GEORG-AUGUST-UNIVERSITÄT

GÖTTINGEN

II. Physikalisches Institut

Studies to Optimize the Accelerating Gradient of the Superconducting TESLA Cavities for the International Linear Collider

by

Marc Wenskat

Dieser Forschungsbericht wurde als Diplomarbeit von der Fakultät für Physik der Georg-August-Universität zu Göttingen angenommen.

Angenommen am:	31. März 2010
Referent:	Prof. A. Quadt
Koreferent:	Prof. E. Elsen

Ein einheitlicher Wille ist eine uneinnehmbare Festung.
Chinesisches Sprichwort

Contents

1 Introduction **1**

2 Theory of superconductivity and cavities **4**
2.1 Superconductivity 4
2.1.1 BCS theory 4
2.1.2 Ways to loose the superconductivity (quench and field emission) . 8
2.2 Toy model - the pillbox cavity 13

3 Control theory applied to superconducting cavities **27**
3.1 Digital signal processing 27
3.1.1 Basic definitions 28
3.1.2 Control theory 29
3.1.3 Vector sum 37
3.1.4 IQ-sampling 38
3.1.5 Digitization and preprocessing 40
3.2 Accelerator as a system 44
3.2.1 What we want to achieve and what we can control 44
3.2.2 Noise sources and disturbances 48
3.3 Cavity behavior 52
3.3.1 Description of a cavity via ODE 52
3.3.2 Description of a pulsed cavity via state space representation 54
3.4 Driving multiple cavities with a single klystron 60
3.4.1 Without Beam 64
3.4.2 With Beam 65
3.4.3 Quench detection 71

4 Implementation of LLRF control at FLASH **72**
4.1 Requirements at FLASH 74
4.2 Feedback 74
4.2.1 Principle layout of LLRF control 74
4.2.2 Open loop, no beam operation 77

4.2.3 Closed loop, no beam . . . 78
4.2.4 Closed loop, with beam . . . 79

5 Implementation of LLRF control at the vertical test stand 80
5.1 Analog test stand . . . 82
5.2 Digital test stand . . . 84
5.2.1 Principle signal path . . . 84
5.2.2 Controller board and algorithm . . . 85
5.2.3 RF box . . . 86
5.2.4 GUI and communication . . . 87
5.2.5 Results of the digital test stand . . . 90
5.2.6 Problems . . . 91
5.2.7 Possible further developments . . . 93

6 9 mA - run at FLASH 95
6.1 General interest and goals . . . 95
6.2 Data analysis - Sep. 09 run . . . 96
6.2.1 Gradient disparities . . . 96
6.2.2 Quench detection . . . 102

7 Summary & Outlook 105
7.1 Discussion of the digital vertical test stand . . . 105
7.2 Discussion of the 9 mA run . . . 106

8 Appendix 108
8.1 sweeptest2.m . . . 108
8.2 QLnoBeam.m . . . 136
8.3 qps.m . . . 139
8.4 ubertragungsfunktion.m . . . 142
8.5 Characteristic curves of test stand hardware . . . 145
8.6 Fouriertransformation of reflected power to identify the noise . . . 150

Bibliography 150

List of Figures

1.1 Sketch of FLASH before Shutdown October '09, [17]. The Number inside the accelerating modules denotes the DESY internal production description. 2

1.2 (1) The electron bunch enters the undulator and starts with spontaneous emission. (2) The emitted radiation reacts back onto the bunch and leads to a density modulation. The bunch gets 'sliced' where the intra-bunch distance of each slice is about the wavelength of the amplified radiation. This results in coherent emission (3). The effect saturates after a certain undulator distance and the electron beam is transported to the beam dump [43]. 3

2.1 Because of phonon interactions between electrons and the atomic lattice, electrons can create stable bound systems and act like bosons [1]. 4

2.2 Phase space with three parameters: the current density j, the external magnetic field H and the temperature of the system T [1]. 5

2.3 Phase diagrams for type I (left) and type II (right) superconductor. . . . 6

2.4 In superconducting materials, external magnetic fields can intrude into the material, but only in a thin layer [24]. 7

2.5 Example of different mechanisms in a cavity which can cause the loss of superconductivity [22]. 9

2.6 Left: electric field strength inside a TESLA cavity, Right: magnetic field strength inside a cavity. High fields are marked with red [25]. 10

2.7 Principal mechanism of a quench [13]. 10

2.8 The mechanism of the Fowler-Nordheim-Emission. Left: the metal surface in vacuum without an external electric field. The potential Barrier has infinite thickness and no electron will tunnel into the vacuum. Right: an external electric field is applied, which decrease the potential barrier and electrons can leave the metal surface [13]. 11

2.9 Some examples of the inner cavity surface which will lead to a quench or field emission [23]. 12

2.10 Example of Temperature Mapping of a superconducting cavity with an impurity on its surface. This impurity leads to a local quench [13]. 12
2.11 a) shows the geometry of a pillbox cavity. b) depicts the J_0 and J_1 Bessel function which describes the radial electric field or the radial magnetic field of the cavity, respectively. 14
2.12 This picture shows the electric (a) and magnetic (b) field distribution inside a pillbox cavity of the lowest Eigenmode, the so called TM_{010} mode. 15
2.13 Different modes have different field contributions, like a deflecting electric or magnetic field along the axis (left: electric field / right: magnetic field). 17
2.14 Since a single cavity will have a single resonance frequency, N coupled cavities will have N resonance frequencies. To describe such a system, Thevenin's Theorem is used. This theorem allows us to draw an equivalent circuit of a coupled system and to describe it in terms of electric circuits [42]. 20
2.15 In this picture, the passband of a single TESLA cavity is shown. The typical bandwidth is 30 MHz with a difference between the π-mode and the 8/9-π-mode of roughly 800 kHz [11]. 21
2.16 Equivalent circuit of a 4-cell cavity. 21
2.17 The eigenvalues of the electric field strength, represented as length of the arrow in each cell, for different modes. In the π-mode, the lowest case here, the distribution is nearly equal in each cell but with a phase shift of π per cell 24

3.1 The concept of impulse response [21]. 30
3.2 The representations in the frequency or time domain for continuous signals are linked via the Laplace-transformation, which can be considered as a more general Fourier-Transformation. For discrete time systems you have to use the so called z-Transformation [21]. 32
3.3 Controller and plant in closed loop operation [21]. 33
3.4 Depending on the poles and zeros of your transfer function, you can identify regions of stability and instability [21]. 36
3.5 Driving multiple cavities with one klystron will make it impossible to regulate individual signal. This situation requires the control of the so called vector sum. 37
3.6 In complex representation and do a coordinate transformation we use the real and imaginary part, the so called I- and Q-Part, instead of the Amplitude and Phase of a signal. [36] 38

3.7 Principle of the sampling scheme at FLASH for IQ control [36]. Left: the phaseshift is shown which need to be compensated by the loop-rotation-matrix. Right: The sampling frequency is 4 times the sampled signal frequency. This guaranties a sampling scheme of Q, I, -Q, -I and will bring every information we need. 39

3.8 Samples are taken from the analog input signal (A). The time discretization is done with the sampling frequency f_s. The voltage is stored in a sample-and-hold device (B) which can be a simple capacitor. Finally the voltage across the capacitor is converted into a digital number (C), usually represented by n bits of digital logic signals [21]. 40

3.9 The principle of an ideal mixer is shown in this graph. The original signal with frequency f_{RF} is mixed with a local oscillator signal with a frequency f_{LO}. This results in an intermediate signal with frequency f_{IF}. 41

3.10 The upper plot shows a down-converter, the lower plot an up-converter. Both signals are mixed with an ideal mixer [36]. 42

3.11 A real mixer will have sidebands, due to non-linearities. These sidebands will be at higher/lower harmonics of the difference and sum of the input frequencies [36]. 42

3.12 (a) The ratio of sampled signal frequency and sampling frequency is 0.09 and 0.31, respectively (b). This results in a proper sampling and the signal will be sampled correctly. (c) The ratio is 0.95 which means, that the sampling frequency is nearly equal to the sampled frequency. Therefore, a shift in the frequency will occur which is called aliasing [21]. 43

3.13 General layout of amplitude feedback of an accelerator (compare Fig.3.3). 44

3.14 Description of the parameters we need to concern. The phase of the bunch Φ_b, the phase φ and amplitude U of the accelerating field [11]. 45

3.15 Time structure at FLASH. The filling time is about 500 ms. After this, the steady state of the cavity will be reached and the beam is injected. The beam pulse length will vary, but the flattop has a length of 800 ms. Then the rf signal is turned off and the field inside the cavity will decay. The pulse structure can have a different amount of bunches, charge per bunch and pulse repetition time [3]. 45

3.16 The filling time is controlled by the Q_L which in turn can be controlled by the coupling of the driving signal to the cavity. The input power coupling and the phase of the driving signal can be controlled via 3-stub-tuner, motorized antenna or phase-shifter. 46

3.17 In this plot, different bandwidths for different couplings and the phase change is shown ($Q_L \propto \beta$, β is not the coupling but the damping in this case) [21]. 47
3.18 Different couplings lead to different reflected waves at the end of the transmission line. 47
3.19 HOMs can arise during operation due to beam - cavity interaction. This will lead to a different field pattern which will be seen by the subsequent bunch [21]. 48
3.20 This plot shows that the maximum frequency detuning, caused by microphonics, even influences the Q_L. To assure that the cavity will stay on resonance with the driving signal in presence of the microphonics, the bandwidth needs to be broadened [35]. Otherwise the klystron power needs to be increased. 49
3.21 Lorentz force will create problems matching the driving signal to the cavity conditions during filling time [11]. 50
3.22 Beam Loading occurs because the bunch traveling through the cavity generates an electric field which will decrease the cavity field (the factor by which the gradient decreases depends on the bunch charge, cavity bandwidth and the phase difference of bunch phase and gradient phase) [40]. 51
3.23 The Klystron - cavity System can be seen as a electric circuit with specific loads, inductance and resistance. The generator current and the beam current need to be matched through the system [14]. 53
3.24 a) H_{11}, cavity on resonance. b) detuned H_{11}. c) detuned H_{21}. Only in a detuned cavity, H_{21} is non-zero. Bandwidth is 215 Hz for both plots, detuning by 100 Hz. 57
3.25 All passband modes, G=1, no loop delay. Nyquist-stability-criterion tells us, that if the amplitude at a certain frequency is higher than 0 dB at a phase shift of 180 degrees, the system can become unstable. This plot shows, that this could happen if we do not control the $\frac{8}{9}\pi$-mode. 58
3.26 Left: no loop delay, right: delay is 2π. Strong oscillations on the Amplitude can be seen, which is the influence of the $\frac{8}{9}\pi$-mode. Therefore, considering loop-delay when planning your digital control is necessary. (see Fig. 3.27) 59
3.27 Stability chart for a single cavity. Simulation of a digital control loop. f_s =1 MHz, β_π- and $\beta_{\frac{8}{9}\pi}$-mode, $f_\pi - f_{\frac{8}{9}\pi}$ = 800 kHz [37]. 59
3.28 This picture shows the gradient spread at FLASH before the upgrade. . . 60

3.29 The accelerating modules 4,5,6 are fed by a single, two armed klystron. The distribution of the RF power is shown here. 61
3.30 Lowering the gradient of a cavity by more than 15%, more klystron power is needed to drive this cavity [5]. 62
3.31 Adjusting individual Q_L, no quench in the presence of the beam will happen but different time constants will occur [5]. 62
3.32 Without beam, a change in Q_L leads to different slopes during flattop so that certain cavities will quench during flattop [5]. 63
3.33 First Case: Generator current is too low to match beam load. Second Case: Generator current is too high to match beam load. Third Case: Generator current fits beam load [5]. 66
3.34 Individual cavity quenching limit below or above vector sum gradient [5]. 67
3.35 The critical vector sum gradient: cavities operating below V_S quench exactly at t_0; cavities operating above V_S quench at t_1 [5]. 68
3.36 Plot of the values calculated for the current distribution parameters with the derived formulas. Blue dotted line is α_i^{min} [5]. 69
3.37 By using the MATLAB cavity Simulator, you can solve this problem numerically and derive the maximum vector sum for a given gradient spread [5]. 70
3.38 This plot shows the time development of a quench. In (a) the transmitted power is shown and in (c) the reflected power. (b) shows the driving signal and (d) the Q_L against the pulse number (cavity no. 2 at ACC6) pulse one is shown in red, the second in black and the third in blue [33]. 71

4.1 Sketch of GDR Model. 72
4.2 Sketch of SEL Mode. 73
4.3 Principle layout of the closed loop system which needs to be controlled [29]. 75
4.4 Layout of the LLRF Control of the ACC1 module at FLASH [30]. 76
4.5 Functional block diagram of the LLRF control system for one cavity [31]. 76
4.6 Algebraic model of the feedback and feedforward system [31]. 77
4.7 Open loop, no beam operation of ACC1. Even with no flat gradient for individual cavities a flat vector sum of the cavities can be achieved [14]. . 78
4.8 With closed feedback loop a higher vector sum gradient can be achieved, the cavity detuning is compensated and the gradients of individual cavities will not drift [14]. 79

5.1 This plot sketches different types of Q-E-Plots and what informations we can derive about the mechanisms inside a cavity [18]. 80
5.2 Engineering drawing of the vertical test stand at DESY [43]. 81

5.3 Layout of the signal path at the current testing system [43]. 82
5.4 Relevant points of the reflected Power. 83
5.5 Layout of the digital testing system. 84
5.6 Picture of the used SimCon Board (courtesy of S. Simrock). 85
5.7 Technical drawing of the rf box used for generating the required frequencies. A picture of the hardware is shown in the section 8.5. 87
5.8 Drawing of the communication of MATLAB with the FPGA via TCP/IP and/or VME Bus [26]. 88
5.9 Latest version of the graphical user interface developed in MATLAB. The GUI is used to control the generator, spectrum analyzer and the FPGA and to monitor every needed value of the test stand. It is also used to plot every needed graph and calculates the needed and searched parameters (MATLAB Code in Section 8.1). 89
5.10 Cavity Z109. Left: Q-E-plot derived with the digital system. Right: Q-E-plot derived with the analog system. 90
5.11 Cavity Z138 with HOM couplers. Left: Q-E-plot derived with the digital system. Right: Q-E-plot derived with the analog system. 90
5.12 Oscillations in the reflected power lead to a huge error in the coupling (see Fig. 5.4). 92
5.13 The quench detection algorithm used at the vertical test stand. 94

6.1 Comparison of XFEL, ILC, FLASH and FLASH 9 mA. 95
6.2 RF distribution of Klystron 4 for ACC5 and ACC 6. 96
6.3 Left: Simulation of the cavity behavior with given parameters from the model. Right: real-time LLRF data from 9 mA - no beam run (ACC 6). The fact, that two cavities in the left plot show a positive tilt and not on the right is due to detuning, which was compensated by a resonance frequency shift [9]. 97
6.4 Comparison of simulation and FLASH data for rf distribution to reach high V_S and no quench with beam [34]. 98
6.5 a) ACC5 - DAQ data from the 9 mA run. b) is a simulation of the cavity behavior with given parameters from the model - no beam run. 98
6.6 Vector sum for all modules plotted against 422 pulses each with 9 mA. . . 99
6.7 Vector sum deviation for module 456 plotted against 422 pulses each with 9 mA. The slope is due to a phase drift in the machine. 100
6.8 Vector sum deviation over longer period of the 9 mA run (each point: 3 mA, 800μs bunch trains) [7]. 100

6.9 Correlation of the beam current to the accelerating gradient. Different working points of the machine correspond with the different linear relation [32]. . . . 101
6.10 In both rows, the red dotted line is always the first graph as reference. The lower row shows the Q_L calculation after rf is turned ($t = 0$ is the point where the rf is turned off after 400μs). A single point in the lower row is a fit of 20μs duration. The dropping of the Q_L by a factor 3 during a quench can be seen [33]. . . . 102
6.11 a) forward power, b) signal at the pick up antenna. c) detuning and d) Q_L during flattop. . . . 103

7.1 Main goal achieved: 2400 Bunche at 3 MHz, each 3nC. [7] . . . 106

8.1 output - input relation of the vectormodulator. nonlinear regime / saturation is reached. . . . 145
8.2 ADC-signal vs. powerinput. . . . 146
8.3 ADC-signal vs. powerinput. . . . 147
8.4 ADC-signal vs. powerinput. . . . 148
8.5 Picture of the RF Box, taken from above. . . . 149
8.6 FFT of reflected power, sidebands generated by downconverter can be seen at IF+ [9, 18, 36, 63] MHz. . . . 150
8.7 FFT of reflected power, sidebands filtered out by low-pass. oscillations still seen in reflected power. . . . 151

1 Introduction

The LHC will bring the new physics of this millennium. But still, the discoveries which will be done at the LHC need further investigations, which need to be carried out by a linear collider. This linear collider is an international project (ILC), where the collaboration partners are spread out all around the world. A huge effort is carried out to realize a machine, powerful enough to reach the high energy electrons in a most effective way. To reach these energies, thousands of accelerating structures, so called cavities, are needed and controlled. Several cavities are grouped in an accelerator module which is driven by a single klystron. The beam will be accelerated over a lenght of ≈ 22 km. This means, that LLRF (low level radio frequency - the system which controlls an accelerator) needs to ensure beam quality, such as density, energy- and momentum spread, emittance and more for a huge distance at a high power level and for multiple modules with an internal structure. The cavities used for the ILC will be standing wave cavities, which means that the electric field will need to remain constant over the time the beam travels through the cavity. This is a different concept if you compare this to traveling wave cavities (TW-cavities). In TW-cavities, the whole field energy, represented by a wave traveling along the cavity, is absorbed by the pulse which will be accelerated. This is an effective way to accelerate pulses in case the pulses are short compared to the field decay time. If the pulse becomes longer, the intra pulse energy deviation will become bigger since the accelerating field will decay over time. To reach long pulses with small energy deviation, standing wave cavities are needed. Unfortunately, due to heating by ohmic losses, a constant high gradient applied to the metallic surface leads to high thermal dissipation into the walls which need to be compensated by expensive cryogenics. To avoid these high losses, superconductivity offers a huge possibility to reach these high gradients with a lower invest in cryogenics. Besides this, superconducting cavities are shorter than comparable normal conducting ones, which causes less disruptive effects onto the beam and allows a shorter hence a cheaper linac. Furthermore, the special geometry possible with superconducting cavities allows for larger beam holes. This enables smaller emittance - respectively higher luminosity. The arguments pointed out above are the reason why such a huge effort is done to realize superconducting standing wave cavities.

