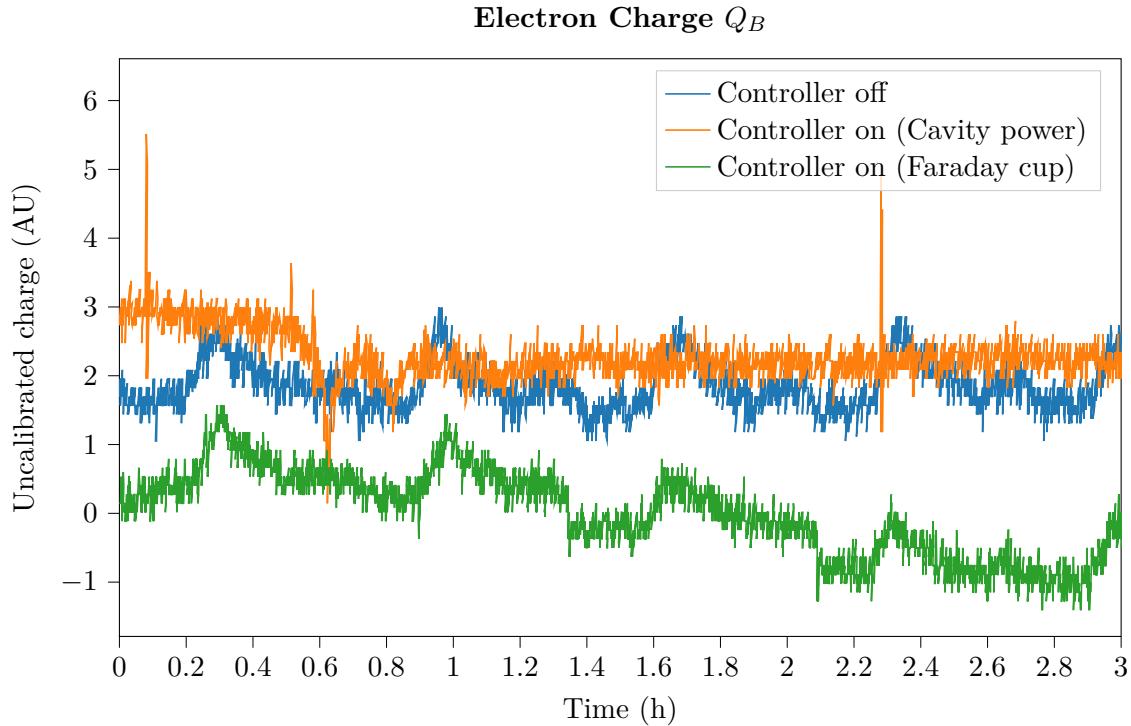
to the range setting and the measured charge. This voltage is measured with a data acquisition system connected to EPICS and thus available in CSS (compare section 4.1).

To set the range of the CSA, its serial interface together with a serial to Ethernet adapter is used (See an example of the command structure in Listing 1).

Figure 5.24 and Figure 5.25 show the cavity power and the (un-calibrated) charge reading of the Faraday cup for three different cases: the control system switched off, the control system on with the cavity power used as control input and the control system with the Faraday cup's readout (via the CSA) as an input.



**Figure 5.24:** Cavity power without control interaction, with the controller acting on the cavity power and with the controller acting on charge; Note the three curves are measured at different times and have no relation to one another



**Figure 5.25:** Electron charge without control interaction, with the controller acting on the cavity power and with the controller acting on charge; Note the three curves are measured at different times and have no relation to one another

## 6. Summary and Outlook

For the operational phase of Ferninfrarot Linac- und Test-Experiment (FLUTE), a stable electron energy enabling stable terahertz (THz) energies will be vital to perform scientific experiments.

At the start of the thesis, the stability of the electron gun was unsatisfactory. As the main source of the instability, the cooling system of the electron gun's body was identified.

This thesis shows that with a control system that interacts with the low power input signal of the klystron, it is possible to improve the stability by creating a closed-loop feedback system using readouts from the Experimental Physics and Industrial Control System (EPICS) control system as sensors and a controllable Radio Frequency (RF) attenuator in the signal path as actuator. The system transfer function (plant) between the attenuator and the power in the electron gun cavity shows  $PT_2$  behavior, so control with a Proportional Integral Derivative (PID) controller is possible and its positive effects on the stability are shown.