The former TESLA Test Facility (TTF) at DESY is a facility to test such cavities and proof the feasibility of a linear accelerator reaching an energy of 500 GeV. In between, TTF is known as FLASH, the Free Electron Laser at Hamburg. After TTF made huge steps towards a superconducting linear collider, FLASH still contributes to the ILC developing process with some unique possibilities. A schematic overview of FLASH is given in Fig.1.1. FLASH is a facility providing ultra short electron bunches to generate laser

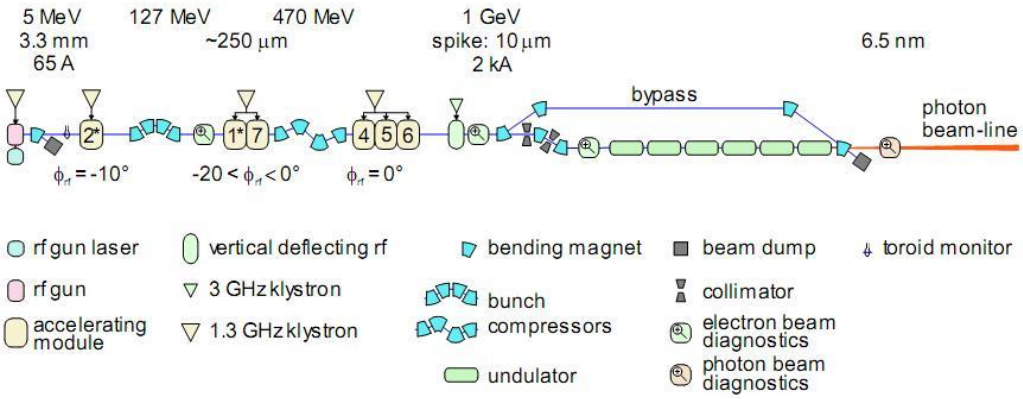


Figure 1.1: Sketch of FLASH before Shutdown October '09, [17]. The Number inside the accelerating modules denotes the DESY internal production description.

light with a wavelength of a few nanometers. The SASE (Self Amplified Spontaneous Emission) principle, which was proven at FLASH, used to generated the intense laser light, is briefly explained in Fig.1.2. The accelerator of a free electron laser (FEL) is the most important part since it needs to deliver an intense high energy electron beam to the magnets with small bunch structure deviations. The relation between the FEL wavelength and the beam energy is $\lambda_L \propto \frac{1}{\gamma^2}$ where γ is the relativistic boost factor depending on the energy of the particle. This is the common point between FLASH and ILC: Reaching stable high energy electron pulses with small emittance. The following chapter deals with the theory and basics of superconductivity and its microscopic explanation, the BCS theory, and the basics of cavity theory. In chapter 3, an overview of control theory is shown, together with digital signal processing and its relations to cavities. Also a method is shown, [4] [5], to prevent cavities to quench under working conditions in presence or absence of beam while reaching the maximum accelerating gradient.

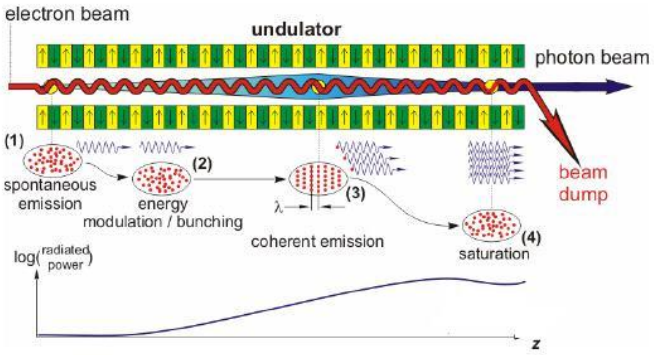


Figure 1.2: (1) The electron bunch enters the undulator and starts with spontaneous emission. (2) The emitted radiation reacts back onto the bunch and leads to a density modulation. The bunch gets 'sliced' where the intra-bunch distance of each slice is about the wavelength of the amplified radiation. This results in coherent emission (3). The effect saturates after a certain undulator distance and the electron beam is transported to the beam dump [43].

The derived principles from chapter 3 are implemented at FLASH and the results are shown in chapter 4 which will explain in detail the feedback system developed and tested at accelerator module 1 (ACC1, No 2* in Fig. 1.1). Chapter 5 describes the work done at the vertical test stand. Furthermore, a description of the hardware and software which has been developed to realize the digital controlled test stand and the results and possible upgrades are presented. In general, the vertical test stand is used to test the cavity to its maximum reachable gradient and to guarantee that a cavity will keep this gradient along its treatment chain at DESY.
Another topic of this work is the so called 9 mA Run, which was performed at FLASH in September '09. This accelerator study aimed at learning as much as possible from a superconducting electron linac with accelerator technology, similar to the technology which will be used at the ILC. Since the ILC technology is not fixed yet, R&D is still important and can lead to many improvements. The general goals of this run are described together with the results. An analysis of the quench prevention, explained in chapter 3, is also contained along with the results of the implementation of the real time quench detection. The main issue of this thesis is to take a look into the high accelerating gradients stability in an accelerator and also to realize such high gradient cavities during production series.

2 Theory of superconductivity and cavities

2.1 Superconductivity

2.1.1 BCS theory

Superconductivity has been discovered in 1913 and describes a state below a critical Temperature T_c where the electric DC resistance of a material vanishes. This effect was characterized by Ginsburg and Landau in a thermodynamical theory but could not explain the microscopic mechanism, the zero resistance and the values of the critical parameters. Bardeen, Cooper and Schrieffler suggested a theory in 1957, named after their initials (BCS-Theory), which explained superconductivity as an effect, based on an attractive force between two electrons in a crystal lattice via virtual phonons, the discrete states of the lattice vibration (see Fig. 2.1). This simple picture implies that a

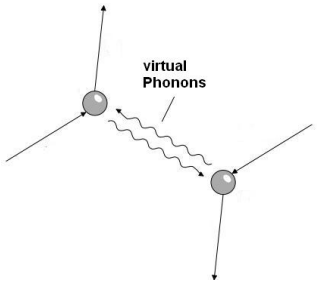


Figure 2.1: Because of phonon interactions between electrons and the atomic lattice, electrons can create stable bound systems and act like bosons [1].

single electron travels along the atomic lattice and polarizes the lattice and an attraction onto a second electron is implied due to this polarization. This attraction is maximized, when the second electron travels along the path of the first electron, but with opposite momentum. At low temperatures, this attraction is strong enough to create bounded states of two electrons (Cooper-pair).

Such a Cooper-pair can be described via quantum mechanics as an antisymmetric spin- and symmetric space-wave function

$$\left\{\vec{k}\uparrow, -\vec{k}\downarrow\right\}$$

This also means that such a pair of electrons can be considered as a boson and all cooper-pairs in the lattice act as a superconducting fluid. The bound state of electrons is energetically favorable below the critical temperature since the two 'condensed electrons' will have a lower energy than two normal electrons would have.

$$E_{2e^-} = 2E_F$$

$$E_{BCS} = 2E_F - \Delta$$

with E_F is the Fermi-energy and Δ is the condensation energy or 'gap' in the range of 10^{-5} eV. The gap itself is temperature-dependent and can be calculated within the BCS-theory to

$$\Delta = 1.76\ k_B T_c$$

If the temperature, the current through the lattice or an external magnetic field, raises to a certain level, the Cooper-pairs will break up and the superconducting state is lost. In Fig. 2.2 a diagram of the phase-space is shown. Inside the surface the superconducting state is given. If too much energy is applied to the lattice, the Cooper-pairs will break

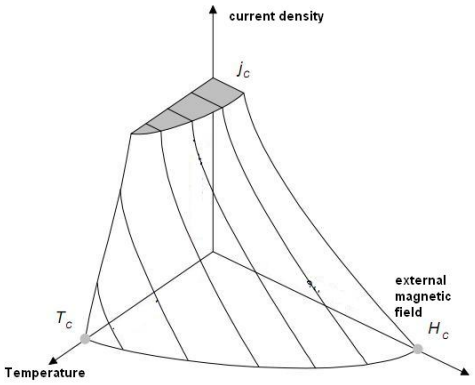


Figure 2.2: Phase space with three parameters: the current density j, the external magnetic field H and the temperature of the system T [1].

up and the material will go into normal conducting state.

These critical parameters are only strictly valid for the so called type I superconductors. For type II superconductors, it is energetically favorable to allow certain magnetic vortices to penetrate the material above the lower critical field. The number of these vortices will raise with the magnetic field until the second critical magnetic field is reached and the whole material will get normal conducting (Fig. 2.3).
The critical magnetic field of a type I superconductor is given by the relation

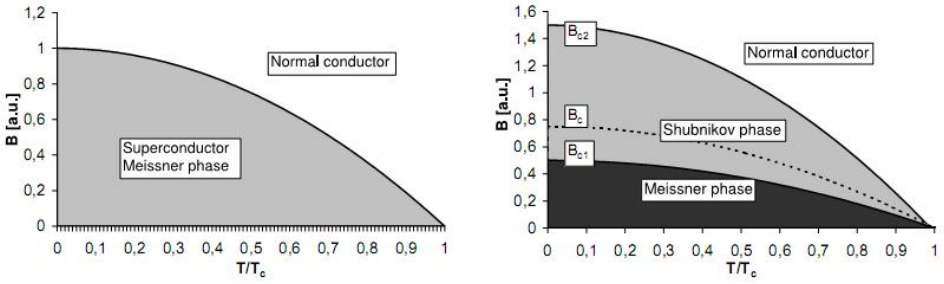


Figure 2.3: Phase diagrams for type I (left) and type II (right) superconductor.

$$B_c(T) = B_c(0)\left[1 - \left(\frac{T}{T_c}\right)^2\right]. \tag{2.1}$$

Another important feature of superconductors is the Meissner-Ochsenfeld effect. If a superconductor is cooled below its transition temperature in presence of a magnetic field, the magnetic flux will be expelled immediately from the volume and a perfect shielding of the inside will appear. This means, that the inside of the volume will be free of any magnetic field. The perfect shielding can be explained in terms of classic physics but not the expulsion of the field. This effect was the evidence, that superconductivity is a seperate thermodynamic phase which has to be described with quantum mechanics. The perfect shielding is generated by a small current, which flows in a thin surface layer. Only this small current will dissipate and create Ohms heating. This is a big advantage of superconducting cavities, since only a small amount of heat has to be cooled, which is an important issue for the cryogenics. To calculate the thickness of the layer, the two-fluid model from London is used.

The so called London penetration depth is given by the characteristic length of the exponential decay of the magnetic field into the superconductors skin

$$B(x) = B(0)\, e^{-\frac{x}{\lambda_L}} \tag{2.2}$$

$$\lambda_L = \sqrt{\frac{m_e}{n_S e^2 \mu_0}}. \tag{2.3}$$

Here λ_L is the penetration depth. With the typical penetration depth for Niobium, a current is flowing within a layer of a thickness of around 30 - 40 nm. Another important parameter, the coherence length ξ_{GL}, describes the distance over which the electrons are correlated. The value can be derived using the Ginsburg-Landau-Theory to

$$\xi_{GL} = \frac{\hbar v_F}{\Delta} \tag{2.4}$$

where v_F is the velocity of the electrons near the Fermi energy. For Niobium, this value is around 39 nm. Interpreting this length as the size of a Cooper pair, it is obvious that this distance is bigger than the lattice constant $a \approx 0.4$ nm and that the Cooper-pairs overlap due to a high electron density per unit cell. This overlap is the reason why all Cooper-pairs act as a uniform fluid. The density of the Cooper-pairs decreases with the depth of the layer due to an energy deposition of the magnetic field at the surface which causes the Cooper-Pairs to break up. In Fig.2.4, this case is depicted. When applying

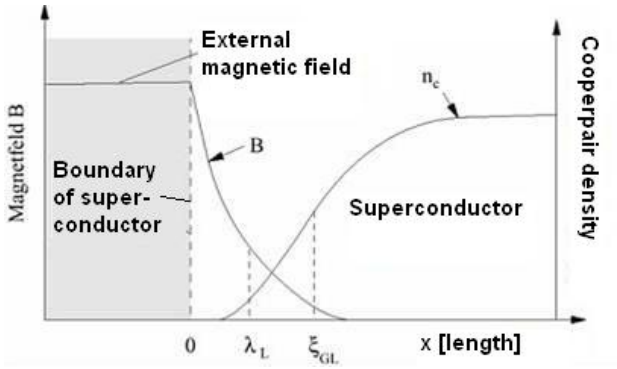


Figure 2.4: In superconducting materials, external magnetic fields can intrude into the material, but only in a thin layer [24].

an external, time independent electromagnetic field, no DC resistance of the flowing super-current arises and no power dissipation occurs. Working with high RF fields, this statement is no longer true.

Cooper-pairs can flow without resistance through the lattice but they still have masses and the inertia has to be taken into account when the electrons are accelerated into different directions within an oscillation period. The formula, describing the surface resistance is

$$R_{surf} = R_{BCS} + R_0 = 2 \cdot 10^{-4} \frac{1}{T} \left(\frac{f}{1.5} \right)^2 \exp\left(-\frac{17.67}{T} \right) + R_0$$

with R_0 as the temperature residual resistance with typical values of $10 - 20$ nΩ. This residual resistance depends on impurities, surface layers of oxides and other foreign material inclusions. Therefore, cleanness during preparation is crucial. This microwave resistance implies, that a critical surface electric field exists, above the current through this layer reaches the critical current density and the Niobium becomes normal conducting. Hence, the surface resistance limits the accelerating gradient to a certain level with respect to the given geometry of the cavity. But the more crucial limitation is the so called quench, which arises when increasing the external magnetic field.

2.1.2 Ways to loose the superconductivity (quench and field emission)

As shown in the previous section, impurities influence the residual resistance of the cavity in the surface skin, because the penetration depth depends on the mean free path of the electrons ℓ in the material. The dependence of ξ on the mean free path is given by

$$\frac{1}{\xi} = \frac{1}{\xi_{GL}} + \frac{1}{\ell}. \tag{2.5}$$

The introduction of the effective penetration depth

$$\lambda_{eff} = \lambda_L \cdot \frac{\xi_o}{\xi} \tag{2.6}$$

reflects that the penetration depth increases with a reduction of the mean free path. For an ideal superconductor ($\ell \to \infty$) one has $\xi = \xi_{GL}$. In the limit of very impure superconductors, we have $\ell << \xi_{GL}$ and the relation becomes

$$\xi = \ell \, .$$

The mean free path of electrons in Niobium is strongly influenced by impurities like Oxygen, Nitrogen, Carbon or metals like Titanium or Iron. While the amount of metals can be reduced by purification, other elements play an important role due to surface treatments with acids, like hydrogen residua.

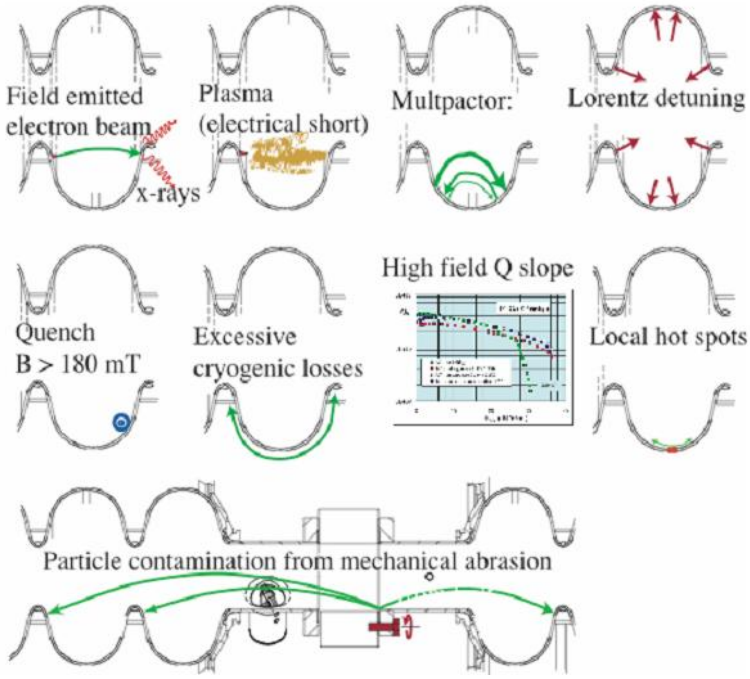


Figure 2.5: Example of different mechanisms in a cavity which can cause the loss of superconductivity [22].

These impurities become relevant when discussing the quality factor of a cavity and its dependence on the accelerating field. Besides the problems due to impurities below and on the surface several other mechanisms exist which show up when applying an external electromagnetic field. They are sketched in Fig.2.5. Two main problems nowadays will be discussed in more detail. The field distribution is shown in Fig.2.6. The magnetic field has its peak value at the equator area, while the peak electric field is at the iris. The

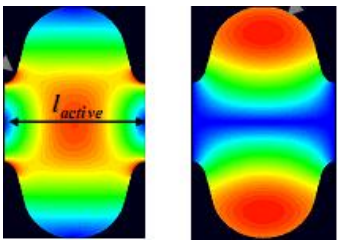


Figure 2.6: Left: electric field strength inside a TESLA cavity, Right: magnetic field strength inside a cavity. High fields are marked with red [25].

critical magnetic field of Niobium is 230 mT at 1.8 K and 180 mT at 2 K (see equation 2.1). Above this value, the cavity will loose its superconducting state or the cavity will quench. But even at lower magnetic fields, a cavity can quench. This problem arises due to extended effects, e.g. foreign material inclusions, which has a lower critical temperature or is not superconducting at all or geometry effects. The mechanism is shown in Fig. 2.7. If the dissipated power into this defect is high enough, the surrounding region can heat up and causes the quench. The area of the normal conducting state will grow exponential, since a bigger area implies a bigger dissipation.

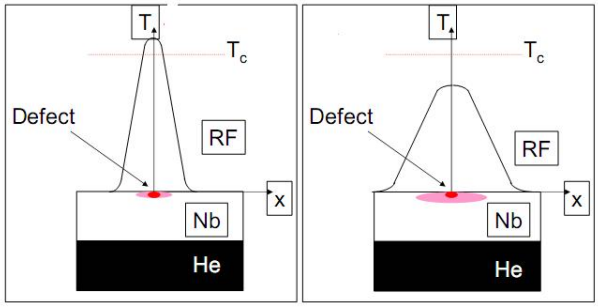


Figure 2.7: Principal mechanism of a quench [13].

Beneath quenching, the so called field emissions puts another barrier before reaching a high field in the cavity. Basically, field emissions are described by the Fowler-Nordheim-Theory (Fig. 2.8). They discussed the emission of electrons from a surface A by applying an external electric field E. The current is then calculated to

$$I_{FN} = A \cdot \frac{C \cdot E^2}{\Phi} \exp\left(-\frac{B\Phi^{\frac{3}{2}}}{E}\right) \tag{2.7}$$

where B and C are constants and Φ is the working function of the metal. When applying this formula to Niobium, assuming a current of 1 μA, an electric field of 3 GV/m will result in field emission. Obviously, this field would be too high to produce field emission in a cavity, since we can reach accelerating fields up to 40 MV/m. But taking into account, that this formula describes perfect flat surfaces, we see where the problem arises. Scratches, pits and bumps on the surface and impurities with different working functions can lead to field emissions (Fig. 2.9) [44].

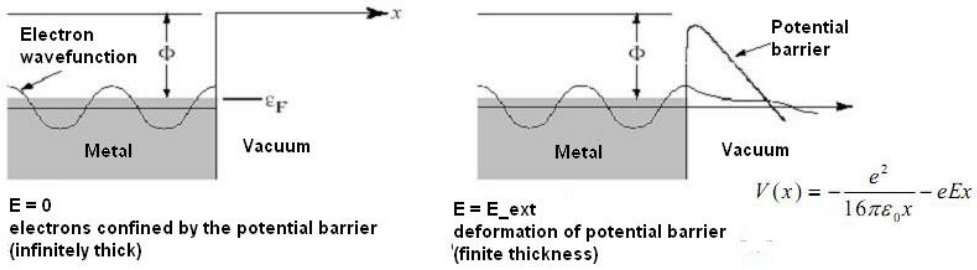
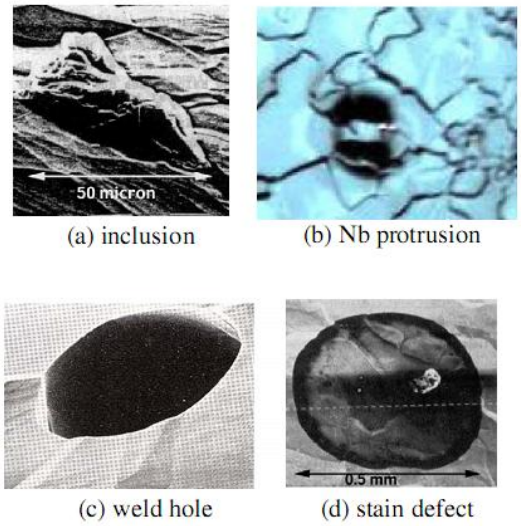


Figure 2.8: The mechanism of the Fowler-Nordheim-Emission. Left: the metal surface in vacuum without an external electric field. The potential Barrier has infinite thickness and no electron will tunnel into the vacuum. Right: an external electric field is applied, which decrease the potential barrier and electrons can leave the metal surface [13].

If we want to identify the spot where the quench arises and take a closer look at the defect, a so called 'Temperature Map' or just T-Map is applied. By assembling thermal dependent resistors to the outer surface of the cavity and running a test, the resistance of the sensors are measured and will change if the area below the resistors heats up. This resistor change is mapped onto the cavity surface and defects can be localized within given tolerances (Fig.2.10).



(a) inclusion (b) Nb protrusion

(c) weld hole (d) stain defect

Figure 2.9: Some examples of the inner cavity surface which will lead to a quench or field emission [23].

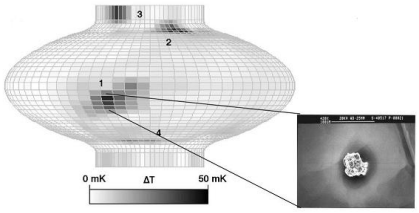


Figure 2.10: Example of Temperature Mapping of a superconducting cavity with an impurity on its surface. This impurity leads to a local quench [13].

2.2 Toy model - the pillbox cavity

To illustrate the basic concepts and important quantities of a cavity, such as the accelerating field, shunt impedance and quality factor, an idealized cavity, the so called pillbox cavity will be used. This is just a toy model since the geometry tends to exhibit strong field emission and no beam pipe is attached but it allows an analytical description of the cavity. With regard to a conducting surface in a vacuum and an external electromagnetic field, the boundary conditions are

$$\vec{E} \perp \vec{n} \wedge \vec{H} \| \vec{n} \tag{2.8}$$

and can be written as

$$\vec{E} \cdot \vec{n} = 0 \wedge \vec{H} \times \vec{n} = 0. \tag{2.9}$$

The field, excited by an rf source in an infinite waveguide with uniform cross-section and a perfect conducting surface is

$$\vec{E}(\vec{x}, t) = \vec{E}(\rho, \phi) e^{ikz - i\omega t} \tag{2.10}$$

$$\vec{H}(\vec{x}, t) = \vec{H}(\rho, \phi) e^{ikz - i\omega t} \tag{2.11}$$

where ω is the angular frequency and k is the wave number.ρ and ϕ are the radius and angle in cylindrical coordinates. Using Maxwell's equations to derive the wave equation

$$\left(\nabla^2 - \frac{1}{c^2} \frac{\partial^2}{\partial t^2} \right) \begin{Bmatrix} \vec{E} \\ \vec{H} \end{Bmatrix} = 0 \tag{2.12}$$

and putting (2.8) and (2.9) into (2.12)

$$\left[\left(\nabla^2 - \frac{\partial^2}{\partial z^2} \right) + \left(\frac{\omega^2}{c^2} - k^2 \right) \right] \begin{Bmatrix} \vec{E} \\ \vec{H} \end{Bmatrix} = 0 \tag{2.13}$$

$$\Leftrightarrow \left[\nabla_\perp^2 + \left(\frac{\omega^2}{c^2} - k^2 \right) \right] \begin{Bmatrix} \vec{E} \\ \vec{H} \end{Bmatrix} = 0 \tag{2.14}$$

The boundary conditions (2.8) and (2.9) can be rewritten to

$$E_z|_S = 0 \wedge \left. \frac{\partial H_z}{\partial n} \right|_S = 0. \tag{2.15}$$

The independence of E_z and H_z and the different boundary conditions imply that the solutions (2.14) form two sets of modes with different eigenvalues.

The names used for these families of modes are TM and TE, where TM means $E \| z$ and $H \perp z$ and the other way around for TE. By adding conducting surfaces at the knots of the wave, the pattern of the field is not changed. Since the knots can be chosen arbitrary, surfaces can be placed at z = 0 and z = d. Due to reflections at the surfaces, standing waves are created as a combination of forward and backward traveling waves

$$E_z(\vec{x},t) = \Psi(\rho,\phi)\cos\left(\frac{p\pi z}{d}\right)e^{i\omega t}\ p \in N,\ \text{TM-Mode} \tag{2.16}$$

$$H_z(\vec{x},t) = \Psi(\rho,\phi)\sin\left(\frac{p\pi z}{d}\right)e^{i\omega t}\ p \in N,\ \text{TE-Mode} \tag{2.17}$$

$$\tag{2.18}$$

With $k = \frac{p\pi}{d}$ and substituting these expressions into (2.14) we obtain the fields $\Psi(\rho,\phi)$ which are the solutions to the eigenvalue equations

$$\left(\nabla_\perp^2 + \gamma_j^2\right)\Psi(\rho,\phi) = 0 \tag{2.19}$$

$$\gamma_j^2 = \left(\frac{\omega j}{c}\right)^2 - \left(\frac{p\pi}{d}\right)^2 \tag{2.20}$$

where γ_j is the j-th eigenvalue. An interesting relation is found, when scaling the electric and magnetic field with the same constant a. The fields will be the same in the cavity, but it can be shown that the relation $\omega_j \propto \frac{1}{a}$ is true. This means, that the mode spectrum is inversely proportional to the cavity size.
Consider a cylindrical conductor of length d and radius R (Fig. 2.11).

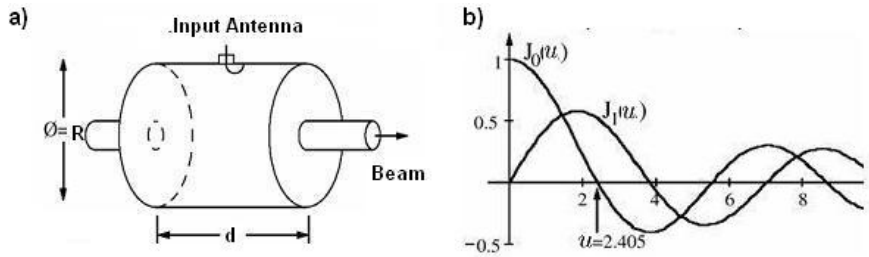


Figure 2.11: a) shows the geometry of a pillbox cavity. b) depicts the J_0 and J_1 Bessel function which describes the radial electric field or the radial magnetic field of the cavity, respectively.

In this case, Ψ is derived as solution of (2.19) applied to this geometry and the eigenvalues are the Bessel-functions. Investigate the case of the lowest frequency, the field distribution is given by

$$E_z = E_0 J_0 \left(\frac{2.405\rho}{R} \right) e^{i\omega t} \tag{2.21}$$

$$H_\phi = -i \frac{E_0}{\eta} J_1 \left(\frac{2.405\rho}{R} \right) e^{i\omega t} \tag{2.22}$$

with $\eta = \sqrt{\mu_0/\epsilon_0}$. A plot of the distribution is shown in Fig. 2.12. The resonance frequency is given by

$$\omega = \frac{2.405c}{R}.$$

This corresponds to the lowest frequency of the pillbox cavity.

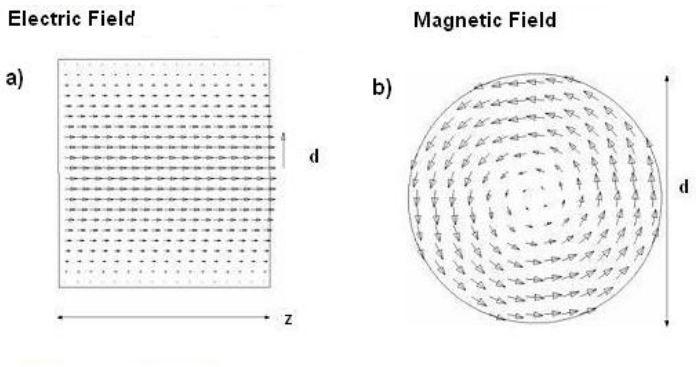


Figure 2.12: This picture shows the electric (a) and magnetic (b) field distribution inside a pillbox cavity of the lowest Eigenmode, the so called TM_{010} mode.

The complete set of TM modes has an own nomenclature which is used for every cavity. The classification is TM_{mnp}. The integers m,n,p describe the number of sign changes the electric field will have, when going in the direction of ϕ, ρ and z. There exists a large number of higher order modes (HOM) since any combination of m,n and p is possible. For the whole set of equations, more details are given in [13]. The TM-modes with the notation TM_{1np} are so called dipole modes (Fig. 2.13 (a) and (c)) which are unwanted in accelerators, because they have a deflecting field along the z-axis which will disrupt the beam. In order to accelerate the beam, a non-vanishing electric field along the z-axis, which is only given by the J_0 Bessel function, is needed. These modes are called monopole modes and the notation is TM_{0np}. Usually, the TM_{010} is chosen, since it has the lowest eigenfrequency. The TE modes do not have a longitudinal electric field and thus ca not accelerate the beam nor can they be excited by the beam.[1] Some other modes are shown in Fig. 2.13. By now, the electric field along the z-axis was calculated. But we want to know the accelerating voltage, e.g. the energy the particle gains while traveling through the cavity. First, we assume that the particle travels close to the speed of light. This is given e.g. for an electron with an energy greater than 10 MeV. It enters the accelerating cavity on z-axis at time t=0 and leaves it at t=d/c=T_{cav}. During this transit, the electron is exposed to a time varying field. The time the electron need to travel through the cavity needs to be the half of an rf period so the electron receives the maximum energy gain from the cavity:

$$T_{cav} = \frac{d}{c} = \frac{\pi}{\omega_0}.$$

Here ω_0 is the frequency of the accelerating mode. If the phase of the electric field and the arrival time of the electron is synchronized in the right way, the electron enters the cavity just when the field changes sign. This implies that the electron will see a field pointing in only one direction. The accelerating voltage is then defined by

$$V_c = \left| \int_0^d E_{el} dz \right| \tag{2.23}$$

with

$$E_{el} = E_z(\rho = 0, z) e^{i\omega_0 z/c + i\varphi}.$$

When φ is an arbitrary phase, (2.23) can be written as

$$V_c = \left| \int_d^0 E_z(\rho = 0, z) e^{i\omega_0 z/c + i\varphi} dz \right| \tag{2.24}$$

[1] This is only true as long no beam pipes are attached to a cavity.