However the studies here also show the limitations of a traditional Linear Time Invariant (LTI) control system. The controller should be designed with a fast response time and sufficient integral gain to reject disturbances quickly. This however leads to small safety margins in the gain and phase response, which makes it likely for the control system to become unstable if system parameters change slightly. Also, the parameterization of the necessary measurement filter is a trade-off between stability of the control system and measurement noise rejection. The control system could be further optimized by switching to a totally different controller architecture. A possible choice might be model predictive control, which exploits the already identified plant transfer function but determines the output through an optimization process rather than a linear system.

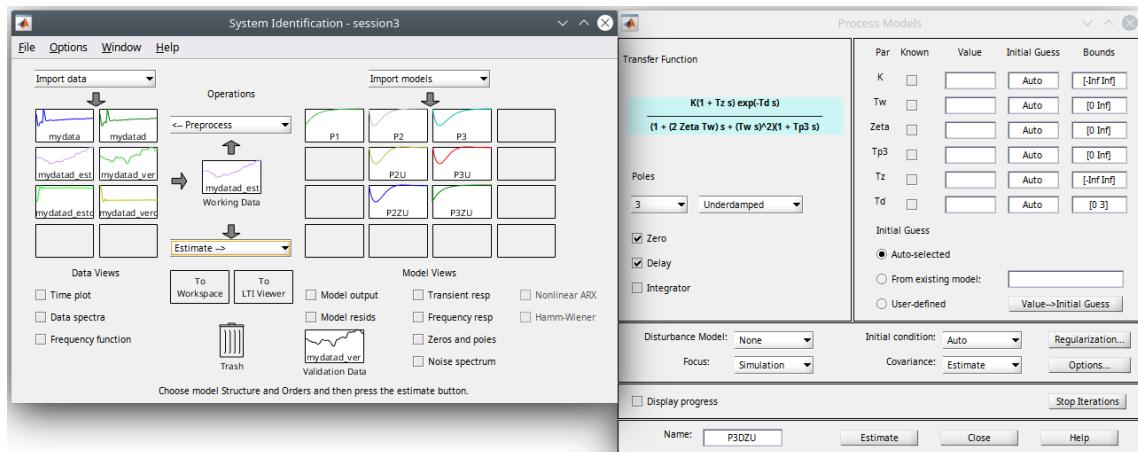
At the moment of writing (June 2021), the Linear Accelerator (LINAC) section is under commission, so FLUTE is shut down for several weeks. With the LINAC in place and the low energy section operational again, measurements of the electron energies before and after the LINAC can be performed. The method currently used is based on the electron bunch, which is steered with an electromagnet by the Lorentz force (see Equation 2.9). By adjusting the electromagnet's coil current in such a way that the electron beam hits a camera screen in the center, the electron energy can be calculated from the coil current. With this destructive and slow process it will be possible to show if the stabilization of the cavity RF power actually improves the energies of the accelerated electrons and if the approach using the Faraday cup is an improvement over the cavity RF power solution.

In the near future, both the electron gun and the klystron are to be upgraded to new versions. Switching to another gun and/or a different RF system could lead to an entirely different behavior of the whole system, but the techniques described in this thesis are universally applicable and the control system can easily be modified for usage with different components.