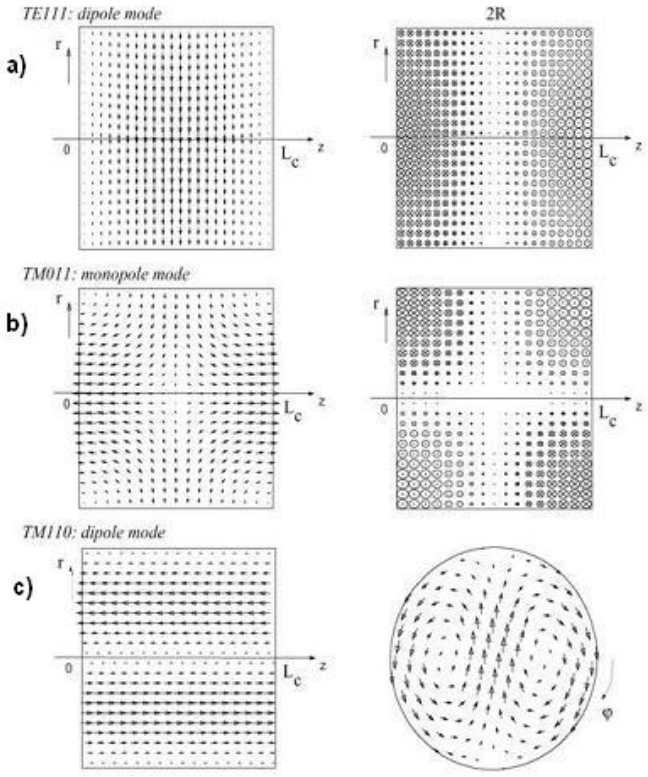


Figure 2.13: Different modes have different field contributions, like a deflecting electric or magnetic field along the axis (left: electric field / right: magnetic field).

If we consider the pillbox cavity in the TM_{010} mode, we put (2.21) into (2.24) and find

$$V_c = E_0 \left| \int_d^0 e^{i\omega_0 z/c} dz \right| = d \cdot E_0 \frac{\sin\left(\frac{\omega_0 d}{2c}\right)}{\frac{\omega_0 d}{2c}} = d \cdot E_0 \cdot T \tag{2.25}$$

The parameter T is known as the transit factor and can be understood as a measure for the effective voltage a particle will see, since it will not see a constant voltage but a time dependent field. This simplified expression for the transit factor only works for the pillbox cavity with the T_{cav} relation from above. The accelerating gradient E_{acc} is defined as

$$E_{acc} = \frac{V_c}{d}$$

which evaluates to $E_{acc} = 2E_0/\pi$ in case of the TM_{010} pillbox cavity. In section 2.1.1, we derived an expression for the AC resistance depending on the temperature and the frequency. At working temperatures of the TESLA cavities, it is in the order of several tens of nano-ohms if we are dealing with well prepared Niobium surfaces. This leads to a small deposition of energy in a thin layer due to Joule heating.

$$\frac{dP_c}{ds} = \frac{1}{2} R_s \left| \vec{H} \right|^2 \tag{2.26}$$

with dP_c/ds as the dissipated power per area and $\vec{H}$ is the local magnetic field. The modifications of the local magnetic field through the current flowing in the thin layer can be neglected. An important quantity, discussing the cavity types/designs and material is the so called (unloaded) quality factor Q_0 which is defined as

$$Q_0 = \frac{\omega U}{P_c} \tag{2.27}$$

where U is the energy stored in the cavity and P_c is the dissipated power into the cavity walls per cycle. It can be interpreted as a number of oscillations the field will do with a given amount of energy since we can decipher the equation as

$$Q_0 = \omega \frac{\text{Energy stored in the cavity}}{\text{dissipation per one oscillation}} \tag{2.28}$$

The total stored energy in a cavity is calculated by

$$U = \frac{1}{2}\mu_0 \int_V \left| \vec{H} \right|^2 dV = \frac{1}{2}\epsilon_0 \int_V \left| \vec{E} \right|^2 dV \tag{2.29}$$

integrating over the cavity volume. Integrating (2.26) yields

$$P_c = \frac{1}{2} R_s \int_A \left|\vec{H}\right|^2 dA. \tag{2.30}$$

Assuming the surface resistance is constant for the whole surface, (2.27) can be revised to

$$Q_0 = \frac{\omega_0 \mu_0 \int_V \left|\vec{H}\right|^2 dV}{R_s \int_A \left|\vec{H}\right|^2 dA} = \frac{G}{R_s} \tag{2.31}$$

where G is known as the geometry constant. An aspect of this geometry constant, when taking a relation mentioned above into account, is

$$G \propto \frac{\int_V \left|\vec{H}\right|^2 dV}{\int_A \left|\vec{H}\right|^2 dA} \propto a \propto \frac{1}{\omega_j} \tag{2.32}$$

This constant is a good parameter to compare different cavity designs, since its definition is independent of material specifications and size but not to the shape of the cavity. Another important quantity for a cavity is the so called shunt impedance. It is a direct measure of the beam-cavity interaction and should be large for the accelerating mode and small for any other mode, otherwise the beam could excite them. It is also used to characterize the losses in a cavity. Many definitions for the shunt impedance exist. For example, the accelerator definition is

$$R = \frac{V_c^2}{P_c} \tag{2.33}$$

which has the unit ohms per cell. A definition, coming from the circuit theory which will be explained later is

$$R^c = \frac{V_c^2}{2P_c} \tag{2.34}$$

with the unit of ohms per unit length. Or even the linear accelerator definition

$$r = \frac{V_c^2}{P_c'} \tag{2.35}$$

given in ohms per meter.

Since R is a quantity of the beam - cavity interaction, it should be maximized in such a way that the dissipation of the energy into the cavity walls is as small as possible and as much energy as possible is transfered to the beam. Typically, the ratio R/Q_0 is given, so that the value is independent of the material since it does not depend on the surface resistance.

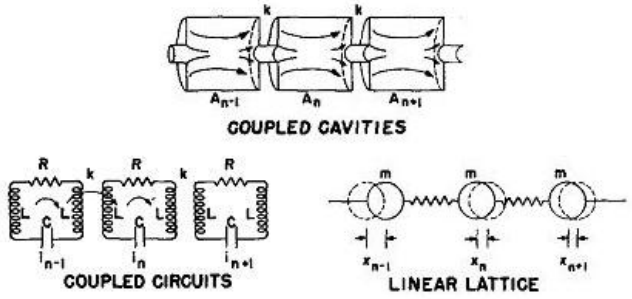


Figure 2.14: Since a single cavity will have a single resonance frequency, N coupled cavities will have N resonance frequencies. To describe such a system, Thevenin's Theorem is used. This theorem allows us to draw an equivalent circuit of a coupled system and to describe it in terms of electric circuits [42].

So far, we discussed a single cell cavity. Since the TESLA cavities are nine-cell cavities we have to think about the implications for a multi-cell cavity. The shunt impedance, the geometry factor and due to this, the peak field of the magnetic and electric field will change. For steering a cavity, these issues are not further important. Taking a look at Fig. 2.14, the biggest difference is depicted. A cavity can be described as coupled resonators. And when only a single acceleration mode was present at the beginning, we will now have 9 'acceleration' modes, which will be called passband (Fig. 2.15). This passband of the lowest eigenmodes of the cavities are all monopole modes, each have a non vanishing acceleration field along the beam axis. But how about the amplitude along the axis? What is the phase shift from cell to cell? If the beam will enter a cell at the wrong phase, just a small acceleration will occur, or even a deceleration can happen. A small discussion about the TM_{010} passband will be given and which mode of this passband should be used for acceleration.

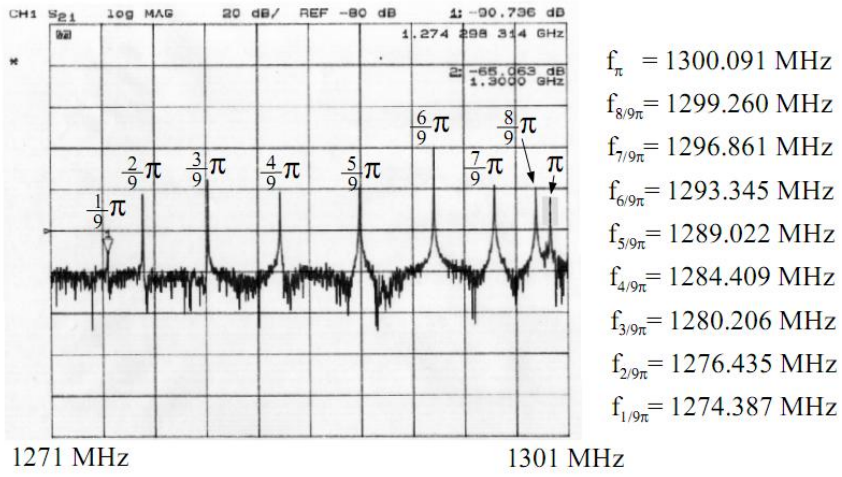


Figure 2.15: In this picture, the passband of a single TESLA cavity is shown. The typical bandwidth is 30 MHz with a difference between the π-mode and the 8/9-π-mode of roughly 800 kHz [11].

To analyze coupled cavities, we assume that each cell has only one resonant mode [13]. Since we are only interested in the TM_{010} mode, this is no disadvantage. An electric circuit model is used, to describe the cell to cell coupling (Fig. 2.16). In this model,

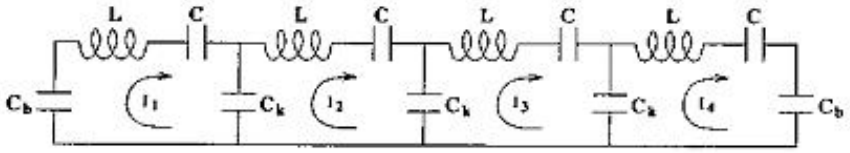


Figure 2.16: Equivalent circuit of a 4-cell cavity.

L and C are the characteristic inductance and capacitance of each cell. Only the next neighbour coupling is considered and this is done capacitively via C_k. The beam tubes are modeled by a capacitance C_b. Since we have $1 \ll Q$, the resistance of circuit can be set to zero. Using Kirchhoff's rules of the summation of voltages in a circuit for each current loop I_j and remembering that the impedances of an inductor is $i\omega L$ and for a

capacitor $1/(i\omega C)$, the following coupled equations for a N-cell cavity can be derived:

$$\left(\frac{1}{i\omega C_b} + i\omega L\right) I_1 + \left(\frac{1}{i\omega C}\right) I_1 + \left(\frac{1}{i\omega C_k}\right)(I_1 - I_2) = 0 \tag{2.36}$$

$$\frac{1}{i\omega C_k}(I_j - I_{j-1}) + \left(i\omega L + \frac{1}{i\omega C}\right) I_j + \left(\frac{1}{i\omega C_k}\right)(I_j - I_{j+1}) = 0 \tag{2.37}$$

$$\left(\frac{1}{i\omega C_k}\right)(I_N - I_{N-1}) + \left(i\omega L + \frac{1}{i\omega C}\right) I_N + \left(\frac{1}{i\omega C_b}\right) I_N = 0. \tag{2.38}$$

Now multiply by $i\omega C$ and define $\omega_0^2 = 1/LC$, $k = C/C_k$, $\gamma = C/C_b$ and

$$\Omega = \frac{\omega^2}{\omega_0^2}$$

then the equations simplify to

$$(1 + k + \gamma)I_1 - kI_2 = \Omega I_1 \tag{2.39}$$

$$-kI_{j-1} + (1 + 2k)I_j - kI_{j+1} = \Omega I_j \tag{2.40}$$

$$-kI_{N-1} + (1 + k + \gamma)I_N = \Omega I_N \tag{2.41}$$

These equations can be written as a vector equation with a tridiagonal matrix equation

$$\mathbf{A}\vec{V} = \begin{pmatrix} 1+k+\gamma & -k & 0 & \dots & 0 \\ -k & 1+2k & -k & & \vdots \\ 0 & & \ddots & & 0 \\ \vdots & & -k & 1+2k & -k \\ 0 & \dots & 0 & -k & 1+k+\gamma \end{pmatrix} \vec{V} = \Omega\vec{V} \tag{2.42}$$

In the following equations, the following notation is used: subscripts refer to the cell number, superscripts to the mode in the passband. The use of the voltages instead of the currents is made to remind, that we are now dealing with cavities and not with circuits anymore. Since we want modes with flat amplitudes the Nth normalized eigenvector must be

$$\vec{V}^{(N)} = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 \\ -1 \\ 1 \\ 1 \\ \vdots \end{pmatrix}. \tag{2.43}$$

with -1 at the nth position of the vector. This boundary condition allows us to determine the beam tube parameter γ, when substituting (2.43) into (2.42). By solving the first two equations, we find that (2.43) is satisfied by $\gamma = 2k$. The solution to (2.42) can be easily found, and the eigenvectors are given by

$$V_j^{(m)} = B^{(m)} \sin\left[m\pi\left(\frac{2j-1}{2N}\right)\right] \tag{2.44}$$

for cell number $j = 1 \ldots N$ and the mode $m = 1 \ldots N$. $B^{(m)}$ is the normalizing coefficient

$$B^{(m)} = \sqrt{(2 - \delta_{mN})/N}.$$

In Fig. 2.17 the eigenvectors for the TESLA cavity are shown. To find the eigenvalues one has to insert (2.44) into the jth equation of (2.42). Solving the given equation for Ω, we get

$$\Omega^{(m)} = 1 + 2k\left[1 - \cos\left(\frac{m\pi}{N}\right)\right] \tag{2.45}$$

With stronger cell to cell coupling, the spacing between the modes will increase and it will decrease with the number of cells. Having a cavity with equal and opposing traveling waves which form a standing wave pattern, the phase advance between neighbour cells can only be zero or π. Since we want a acceleration along a multicell cavity, we use the phase advance per cell of π along the structure. This is the origin of the name π-mode used for acceleration because $\frac{m\pi}{N}$ depicts the phase shift per cell for the m-th mode and with N cells.