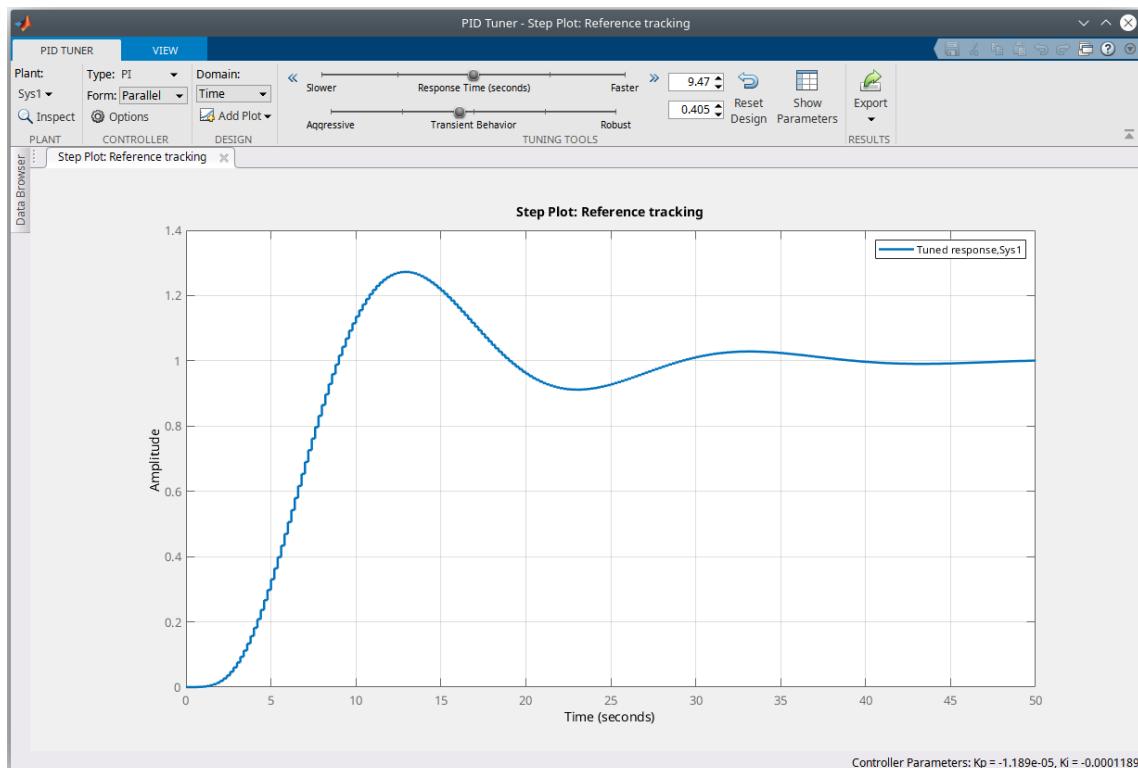


# Appendix

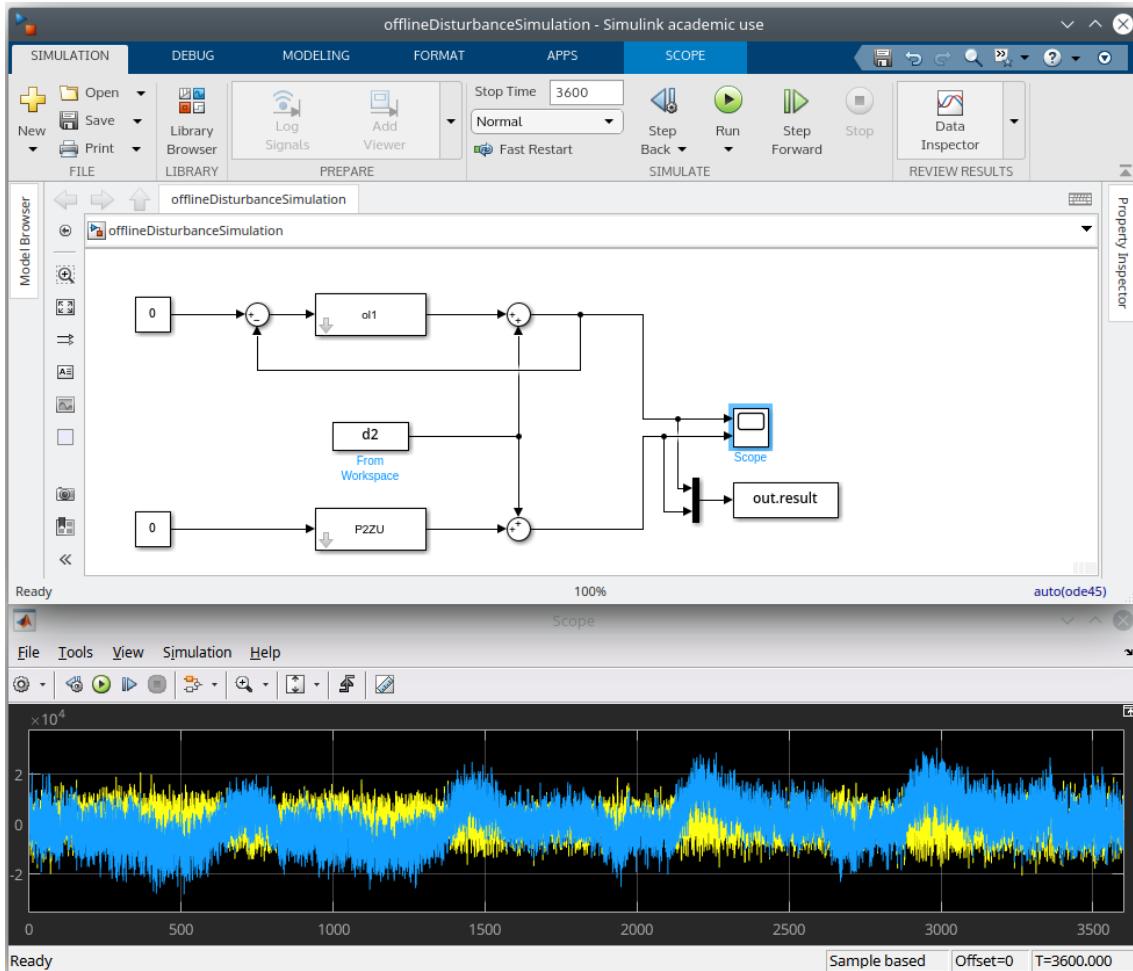
## A Complementary Material Controller Design



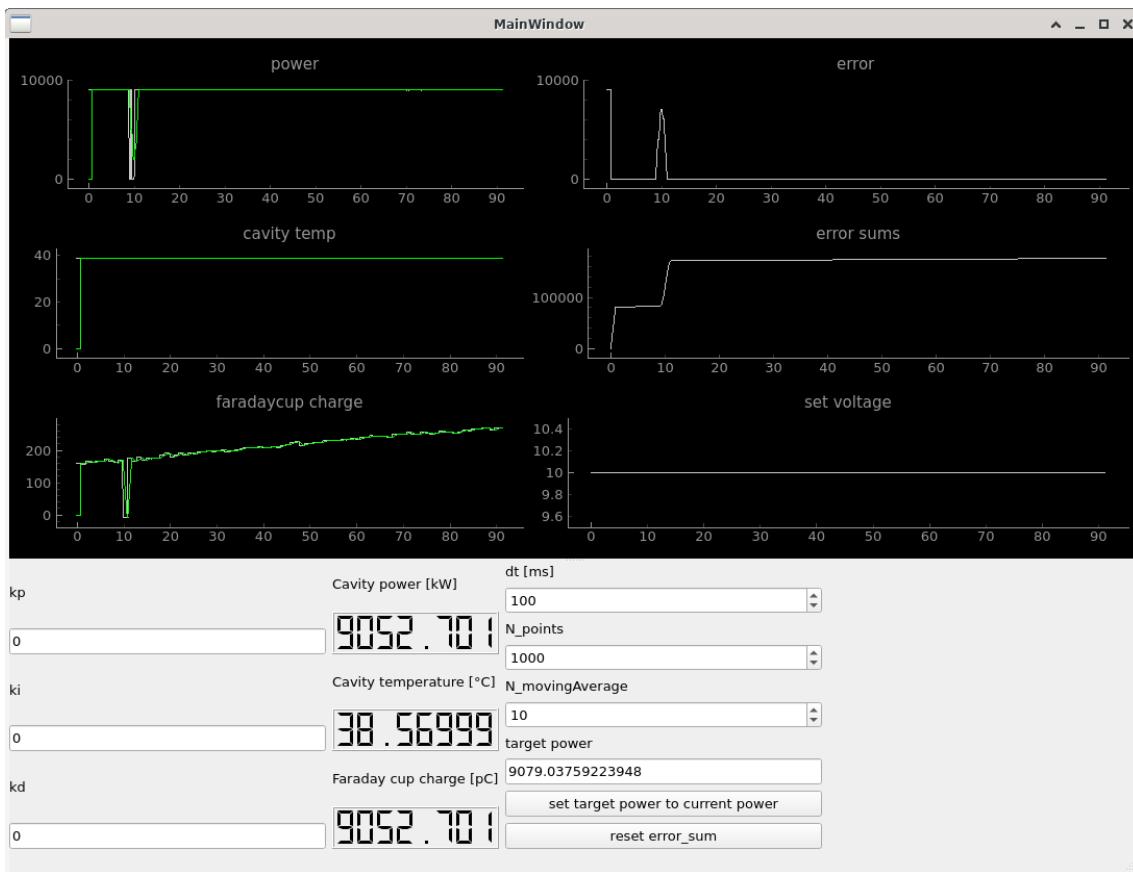
**Figure A.1:** Screenshot of the Matlab System Identification Toolbox; to the right the process models estimator window