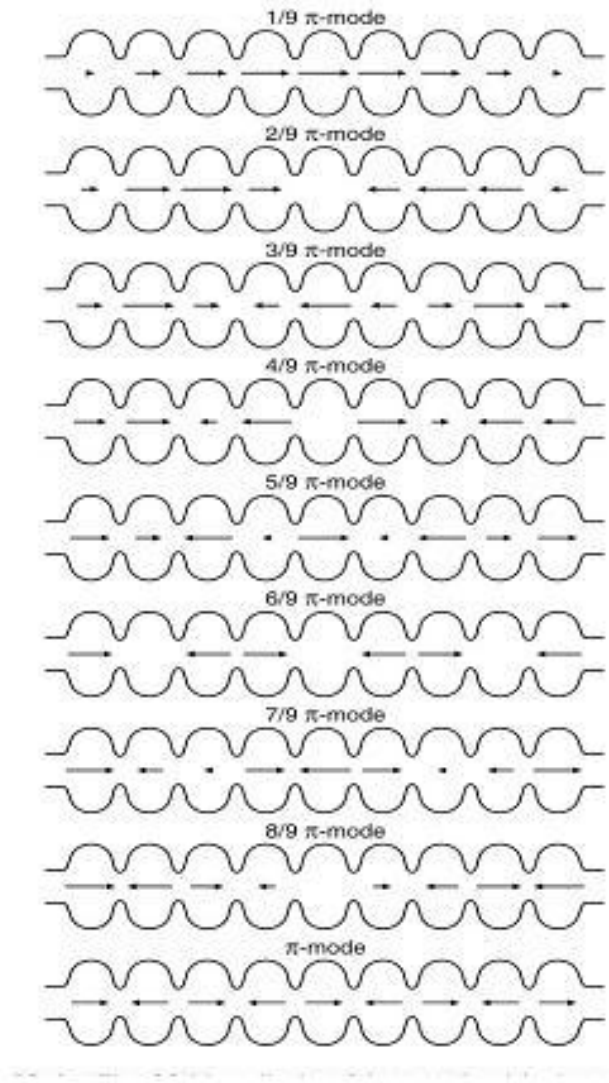


Figure 2.17: The eigenvalues of the electric field strength, represented as length of the arrow in each cell, for different modes. In the π-mode, the lowest case here, the distribution is nearly equal in each cell but with a phase shift of π per cell

Now, we derived and described the basic and elementary quantities for a cavity. Since one goal of this thesis is to achieve not only high gradients but also high quality factors, how can we determine the unloaded Qualityfactor, Q_0, during a measurement? For superconducting cavities this is done by sending a rectangular pulse with frequency of the mode of interest into the cavity. The cavity will fill, which means that the electric field will increase until it reaches its maximum amplitude with the given input power. After turning the RF off, power conservation gives us

$$P_{tot} = P_c + P_e + P_t. \tag{2.46}$$

The total power inside the cavity is the sum of the ohmic losses in the cavity walls P_c, the power leaking back through the input coupler P_e and the power picked up in the probe antenna P_t. Analog to the unloaded Q_0, a loaded Qualityfactor can be defined as

$$Q_L = \frac{\omega U}{P_{tot}}. \tag{2.47}$$

This loaded Qualityfactor, Q_L, characterizes a cavity with input coupler, or in general with any attached device. Usually the attached devices will change the Q_L value in such a way, that it is several orders of magnitude smaller than the Q_0. This is clear, because any attached device allows an additional dissipation effect. The cavity field energy decaying after the rf is turned off can be written as

$$\frac{dU}{dt} = -P_{tot} = -\frac{\omega U}{Q_L} \tag{2.48}$$

The solution is a simple exponential decaying field

$$U(t) = U_0 exp\left(\frac{-\omega t}{Q_L}\right) \tag{2.49}$$

$$\Rightarrow \tau_c = \frac{Q_L}{\omega}. \tag{2.50}$$

This means that the measurement of the decay time is a direct measurement of the Q_L. Next step is to derive the Q_0 from this. We put (2.47) into (2.48) and take the inverse, leading to

$$\frac{P_{tot}}{\omega U} = \frac{P_c + P_e + P_t}{\omega U} \tag{2.51}$$

$$\Rightarrow \frac{1}{Q_L} = \frac{1}{Q_0} + \frac{1}{Q_e} + \frac{1}{Q_t} \tag{2.52}$$

when we assume analog loss mechanisms for each dissipation, which allows us to state a quality factor for each mechanism. If we now define the coupling strength β

$$\beta_x = \frac{Q_0}{Q_x} \tag{2.53}$$

$$\Rightarrow \frac{1}{Q_L} = \frac{1}{Q_0}\left(1 + \beta_e + \beta_t\right) \tag{2.54}$$

$$\Leftrightarrow Q_0 = Q_L\left(1 + \beta_e + \beta_t\right) \tag{2.55}$$

we can calculate the Q_0 with the measured parameters and when the coupling is known. More details are shown in chapter 5.

3 Control theory applied to superconducting cavities

Control theory deals with the design and analysis of closed loop control systems. The design of a digital signal processing loop is hardly possible without a basic mathematical understanding of the signal and its properties. Both, transferfunction (classical control) and state-space representation (modern control) and the mathematics and physics of the underlying processes are described. After the basics have been set in Section 3.1, we will concentrate on the accelerator itself (Section 3.2) and in Section 3.3 especially onto a single cavity. The last section will consider the use of the derived results at FLASH.

3.1 Digital signal processing

In general there are several reasons for digital accelerator control. A high flexibility, easy upgrades and repeatability is given and multi user operation is possible. Also, a longer distance between the sampling point and processing of the signal is possible in comparison to analog control. Remote control and diagnostics is possible without additional hardware. In comparison, analog control shows shorter latency and bigger radiation hardness. But these problems of the digital control can be solved by further, more sophisticated technologies. The algorithms which are needed to be implemented to have an finite state machine are quite sophisticated but can be implemented in digitally controlled systems. Besides feedback, feedforward and even adaptive feedforward can be realized in digital controller. Fault recovery, exception handling and the changing of beam settings, turn on procedures and other high level applications (e.g. automated cavity tuning) can be managed much easier and completely automated. The needed methods and definitions will be explained and derived in this section.

3.1.1 Basic definitions

Before we can get deeper into control theory, we need to define the term 'signal'. A signal s(t) is generated by a measurement and can be seen as a real, time-dependent property and it should be an energy signal, which is

$$\int_{-\infty}^{\infty} s^2(t)dt < \infty.$$

The instant power is defined as $s^2(t)$. Most of the elementary functions are no energy signals, because they are infinitely long and the integral does not converge. But treating them as power signals

$$\lim_{T\to\infty} \int_{-T/2}^{T/2} s^2(t)dt < \infty$$

we will have finite values. In the end, we will get a signal, which can be discrete or continuous, analog or digital, causal or non-causal. We will only consider causal systems, which means that the future state of the system only depends on the past and the present state and can be calculated with the previous signals. Another definition is the domain. We will consider only two domains, time and frequency domain, where it depends on the simplification of the problem which representation will be chosen. How to switch between these representations will be explained in the next section. After all, we will consider only **l**inear, causal, **t**ime-**i**nvariant system (LTI-systems) which can be described by **l**inear **o**rdinary **d**ifferential **e**quations (LODE). The complete definition of such a LTI-system is

1. **Linearity**
   Given system F with $F(x_1[n]) = y_1[n]$ and $F(x_2[n]) = y_2[n]$, then F is called linear if
   $$F(x_1[n] + x_2[n]) = F(x_1[n]) + F(x_2[n])$$
   and with this, for two linear systems:
   $$F_2(F_1((x[n])) = F_1(F_2(x[n]))$$

2. **Time-invariance**
   Given F with $F(x[n]) = y[n]$, then F is time- or shift-invariant if
   $$F(x[n-k]) = y[n-k] \;\forall\; k \in \mathbb{N}$$

3. **Causality**
   If the output depends only on present and past input values it can be written as

$$y[n] = \sum_{i=-\infty}^{i=0} a_i \cdot x[n-i]$$

The last distinction we use is the type of a system, depending on the number of inputs and outputs.

1. MIMO (multiple input - multiple output) systems; most general
2. SISO (single input - single output) systems; elementary systems like gain, delay and combinations of those
3. MISO (multiple input - single output) systems; for example an adder (in rf nomenclature a mixer)

With these basic definitions, we can start to construct everything we need.

3.1.2 Control theory

Impulse response

The impulse response of a linear system is its response to a δ-pulse on its input (see Fig. 3.1). A quite often used analogy is the ringing bell which got hit by a hammer. The hammer excites a δ-like excitation and the bell will ring. The ringing is the impulse response of the system 'bell'. The way, the bell rings is characteristic for the system and contains eigenfrequencies, each decaying with its own characteristic time constant. The impulse response h[n] is the fingerprint of the system and all informations can be found in the impulse response.
Two systems with the same impulse response are identical. The input signal is written as

$$x[n] := \sum_{i=0}^{N-1} x_n \delta[n-i] \tag{3.1}$$

then the output of the system can be expressed as

$$y[n] := \sum_{i=0}^{N-1} x_n h[n-i] \tag{3.2}$$

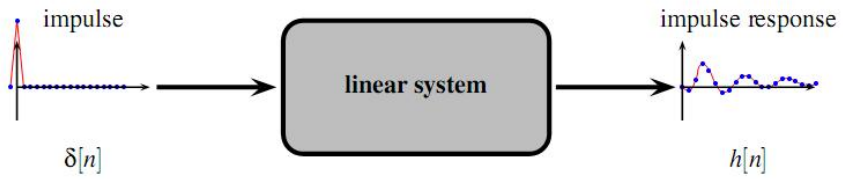


Figure 3.1: The concept of impulse response [21].

This is only possible in linear systems because a sum will stay a sum and a product of scalars will stay a product of scalars. Knowing the impulse response of the system, one can calculate the output of any given input signal [1]. Another way to write (3.2) is

$$y[n] = x[n] * h[n]$$

where $*$ is the convolution operator which is defined as

- **continuous convolution**
  Given two functions $f, g : D \to \mathbb{R}$ where $D \subseteq \mathbb{R}$ the convolution of g with f, written $f * g$ is defined as
  $$(f * g)(t) := \int_D f(\tau) g(t - \tau) d\tau$$
- **discrete convolution**
  Given two functions $f, g : D \to \mathbb{C}$ where $D \subseteq \mathbb{Z}$ the convolution of g with f, written $f * g$ is defined as
  $$(f * g)[n] := \sum_{k \in D} f[k] g[n - k]$$

Transformations

As mentioned before the domain in which we represent the LODE or impulse response of the system, is chosen in such a way that it can be treated and understood in a simple way. The principles of domain transformation are explained in this section. In general one can choose between time or frequency domain, but also auto-correlation or other spatial domains exist.

[1] See the analogy to the Green function in mechanics or electrodynamics or to the propagators in quantum mechanics.

For our case it is appropriate limiting ourselves to the time and frequency domain. We start with the Fourier transformation, a linear operator, which maps functions to the complex space. It can be interpreted as a decomposition of a time-continuous signal into its frequency components $x(t) \rightarrow U(\omega)$. The definition is as follows:
Given $f : D \rightarrow \mathbb{C}, D \subseteq \mathbb{R}$, the Fourier transformation of f is

$$F(\omega) := \int_D f(t) e^{-i\omega t} dt.$$

An important feature of the Fourier transformation is

$$F\left\{x_1(t) * x_2(t)\right\} = X_1(\omega) \cdot X_2(\omega).$$

Therefore the Fourier Transformation can be quite handy to describe LTI systems, but it is not always well defined, as damped or growing oscillations can not be decomposed into ordinary sine and cosine functions. Furthermore the step function would lead to an infinite frequency spectrum. But these inconsistencies can be solved. Adding exponential damped or growing sine and cosine functions by substituting the frequency $i\omega$ by a complex number p:

$$p = \sigma + i\omega$$

where ω is the known real frequency and σ is an arbitrary (real) damping term. The result is the more generalized Laplace Transformation

$$S(p) := \int_0^\infty s(t) e^{-pt} dt.$$

In Fig.3.2 a sketch of the link between the domains due the Laplace transformation is given. The z-transformation, a tool which is as powerful as the Laplace-transformation, but which is applicable to digital systems and signals, can be introduced. The concept is based on the periodicity of digital signals.
Given $h : \mathbb{Z}_0^+ \rightarrow \mathbb{R}$, the z-transformation of h is

$$H(z) := \sum_{n=-\infty}^{\infty} h[n] z$$

with $z = e^{i\omega}$ This substitution maps the frequency axis to the unit circle in the complex plane. This concept is useful because it automatically accounts for the periodicity of ω. The z-plane or the unit circle is a representation of one period of the digital frequency. By adding a damping term σ to the digital frequency, we can extend this concept onto the whole complex plane $\mathbb{C}$. This representation allows a gentle way to discuss the stability of a system, what can be seen in Fig. 3.4.

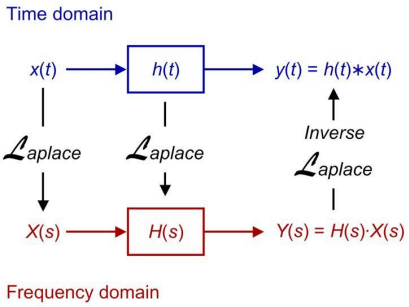


Figure 3.2: The representations in the frequency or time domain for continuous signals are linked via the Laplace-transformation, which can be considered as a more general Fourier-Transformation. For discrete time systems you have to use the so called z-Transformation [21].

Transferfunctions and state space

The transferfunction (of a continuous signal) is defined as the Fourier- or Laplace transformation of the impulse response and can be written as

$$Y(s) = H(s)U(s)$$

where Y(s) and X(s) are the transformed signal of the output and input (see Fig. 3.3)

$$H(s) = \frac{Y(s)}{U(s)}$$

The general form of a LODE is

$$y^{(n)}(t) + a_{n-1}y^{n-1}(t) + \ldots + a_0 y(t) = b_m u^{(m)}(t) + b_{m-1}u^{n-1}(t) + \ldots + b_0 u(t)$$

with m,n $\in \mathbb{N}_0$ and $a_i, b_i \in \mathbb{R}$. With this LODE, the transferfunction in its polynomial form looks like

$$H(s) = \frac{\sum_{i=0}^{i=n} a_i s^i}{\sum_{i=0}^{i=m} b_i s^i}$$

While the transferfunction is only valid for SISO-systems, the State-Space-Representation is applicable to MIMO-systems and will be the concept of choice throughout the later sections when we describe the cavity and model the behavior of the complete system.

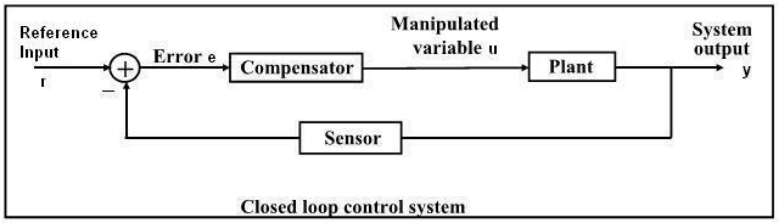


Figure 3.3: Controller and plant in closed loop operation [21].

The concept of the state of a dynamic system refers to a minimum set of state variables, which describes the system and its response to any given set of inputs completely. A state-determined system model has the characteristic that a mathematical description of the system in terms of a minimum set of variables $x_i(t), i = 1, \ldots, n$ together with the knowledge of those variables at an initial time t_0 and the system inputs for time $t \geq t_0$, are necessary to predict the future system state and outputs for all time. The state variables are an internal description of the system which completely characterize the system state at any time and from which any output variables $y_i(t)$ may be computed. There is no unique set of state variables that describe any given system. Many different sets of variables may be selected, but for a given system the order n is unique and is independent of the particular set of state variables chosen. The description can be chosen in terms of variables which are measurable or as indirect variables. The important point is that any set of state variables must provide a complete description of the system. Consider SISO-system we only had a single LODE. But considering MIMO-systems, the system is expressed as a set of n coupled first-order ODE, known as state equations, which in general case look like

$$\begin{aligned} \dot{x}_1 &= f_1\left(\vec{x}, \vec{u}, t\right) \\ \dot{x}_2 &= f_2\left(\vec{x}, \vec{u}, t\right) \\ &\vdots \\ \dot{x}_n &= f_n\left(\vec{x}, \vec{u}, t\right) \end{aligned} \tag{3.3}$$

where the $f_i\left(\vec{x}, \vec{u}, t\right)$ may be in general nonlinear, time varying functions of the state variables, the system inputs and time. A way to simplify this, is to write down a state vector $\vec{x}(t) = [x_1(t), x_2(t), \ldots, x_n(t)]^T$ and an input vector $\vec{u}(t) = [u_1(t), u_2(t), \ldots, u_r(t)]^T$. The system state at any instant time may be interpreted as a point in an n-dimensional state space, and the dynamic state response can be interpreted as a path in this state space.

With vector notation, (3.3) can be written as

$$\dot{\vec{x}}(t) = \vec{f}(\vec{x}, \vec{u}, t) \tag{3.4}$$

where $\vec{f}(\vec{x}, \vec{u}, t)$ is a vector function. Since we are still considering only the case of LTI-Systems, (3.4) becomes a set of n coupled first-order LODEs with constant coefficients

$$\begin{array}{rccccccc} \dot{x}_1 = & a_{11}x_1 + & \ldots + & a_{1n}x_n + & b_{11}u_1 + & \ldots + & b_{1r}u_r \\ \dot{x}_2 = & a_{21}x_1 + & \ldots + & a_{2n}x_n + & b_{21}u_1 + & \ldots + & b_{2r}u_r \\ \vdots & & \vdots & & & & \\ \dot{x}_n = & a_{n1}x_1 + & \ldots + & a_{nn}x_n + & b_{n1}u_1 + & \ldots + & b_{nr}u_r \end{array} \tag{3.5}$$

Equation (3.5) can be written into a compact matrix form

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & & & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b_{11} & \ldots & b_{1r} \\ b_{21} & \ldots & b_{2r} \\ \vdots & & \vdots \\ b_{n1} & \ldots & b_{nr} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_r \end{bmatrix} \tag{3.6}$$

which can be summarized as

$$\dot{\vec{x}} = \mathbf{A}\vec{x} + \mathbf{B}\vec{u} \tag{3.7}$$

where $\mathbf{A}$ is the state matrix and $\mathbf{B}$ the input matrix. The system output is defined to be any system variable of interest. An important property of the linear state equation description is that all system variables may be represented as a linear combination of the state variables x_i and the system input u_i. An arbitrary output variable in a system of order n with r inputs may be written as

$$y(t) = c_1x_1 + c_2x_2 + \ldots + c_nx_n + d_1u_1 + \ldots + d_ru_r \tag{3.8}$$

where c_i and d_i are constants. For a total amount of m output variables, we yield m equations

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \ldots & c_{1n} \\ c_{21} & c_{22} & \ldots & c_{2n} \\ \vdots & & & \vdots \\ c_{m1} & c_{n2} & \ldots & c_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} d_{11} & \ldots & d_{1r} \\ d_{21} & \ldots & d_{2r} \\ \vdots & & \vdots \\ d_{n1} & \ldots & d_{mr} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_r \end{bmatrix} \tag{3.9}$$

and (3.9) can be written in the compact form

$$\vec{y} = \mathbf{C}\vec{x} + \mathbf{D}\vec{u} \tag{3.10}$$

where $\mathbf{C}$ is the output matrix and $\mathbf{D}$ the feedthrough matrix. The equations (3.7) and (3.10) are the key equations in the concept of state space representation. The matrices $\mathbf{A}$ and $\mathbf{B}$ represent the properties of the system and are determined by the system structure and elements. The output matrices $\mathbf{C}$ and $\mathbf{D}$ are chosen to derive the wanted output variables. If we consider the case of a SISO-system, both methods, transferfunction and state-space, are equivalent and can be transformed into each other. Even if we talk about MIMO-systems, a matrix-transferfunction can be derived, where each element is an individual scalar transferfunction relating a given component of the output $\vec{Y}(s)$ to a component of the input $\vec{U}(s)$. The matrix-transferfunction can be calculated through

$$\mathbf{H}(s) = \frac{(\mathbf{C}\ adj(s\mathbf{I} - \mathbf{A})) + \det[s\mathbf{I} - \mathbf{A}]\mathbf{D}}{\det[s\mathbf{I} - \mathbf{A}]} \tag{3.11}$$

with $\mathbf{I}$ as unity matrix. An important issue is, that all the entries $H_{ij}(s)$ of the matrix-transferfunction in (3.11) have the same denominator. This means, all input-output ODE's for the system will have the same characteristic polynomial. This will be under investigation in the next section, when we discuss the stability of a system.

Stability considerations

First, we will discuss stability in terms of the transferfunction. A transferfunction can be written as

$$H(s) = \frac{\sum_{i=0}^{i=n} a_i s^i}{\sum_{i=0}^{i=m} b_i s^i}$$

It is often convenient to factorize the polynomials in the numerator and denominator and to write the transferfunction in terms of those factors:

$$H(s) = \frac{N(s)}{D(s)} = \left|\frac{a_m}{b_n}\right| \cdot \frac{\prod_{k=1}^{M}(s - z_k)}{\prod_{i=1}^{N}(s - p_i)} \tag{3.12}$$

This is, as mentioned in the section above, concerning equation (3.11), the reason why it is important that the matrix entries will have the same denominator. The root of N(s) equal zero is called zero, while the root of D(s) equal zero is called pole. If the location of the zeros and poles are known, you can immediately say whether the system will be stable or not.

Zeros means, that signals of these frequencies are completely rejected by the system. They are of major interest, because they can cancel out poles. System instabilities are represented through poles in the polynomial at these frequencies. This means, that the output signal will grow till infinity.

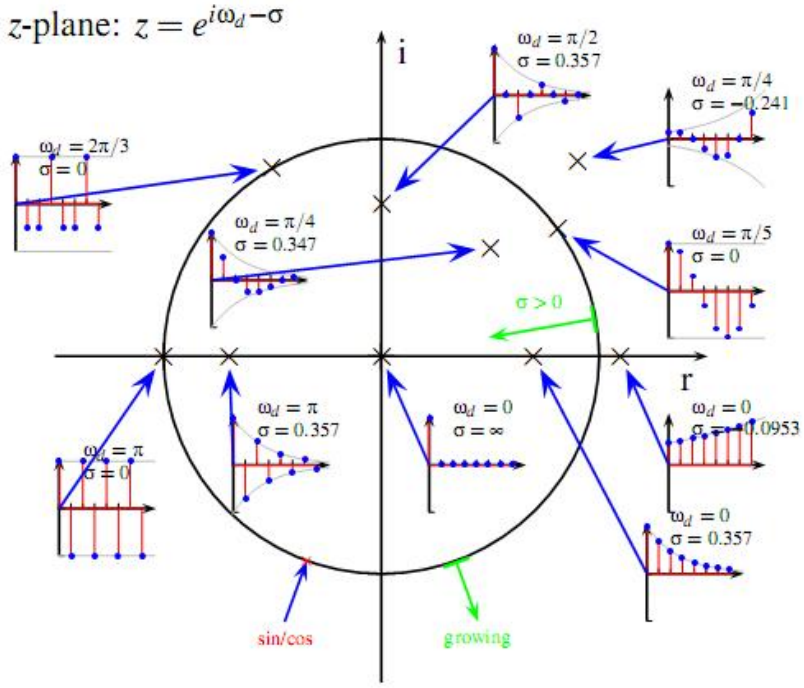


Figure 3.4: Depending on the poles and zeros of your transfer function, you can identify regions of stability and instability [21].

By knowing the zeros and poles of the polynomial, the system behavior can be foreseen under known input (see Fig. 3.4). If the real part of the pole will be in the left half plane of the z-plane one will obtain a stable system and vice versa one has an unstable system if the real part lies in the right half plane. Different techniques can be used to derive a statement about the stability of the system. For example

- Hurwitz criterion
- Direct calculations like root locus, Bode Plot, Nyquist criterion
- Simulation (often used for non-linear systems)

When talking about the stability of the state-space-representation, we have to look at equation (3.11). The denominator $\det[s\mathbf{I} - \mathbf{A}]$ is the characteristic polynomial. If the eigenvalues solve the relation $\Re(\lambda_i) \leq 0 \; \forall i$, the system state is stable. Another stability is the BIBO-stability. This means, that if the input is bounded and below a certain value, then the output will also be bounded.

3.1.3 Vector sum

The vector sum is defined as

$$\vec{V}_{sum} \cos(\phi_{sum}) = \sum_i \vec{V}_{i,cal} \cos(\phi_{i,cal}) = \sum_i \begin{pmatrix} a & b \\ -b & a \end{pmatrix} \vec{V}_{i,cav} \cos(\phi_{i,cav}) \tag{3.13}$$

The effective accelerating gradient is the sum of the calibrated individual cavity fields. The calibrated field is the field at the pick up antenna inside the cavity which is rotated and scaled. The scaling is necessary due to different attenuation of the signals. The

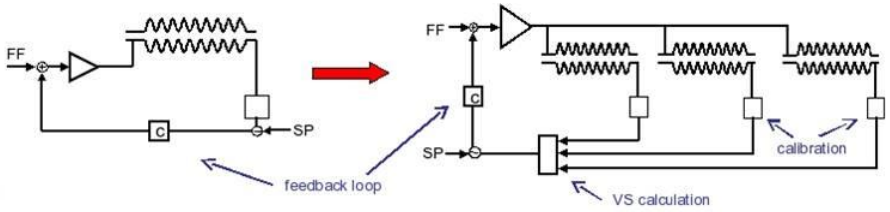


Figure 3.5: Driving multiple cavities with one klystron will make it impossible to regulate individual signal. This situation requires the control of the so called vector sum.

different cable lengths lead to individual phase drift of the signals which needs to be compensated by rotating the signals in the complex plane. Another reason for the rotation is explained in section 3.1.4. The advantages for the vector sum are the cost savings and the reduced maintenance since less units need to be controlled. The disadvantages are that the calibration is quite a challenge (see 3.2.2) and that individual limits can not be considered as easy in comparison to single driven cavities. Also bypassing of individual cavities can be difficult if it becomes necessary during operation.

3.1.4 IQ-sampling

In general, the so called IQ-sampling, can be favorable in comparison to the Amplitude-Phase-sampling, when large errors in amplitude and phase are expected. The reason for this is that you can control amplitude, phase and even frequency with two complete similar feedback paths which can act in parallel onto each component. The terminology 'IQ' comes from the origin of the representation.
Any sinusoidal signal

$$y(t) = A \cdot \sin(\omega t + \varphi_0) \tag{3.14}$$

can be represented as a phasor, which is a similar concept to a vector, in relation to a local oscillator phasor in the complex plane (see Fig. 3.6). If the frequency is positive, the phasor will rotate anti-counterclockwise in the complex plane. By using basic trigonometric relations, we can decompose (3.14) into two parts

$$y(t) = A \cdot \sin(\omega t + \varphi_0) \tag{3.15}$$

$$\Leftrightarrow y(t) = \underbrace{A\cos(\varphi_0)}_{:=I} \sin(\omega t) + \underbrace{A\sin(\varphi_0)}_{:=Q} \cos(\omega t) \tag{3.16}$$

$$y(t) = I \cdot \sin(\omega t) + Q \cdot \cos(\omega t) \tag{3.17}$$

Here you see that the so called in-phase (I) component has a sine-term, while the

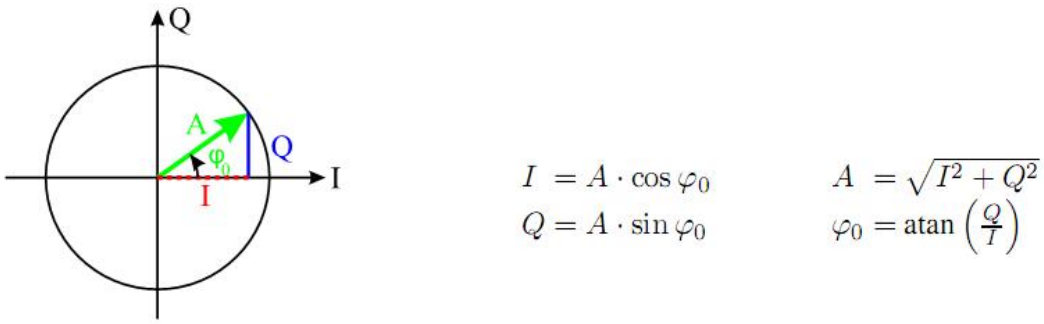


Figure 3.6: In complex representation and do a coordinate transformation we use the real and imaginary part, the so called I- and Q-Part, instead of the Amplitude and Phase of a signal. [36]

quadrature-phase (Q) component has a cosine term, so it is phase shifted by 90 degrees to the I component.

Usually, for feedback and controller loops IQ is used while for a graphical display of the signal A-ϕ representation is used, since it is more intuitive for our imagination. An ADC sampling of the signal will only allow us to measure one of the components at a time. But with a simple method we can still derive all needed informations. If the IQ-phasor is measured at a certain, well defined time, this phasor can be compared to a reference phasor, the setpoint, to see if any changes in amplitude or phase have happened. Since a time advance also means a phase advance, a rotation of the phasor needs to be done. This rotation is done by the so called loop-rotation matrix, which need to compensate the phase shift for the comparison with the setpoint, but can also compensate phase drifts due to different cable lengths and scale the phasor, if needed due to different attenuations (see equation 3.13). By having a phase shift of 90 degrees between each sample, which means that the sampling frequency needs to be 4 times the frequency of the sampled signal (see Fig. 3.7), you can assure that both components, I and Q, can be sampled and digitized within one period of the sampled signal. Of course, any relation between sampling frequency and signal frequency can be used, as long as the ratio is an integer, implying a phase shift of $\Delta\varphi = \frac{2\pi}{m}$ between two samples. With m=4, the loop rotation matrix will simplify in such a way, that it consists only of 0,1 and -1. This makes digital signal processing a lot faster than using linear approximation of the transcendental function sine and cosine but can also lead to problems (Chapter 6).

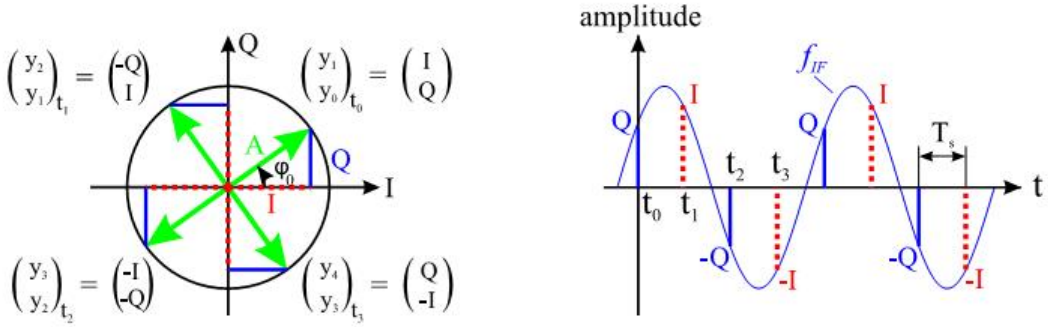


Figure 3.7: Principle of the sampling scheme at FLASH for IQ control [36]. Left: the phaseshift is shown which need to be compensated by the loop-rotation-matrix. Right: The sampling frequency is 4 times the sampled signal frequency. This guaranties a sampling scheme of Q, I, -Q, -I and will bring every information we need.

3.1.5 Digitization and preprocessing

In this section, some basic techniques and concepts used for preprocessing and digitization of an analog signal will be explained. The principle of an Analog to Digital Converter (ADC) is explained Fig. 3.8. Digitization is a process which maps a continuous signal onto a sample

$$s(t) \rightarrow s_n := s\,[n] := s(nT)\ ;\ n \in \mathbb{Z} \vee \mathbb{N}_0$$

where s is seen as constant over the time interval. T is the period of the sampling frequency $T = 1/f_s$. Two effects have to be considered:

1. time discretization
2. amplitude quantization

Amplitude quantization can be quite precise, when choosing high resolution ADCs up to 20 bits, but this means a decrease in your sampling frequency.

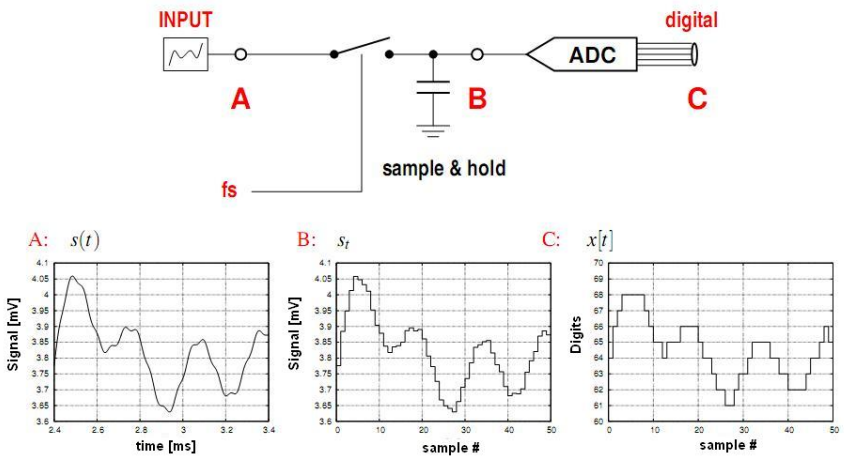


Figure 3.8: Samples are taken from the analog input signal (A). The time discretization is done with the sampling frequency f_s. The voltage is stored in a sample-and-hold device (B) which can be a simple capacitor. Finally the voltage across the capacitor is converted into a digital number (C), usually represented by n bits of digital logic signals [21].

Time discretization can be influenced by clock jitter (phase drift of the trigger signal) and this effect will become stronger at higher frequencies. This is the reason why we are using down- and upconverters to switch between different frequencies in a signal processing chain. For example, at FLASH the 1.3 GHz driving signal is downconverted to 250 kHz as an intermediate frequency and will be sampled with 1 MHz or at the vertical test stand the driving signal is converted to 9 MHz and sampled with 81 MHz. A converter usually consists of an rf-mixer and a filter at the mixer output. First the rf-mixer is explained (Fig. 3.9). When mixing two sinusoidal signals, the output signal can be calculated to

$$y(t) = y_{RF}(t) \cdot y_L O(t) = \frac{1}{2} A_{LO} A_{RF} \cdot \left(\underbrace{(sin\left[(\omega_{RF} - \omega_{LO})\, t + (\varphi_{RF} - \varphi_{LO})\right])}_{lower\ \ sideband} + \underbrace{(sin\left[(\omega_{RF} + \omega_{LO})\, t + (\varphi_{RF} + \varphi_{LO})\right])}_{upper\ \ sideband} \right) \tag{3.18}$$

This means, that the input signal will be mirrored onto two frequencies, $(\omega_{RF} - \omega_{LO})$ and $(\omega_{RF} + \omega_{LO})$. Applying the IF as an input signal you will have an upconversion, if the filter is a highpass. If the input signal is the RF signal and the filter a lowpass you will have a down-conversion (Fig. 3.10)

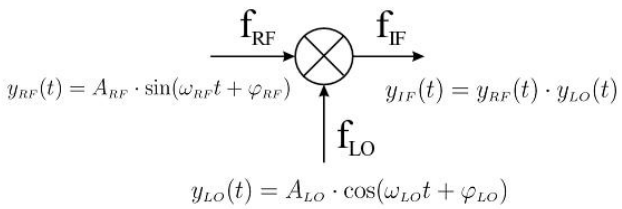


Figure 3.9: The principle of an ideal mixer is shown in this graph. The original signal with frequency f_{RF} is mixed with a local oscillator signal with a frequency f_{LO}. This results in an intermediate signal with frequency f_{IF}.