**Figure A.2:** Screenshot of the Matlab PID Tuner from the Control Systems Toolbox



**Figure A.3:** Simulink model to evaluate the designed controller together with the measurement filter (`ol1`) compared to the uncontrolled system (in `P2ZU`) using measured disturbance data (in the vector `d2`); below a view of the scope data



**Figure A.4:** Screenshot of the control system's GUI application

**Listing 1:** Java class of the PCB421A25 charge amplifier demonstrating the command structure and checksum calculation for integration of the amplifier into Control System Studio (CSS)

---

```

1  class PCB421A25 {
2      final static char STX = '\u0002';
3      static enum FixedRange {
4          RANGE_1000000("01"),
5          RANGE_500000("02"),
6          RANGE_200000("03"),
7          RANGE_100000("04"),
8          RANGE_50000("05"),
9          RANGE_20000("06"),
10         RANGE_10000("07"),
11         RANGE_5000("08"),
12         RANGE_2000("09"),
13         RANGE_1000("10"),
14         RANGE_500("11"),
15         RANGE_200("12"),
16         RANGE_100("13");
17     public final String command;
18     private FixedRange(String command) {
19         this.command = command;
20     }
21 }
22
23 public PCB421A25() {};
24
25 public void setFixedRange(FixedRange fixedRange) {
26     String command=STX+"c"+fixedRange.command;
27     command += calculateChecksum(command);
28     sendCommand(command);
29 }
30
31 public boolean setVariableRange(int variableRange) {
32     if (!(variableRange>100 && variableRange<1000000)) return false;
33     String command=STX+"d"+String.format("%06d", variableRange);
34     command += calculateChecksum(command);
35     sendCommand(command);
36     return true;
37 }
38
39 private char calculateChecksum(String command) {
40     int checksum=0;
41     for(int i=0;i<command.length();i++)
42         checksum+=(int)command.charAt(i);
43     String checksum_hexstr=Integer.toHexString(checksum).toUpperCase();
44     return checksum_hexstr.charAt(checksum_hexstr.length()-1);
45 }
46
47 private void sendCommand(String command){
48     System.out.println("Command to send: "+command+" (length: "+command.length()+"");
49     // [...]
50 }
51
52 public static void main(String[] args) {
53     PCB421A25 chargeSensitiveAmplifier = new PCB421A25();
54
55     //Test fixed ranges
56     System.out.print("Fixed range 1000000;\t");
57     chargeSensitiveAmplifier.setFixedRange(FixedRange.RANGE_1000000);
58     System.out.print("Fixed range 100;\t");
59     chargeSensitiveAmplifier.setFixedRange(FixedRange.RANGE_100);
60
61     //Test variable ranges

```

```
62     System.out.print("Variable range 123456;\t");
63     chargeSensitiveAmplifier.setVariableRange(23500);
64     System.out.print("Variable range 999;\t");
65     chargeSensitiveAmplifier.setVariableRange(999);
66 }
67 }
```

---

## B Lab Test and Measurement Equipment

### B.1 Benchtop multimeters

#### B.1.1 Agilent 34411A

**Table B.1:** Agilent 34411A - Specifications

Specification	Value
Digits	6 1/2
Measurement method	cont integrating multi-slope IV A/D converter
Accuracy (10 V range, 24 hours)	0.0015 %+0.0004 % (% of reading + % of range)
Bandwidth	15 kHz (typ.)

**Table B.2:** Agilent 34411A - Relevant SCPI commands

Description	Example command	Example return
Read current measurement	READ?	+2.84829881E+00 (2.848 V)

#### B.1.2 Keysight 34470A

**Table B.3:** Keysight 34470A - Specifications

Specification	Value
Digits	7 1/2
Measurement method	cont integrating multi-slope IV A/D converter
Accuracy (10 V range, 24 hours)	0.0008 %+0.0002 % (% of reading + % of range)
Bandwidth (10 V range)	15 kHz (typ.)