When the LO frequency and phase are constant in time, all information of the RF signal is transfered to the IF signal. Until now, an ideal mixer has been considered. A real mixer will generate higher harmonics since they are non-linear devices. This means, that the spectrum of the output will have signals at $m \cdot f_{RF} \pm n \cdot f_{LO}$. Fig. 3.11. shows the principle output spectrum.

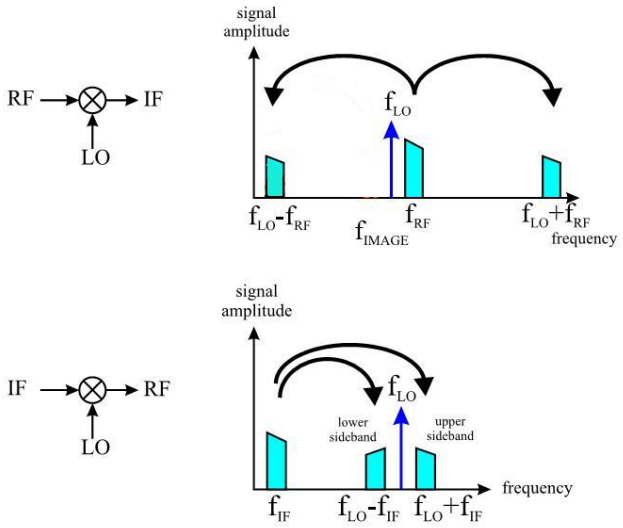


Figure 3.10: The upper plot shows a down-converter, the lower plot an up-converter. Both signals are mixed with an ideal mixer [36].

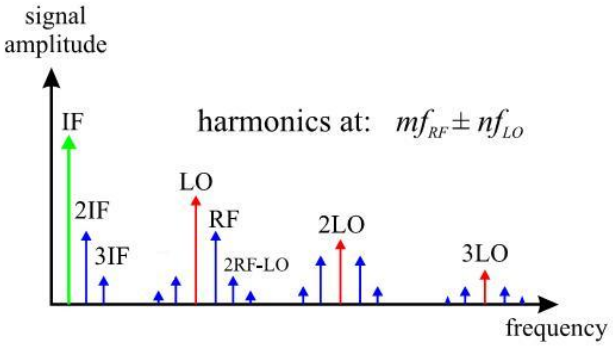


Figure 3.11: A real mixer will have sidebands, due to non-linearities. These sidebands will be at higher/lower harmonics of the difference and sum of the input frequencies [36].

Now we know how the rf signal is converted to a lower frequency to assure a proper sampling and how the digitization works. But what does 'proper' sampling mean? This question leads to the important Nyquist-Shannon-Theorem, published in 1940. Proper sampling means, that we can reconstruct exactly the analog signal from our samples. It should be clear, if we choose (infinite) high sampling frequencies and sample every point of our analog signal this goal can be achieved. Here we have the problem (besides the issue to generate infinite high frequencies) that we will get a huge amount of data. We want to choose the lowest frequency which is needed, to assure proper sampling. The solution to this problems is the mentioned Nyquist-Shannon-Theorem, which can be written as:

> 'A continuous signal can be properly sampled if it does not contain frequency components above $f_{crit} = \frac{f_s}{2}$.'

Frequency components in the signal which are larger than the Nyquist frequency ($f \geq f_{crit}$) will be 'aliased' to a mirror frequency $f^* = f_{crit} - f$ where f_s is the so called Nyquist-frequency. This means, that the sampling frequency needs to be at least twice the frequency of interest. Some examples are shown in Fig. 3.12.

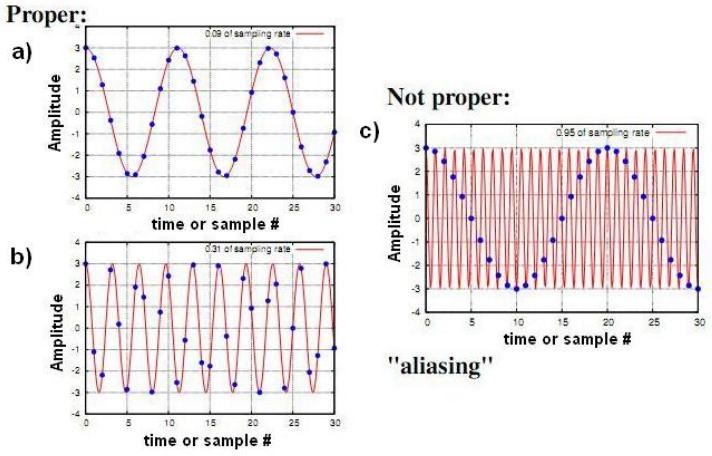


Figure 3.12: (a) The ratio of sampled signal frequency and sampling frequency is 0.09 and 0.31, respectively (b). This results in a proper sampling and the signal will be sampled correctly. (c) The ratio is 0.95 which means, that the sampling frequency is nearly equal to the sampled frequency. Therefore, a shift in the frequency will occur which is called aliasing [21].

3.2 Accelerator as a system

3.2.1 What we want to achieve and what we can control

We can consider a cavity as a system, where the ODE will be derived in section 3.3.1. This allows, that the principles of control theory can be used to describe and control an accelerator (see Fig. 3.13). For details what requirements we want to achieve I refer to section 4.1. In this section, a principle sketch of the variables we have to control and the turning knobs to steer the system will be discussed. Important for the beam is a high

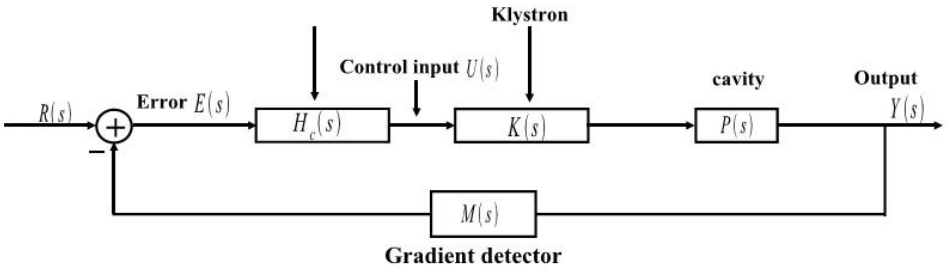


Figure 3.13: General layout of amplitude feedback of an accelerator (compare Fig.3.3).

stability of the accelerating gradient and the phase. Fluctuations in the field strength would lead to fluctuations in the beam energy. And if the phase of the accelerating field is not synchronized with the beam phase, high energy conversion to the beam can not be achieved. Both, amplitude and phase of the accelerating gradient, are the parameters we have to control. (Fig. 3.14)
Considering the cavity, a good timing of beam arrival and cavity filling is needed, since we want to operate close to the quenching limit. The beam induced voltage will compensate the additional power sent to the cavity, which will add up to a constant field strength if beam current and generator current matched in their amount and time. If not, additional power could be send to the cavity and will raise the field strength which can lead to a quench. And we have to make sure, that the resonance criterion is fulfilled, which means that the driving signal from the klystron matches the frequency and the phase of the accelerating field. In Fig. 3.15 the time structure of the beam at FLASH is given. The frequency itself can be tuned at the klystron, but the phase for each cavity needs to be adjusted just in front of the cavity (Fig. 3.16). This can be done by motorized input couplers and phase shifter to steer to a wished state.

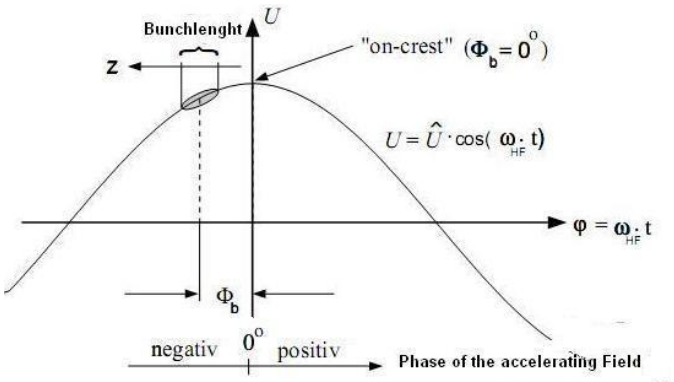


Figure 3.14: Description of the parameters we need to concern. The phase of the bunch Φ_b, the phase φ and amplitude U of the accelerating field [11].

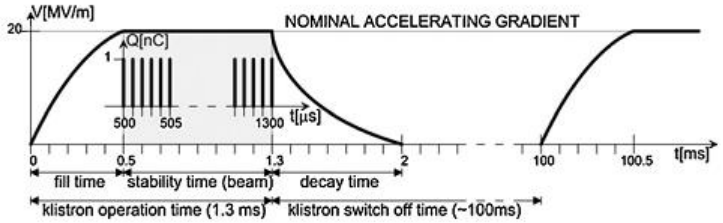


Figure 3.15: Time structure at FLASH. The filling time is about 500 ms. After this, the steady state of the cavity will be reached and the beam is injected. The beam pulse length will vary, but the flattop has a length of 800 ms. Then the rf signal is turned off and the field inside the cavity will decay. The pulse structure can have a different amount of bunches, charge per bunch and pulse repetition time [3].

The ratio of the power send to an individual cavity with respect to the klystron power is usually fixed and can not be tuned. The coupling β as introduced in chapter 2 is correlated to the power we need to fill the cavity to its equilibrium state and to the time we need for that. This coupling, which can be controlled via antenna penetration depth is linked to the Q_L, where Q_L is the value we want to regulate [2]. Of course the phase of the driving signal is also steered, but the phase will be set once and any fast changes in the phase will be compensated by the fast piezo-tuners attached to the cavity instead of the slower phase-shifter in the transmission line. The Q_L again has influence on the cavity bandwidth and to the decay time of the cavity. A high Q_L would be desirable to

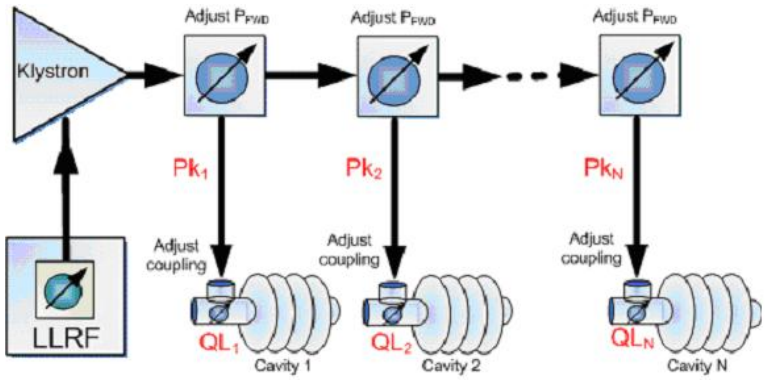


Figure 3.16: The filling time is controlled by the Q_L which in turn can be controlled by the coupling of the driving signal to the cavity. The input power coupling and the phase of the driving signal can be controlled via 3-stub-tuner, motorized antenna or phase-shifter.

decrease the cryogenic cost and gain a higher field stability. But a high Q_L would also mean longer filling time, so the pulse to pulse spacing needs to increase and the bandwidth of the resonance would become narrower. A too narrow resonance amplitude[3] would result in a small window for the controller to stay on resonance. Therefore the lock on the peak could be lost by small changes of the resonance frequency itself. But if we have a wide bandwidth to ensure the lock no matter what disturbances arise, then the decay time is too low, the field would decay too fast and the cavity would need more

[2] The controller will translate a change in Q_L into an antenna depth variation. This is an example for the state variables which describe a system. There are several sets which can be used. They can have an engineering or physical meaning but they can be mapped onto each other.

[3] e.g. $\omega_0 = 1.3$ GHz, $Q_L \approx 10^{9-10} \rightarrow \omega_{1/2} \approx 0.1$ - 1 Hz

power to keep a constant field. The influence of the coupling to the bandwidth and phase of the resonance curve is shown in Fig. 3.17. The coupling and hereby Q_L can be derived

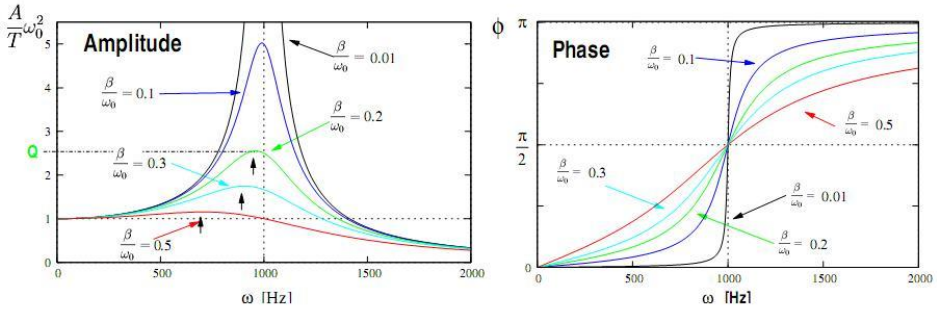


Figure 3.17: In this plot, different bandwidths for different couplings and the phase change is shown ($Q_L \propto \beta$, β is not the coupling but the damping in this case) [21].

by sampling and analyzing the reflected Power (as shown in chapter 2). Examples of the influence of the coupling to the reflected power are shown in Fig. 3.18. The basic set of

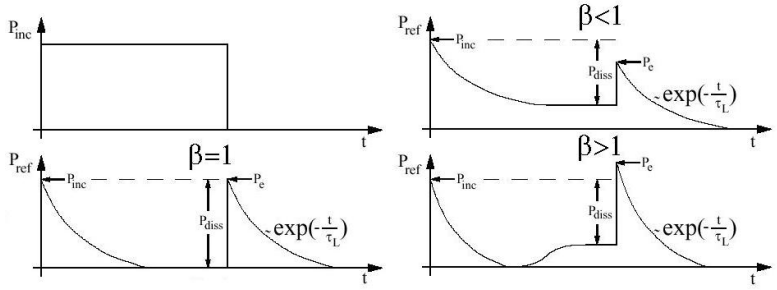


Figure 3.18: Different couplings lead to different reflected waves at the end of the transmission line.

values we can steer to maintain a constant accelerating gradient is the klystron power and the Q_L. Of course, other infrastructure parameters, like helium-pressure inside the modules, injector gun parameters, are also relevant but are not taken into account.

3.2.2 Noise sources and disturbances

Now it is evident that we need to adjust the frequency and phase of the driving signal from the klystron and the Q_L. These parameters are our steering knobs besides the amplitude of the driving signal, which we need to consider when we plan our control system. In the following we will discuss, which disturbances arise during acceleration and what noises need to be compensated to reach high energy stability of the beam. One has to take the noise sources and disturbances into account the realize a correct model of our accelerator, which we want to describe from a mathematical point of view.

HOM excitations

In chapter 2 we could see, that higher order modes (HOMs) exist in the spectra of coupled accelerating cells. These HOMs are unwanted, since they can deflect, interrupt or decelerate the beam. They can be induced by the electric field of the bunch passing through the cavity (so called wakefield) and should be only from the set of TM-modes. But this is only true for a pillbox cavity, since beam pipes can create different field distributions and the bunch is not always traveling along the cavity symmetry axis but will do some oscillations (Fig. 3.19). These off axis passages can induce TE-modes or hybrid modes like TE/TM and TM/TE.

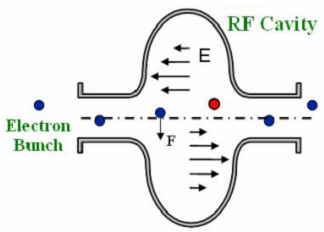


Figure 3.19: HOMs can arise during operation due to beam - cavity interaction. This will lead to a different field pattern which will be seen by the subsequent bunch [21].

HOMs can be damped in a cavity when attaching so called HOM-couplers to it. They are output-couplers which will extract all unwanted modes inside the cavity in a specific band (notchfilter) but will leave the π-mode unchanged. HOMs are one of the reasons why the number of cells should not be too large, since they can be trapped inside a cavity and their electric or magnetic field in the end cell is to weak to be coupled out.