**Table B.4:** Keysight 34470A - Relevant SCPI commands

Description	Example command	Example return
Read current measurement	READ?	+9.99710196E+00 (9.997 V)

## B.2 Data Acquisition/Switch Unit

### B.2.1 Keysight 34972A

**Table B.5:** Keysight 34972A - Specifications

Specification	Value
	34907A (Multifunction module)
DAC range	$\pm 12\text{ V}$
DAC resolution	16 bit ( $2^4\text{ V}/2^{16} = 366.21\text{ }\mu\text{V}$ per bit)
DAC maximum current	10 mA
	34901A (20 channel multiplexer)

**Table B.6:** Keysight 34972A - Relevant SCPI commands

Description	Example command	Example return
Read current measurement	READ?	+2.0020000E+01 (20.02 °C)
Set DAC voltage of ch 204 to 3.1 V	SOUR:VOLT 3.1, (@204)	

### B.3 RF signal generator

#### B.3.1 Rohde and Schwarz SMC100A

**Table B.7:** Rohde and Schwarz SMC100A - Specifications

Specification	Value
Frequency range	9 kHz to 3.2 GHz
Maximum power level	17 dBm
SSB phase noise (@ 1 GHz, $f_o = 20$ kHz, $BW = 1$ Hz)	-111 dBc
Level error	<0.9 dB

**Table B.8:** Rohde and Schwarz SMC100A - Relevant SCPI commands

Description	Example command	Example return
Set RF power level to 10.5 dBm	SOUR:POW 10.5	
Set RF frequency to 3.1 GHz	SOUR:FREQ:FIX 3.1e9	
Enable the RF output	OUTP on	

### B.4 RF power meter

#### B.4.1 HP E4419B

**Table B.9:** HP E4419B - Specifications

Specification	Value
Digits	4
Accuracy (abs. without power sensor) Power probe: E4412A	±0.02 dB
Frequency range	10 MHz to 18 GHz
Power range	-70 dBm to 20 dBm

**Table B.10:** HP E4419B - Relevant SCPI commands

Description	Example command	Example return
Measure power on input 1	MEAS1?	+2.89435802E+000 (2.894 dBm)

## B.5 Vector Network Analyzer

### B.5.1 Agilent E5071C

**Table B.11:** Agilent E5071C - Specifications

Specification	Value
Frequency range	9 kHz to 8.5 GHz

## B.6 Phase noise analyzer

### B.6.1 Holzworth HA7062C

**Table B.12:** Holzworth HA7062C - Specifications

Specification	Value
DUT input frequency	10 MHz to 6 GHz
Measurement bandwidth	0.1 Hz to 40 MHz offsets

# Acknowledgments

First I would like to thank Prof. Anke-Susanne Müller and the whole FLUTE team for providing me with the opportunity to write my thesis at their research facility. FLUTE proved to be a one-of-a-kind machine and I really appreciate the work that has already been put into it.

I am very grateful to Prof. Dr.-Ing. John Jelonnek from the Institute for Pulsed Power and Microwave Technology for taking the responsibility of being my first reviewer. Also many thanks to his PhD students Benjamin Ell and Alexander Marek for listening patiently to my presentations and providing great support in optimizing them.

I'm especially in debt to my excellent advisor Dr. Nigel John Smale, for always having the time to go over both fundamentals and nifty details of electronics and physics alike. Without his support I wouldn't even be able to switch on FLUTE by now and much less a successful thesis would've been possible.

Valuable inputs from Andreas Böhm regarding the RF attenuator and electronics in general were a great help.

Thanks to Igor Križnar for constructive discussions and programming the graphical user interface and its back-end to control the charge sensitive amplifier.

Construction of the charge sensitive amplifiers connection box by Jürgen Schmid is greatly appreciated.

Thanks to Olena Manzhura from KIT-IPE for her support and for prove-reading the thesis multiple times.

Finally but maybe most importantly, a big thanks goes out to my family, that is my mom Sonja, Stefan and my grandparents Ernst and Gretel. I owe them my deepest appreciation for their support over the last years.