Microphonics

An important noise source are 'microphonics'. They are generated by vibration in the helium pumps, ground vibrations or more general by any mechanical vibration in or near the machine. It is random generated noise with a Gaussian distribution, which can detune the resonance frequency, e.g. up to 10 Hz at FLASH. They can not be reduced but suppressed with a high gain in the feedback loop. On the other hand, a too high gain can cause oscillations in the feedback system, where this limits the gain used in operation. Nevertheless, microphonics are an important issue when choosing the Q_L for the cavities, which can be seen in Fig. 3.20.

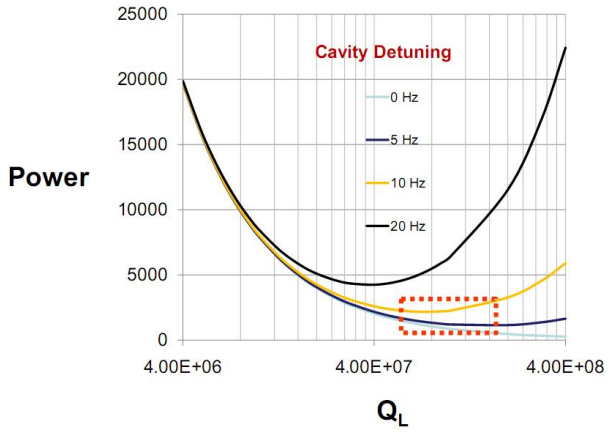


Figure 3.20: This plot shows that the maximum frequency detuning, caused by microphonics, even influences the Q_L. To assure that the cavity will stay on resonance with the driving signal in presence of the microphonics, the bandwidth needs to be broadened [35]. Otherwise the klystron power needs to be increased.

Microphonics can play an essential role since we want to push Q_L to new frontiers for future accelerators.

Lorentz force

A dominant and repetitive error source at accelerators is the so called Lorentz force detuning. The rf-pulse sent to the cavity and the time-dependent induced field by the electron bunch will create attractive forces at the iris and repulsive force at the equatorial area of the cavity.

This results in mechanical oscillations of the cavity and will detune the cavity with each pulse/bunch. Fig. 3.21 shows such a detuning. The maximum change of the resonance peak depends on the accelerating gradient. To compensate these deformations, fast

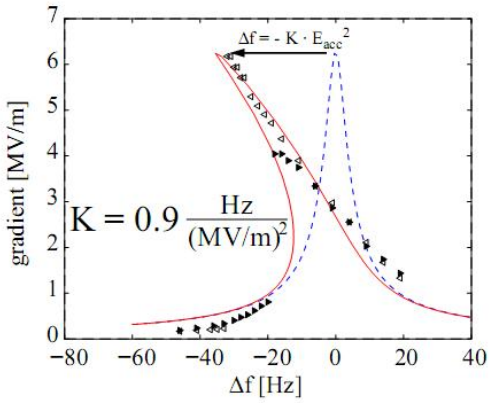


Figure 3.21: Lorentz force will create problems matching the driving signal to the cavity conditions during filling time [11].

piezo-tuners are attached to the cavities. Two sets are assembled to each cavity. One is working as a sensor, the other as an actuator [41]. The Lorentz force detuning can also be regulated through feedforward or adaptive feedforward since we are dealing with repetitive and predictable errors.

Beam loading

One of the main topics in cavity control systems is the so called beam loading. A single bunch passing through the cavity induces a voltage which is out of phase with the accelerating gradient and will lower the effective field for the next passing bunch. This effect can be neglected if the bunch repetition rate is low in comparison to the cavity time constant. But with intense long bunch trains this effect will become more important. To compensate beam loading, feedforward is applied, which means raising the amplitude of the driving signal for the duration of the beam in the cavity. A correction signal can be send to the klystron before the beam reaches the cavity. Because the pulse structure is loaded to the rf gun before the beam is generated, the information can be send to the feedforward controller and is fed to the klystron.

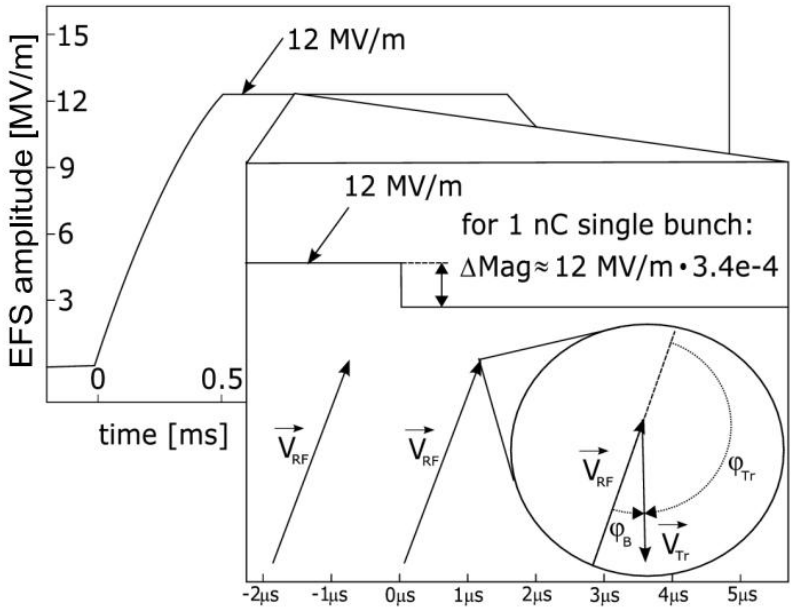


Figure 3.22: Beam Loading occurs because the bunch traveling through the cavity generates an electric field which will decrease the cavity field (the factor by which the gradient decreases depends on the bunch charge, cavity bandwidth and the phase difference of bunch phase and gradient phase) [40].

3.3 Cavity behavior

In this section, an ODE describing the cavity will be derived and transfered into state space representation. All physical quantities introduced in this chapter are equal to those in chapter 2. Only one new definition will be needed. As has been shown in chapter 2, cell to cell coupling is possible via magnetic and/or electric field and the system can be described in terms of an equivalent LCR circuit. Modeling an LCR circuit needs to define a resistor R, in which the same power is dissipated as in the cavity

$$P_{diss} = \frac{1}{2}\frac{V_{cav}^2}{R}$$

and this leads to a useful relation[4]

$$R = \frac{1}{2} \cdot \left(\frac{r}{Q}\right) \cdot Q_0$$

3.3.1 Description of a cavity via ODE

In Fig. 3.23, the driven LCR-circuit which models a single cavity is sketched. The rf power source is a klystron. It can be modeled as a current source. The input coupling is a lossless transformer and has a ratio of 1:N, therefore the impedance relation is

$$Z_{cav} = N^2 \cdot Z_0. \tag{3.19}$$

A circulator is inserted into the transmission line to absorb any reflected wave, which occurs at the cavity, since the klystron could be destroyed.
If we now use basic formulas from circuit theory

$$W = \frac{1}{2}CV_0^2 \tag{3.20}$$

$$P_{diss} = \frac{V_0^2}{2R} \tag{3.21}$$

the quality factor Q_0 can be written as

$$Q_0 = \frac{2\pi}{T} \cdot \frac{\frac{1}{2}CV_0^2}{\frac{1}{2}\frac{V_0^2}{R}} \tag{3.22}$$

[4]Here I refer back to chapter 2.

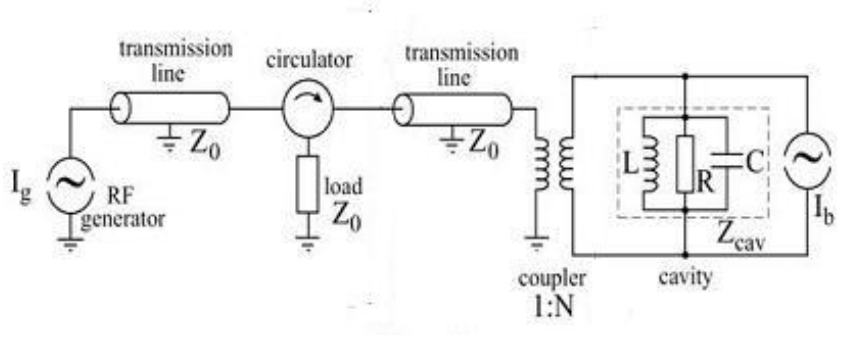


Figure 3.23: The Klystron - cavity System can be seen as a electric circuit with specific loads, inductance and resistance. The generator current and the beam current need to be matched through the system [14].

where V_0 is the amplitude of the oscillating voltage and T the time period. Since the resonance frequency of an undamped LC-circuit is $\omega_0 = \frac{1}{\sqrt{LC}}$, (3.22) can be rewritten to

$$Q_0 = \omega_0 RC = \frac{R}{L\omega_0} = \frac{\omega_0 W}{P_{diss}}. \tag{3.23}$$

Instead of the transformation ratio 1:N, we will introduce the coupling constant β, to give this variable another new interpretation. It is defined, in circuit theory, as the ratio of the resistor R in the LCR circuit to the transformed external load

$$\beta = \frac{R}{N^2 Z_0} \;\rightarrow\; N = \sqrt{\frac{R}{\beta Z_0}} \tag{3.24}$$

with this definition we can now describe the external load Z_{cav} and the parallel cavity resistor R in a simple way. With the definition of the loaded shunt impedance R_L

$$\frac{1}{R_L} = \frac{1}{R} + \frac{1}{Z_{cav}} \tag{3.25}$$

$$\underbrace{\rightarrow}_{(3.24)} R_L = \frac{R}{1+\beta} \tag{3.26}$$

$$\underbrace{\leftrightarrow}_{def.\ in\ 3.3} R_L = \frac{Q_L}{Q_0} R. \tag{3.27}$$

This also leads to the relation

$$\frac{R}{Q_0} = \omega_0 L = \frac{1}{\omega_0 C} = \sqrt{\frac{L}{C}}. \tag{3.28}$$

Now describe the driven LCR circuit with Kirchhoff's rule

$$I_C + I_R + I_L = I \tag{3.29}$$

and insert the formulas

$$\dot{I}_L = \frac{V}{L}$$
$$\dot{I}_R = \frac{\dot{V}}{R_L}$$
$$\dot{I}_C = C\ddot{V}$$

replacing the inductance and capacitance by relations derived above, one obtains the differential equation

$$\ddot{V}(t) + \frac{1}{R_L C}\dot{V}(t) + \frac{1}{LC}V(t) = \frac{1}{C}\dot{I}(t) \tag{3.30}$$

$$\Leftrightarrow \ddot{V}(t) + \frac{\omega_0}{Q_L}\dot{V}(t) + \omega_0^2 V(t) = \frac{\omega_0 R_L}{Q_L}\dot{I}(t) \tag{3.31}$$

3.3.2 Description of a pulsed cavity via state space representation

From ODE to LODE to matrix equation

Equation (3.31) is used as a starting point to discuss the cavity behavior with transient behavior, which means turn rf on/off and beam will be injected after some delay. We will limit ourselves to the π mode. Begin with equation 3.31, but in a vector-representation, since we want to switch to complex equations

$$\ddot{\vec{V}}(t) + \frac{\omega_0}{Q_L}\dot{\vec{V}}(t) + \omega_0^2 \vec{V}(t) = \frac{\omega_0 R_L}{Q_L}\dot{\vec{I}}(t). \tag{3.32}$$

Because Q_L is high, we have a weakly damped system. This means that the resonance frequency is only slightly changed [5]

$$\omega_{res} = \omega_0 \sqrt{1 - \frac{1}{4Q_L^2}} \approx \omega_0$$

[5] 0.024 Hz with ILC design value of Q_L

The driving current has a time dependence $e^{i\omega t}$ and the Fourier component of the beam current of the pulsed beam are harmonic to it. Therefore, we can separate the fast rf oscillation from the slowly changing amplitudes and phases or from real and imaginary parts of the field vector.

$$\vec{V}(t) = (V_r(t) + iV_i(t)) \cdot e^{i\omega_{rf}t}$$
$$\vec{I}(t) = (I_r(t) + iI_i(t)) \cdot e^{i\omega_{rf}t}$$

When we now insert these information into the ODE (3.32), we obtain a linear ODE

$$\dot{V}_r + \omega_{1/2}V_r + \Delta\omega V_i = R_L\omega_{1/2}I_r$$

$$\dot{V}_i + \omega_{1/2}V_i + \Delta\omega V_r = R_L\omega_{1/2}I_i$$

Here $\omega_{1/2} = \frac{\omega_0}{2Q_L}$ is the cavity bandwidth and $\Delta\omega = \omega_0 - \omega_{rf}$ the detuning of the cavity. This LODE can be written in a matrix form:

$$\frac{d}{dt}\begin{bmatrix} V_r \\ V_i \end{bmatrix} = \begin{bmatrix} -\omega_{1/2} & -\Delta\omega \\ \Delta\omega & \omega_{1/2} \end{bmatrix}\begin{bmatrix} V_r \\ V_i \end{bmatrix} + \begin{bmatrix} R_L\omega_{1/2} & 0 \\ 0 & R_L\omega_{1/2} \end{bmatrix}\begin{bmatrix} I_r \\ I_i \end{bmatrix}. \tag{3.33}$$

State space and stability of the cavity with feedback

The matrix equation (3.33) is the starting point to derive the state space representation. First, we have to apply the Laplace-transformation and solve for $U_r(s)$ and $U_i(s)$ (with $\vec{U}(s) = R_L\vec{I}(s)$)

$$\vec{Y}(s) = \begin{bmatrix} V_r(s) \\ V_i(s) \end{bmatrix} = \frac{\omega_{1/2}}{\Delta\omega^2 + (s+\omega_{1/2})^2}\begin{bmatrix} s+\omega_{1/2} & -\Delta\omega \\ \Delta\omega & s+\omega_{1/2} \end{bmatrix}\begin{bmatrix} R_LI_r(s) \\ R_LI_i(s) \end{bmatrix}. \tag{3.34}$$

Because the transfermatrix $\mathbf{H}$(s) is defined by $\vec{Y}(s) = \mathbf{H}(s) \cdot \vec{U}(s)$, the cavity transfermatrix is

$$\mathbf{H}_{cav}(s) = \frac{\omega_{1/2}}{\Delta\omega^2 + (s+\omega_{1/2})^2}\begin{bmatrix} s+\omega_{1/2} & -\Delta\omega \\ \Delta\omega & s+\omega_{1/2} \end{bmatrix} = \begin{bmatrix} H_{11}(s) & H_{12}(s) \\ H_{21}(s) & H_{22}(s) \end{bmatrix} \tag{3.35}$$

The off-diagonal elements describe the coupling between real and imaginary parts, when the cavity is detuned. If the cavity is on resonance, the real and imaginary parts are completely decoupled[6]. In Fig.3.24, Bode-plots of a detuned and tuned cavity are shown. The transfermatrix describes only one passband mode. For a short discussion of the

[6]Interestingly, life is also complex. It has real and imaginary parts, where both have to be taken into account when considering the value of life.

stability of the rf control system, one need to consider all passband modes. Assuming the gain is G=1, no latency in the feedback loop and adding the other passband modes, Fig.3.25 was derived. A phaseshift of 180 degrees at each resonance frequency comes from the coupling relations of the modes, since each fundamental mode from the passband has an electric field with opposite sign in the end cell (Fig. 2.17). The Nyquist-stability-criterion tells us, that if the amplitude at such a phase shift is higher than 0 dB, the system gets instable. The $\frac{1}{9}\pi$- to $\frac{7}{9}\pi$-mode will not need to be considered, since they are filtered out through the small bandwidth of the cavity and the small bandwidth of the klystron. But the $\frac{8}{9}\pi$-mode needs to be considered. The loop delay can have influence, whether the $\frac{8}{9}\pi$-mode will see positive or negative feedback [38], [37]. A relation between controller stability and loop-delay exists, which shows periodic structure. Until now, we were only discussing time continuous systems. If we switch to a digital controlled system, we should apply the z-transformation and not Laplace-transformation. This also leads to some artifacts, such as higher amplitudes due to aliasing in the system. The transfermatrix, which needs to be considered, is the product of the transfer matrices along the controller path. When taking these new assumptions into account, a stability map can be plotted for the feedback loop delay (Fig. 3.27). The sampling itself can also lead to a filtering. This is the reason why an upgrade to a sampling frequency of 54 MHz is planned at FLASH and the installation of a digital notchfilter in the feedback loop to get rid of any unwanted signal of the $\frac{8}{9}\pi$-mode. This would increase the field stability inside the cavity, since any sidebands limit the gain at FLASH, which should be as high as possible to suppress uncorrelated, random noise such as microphonics.

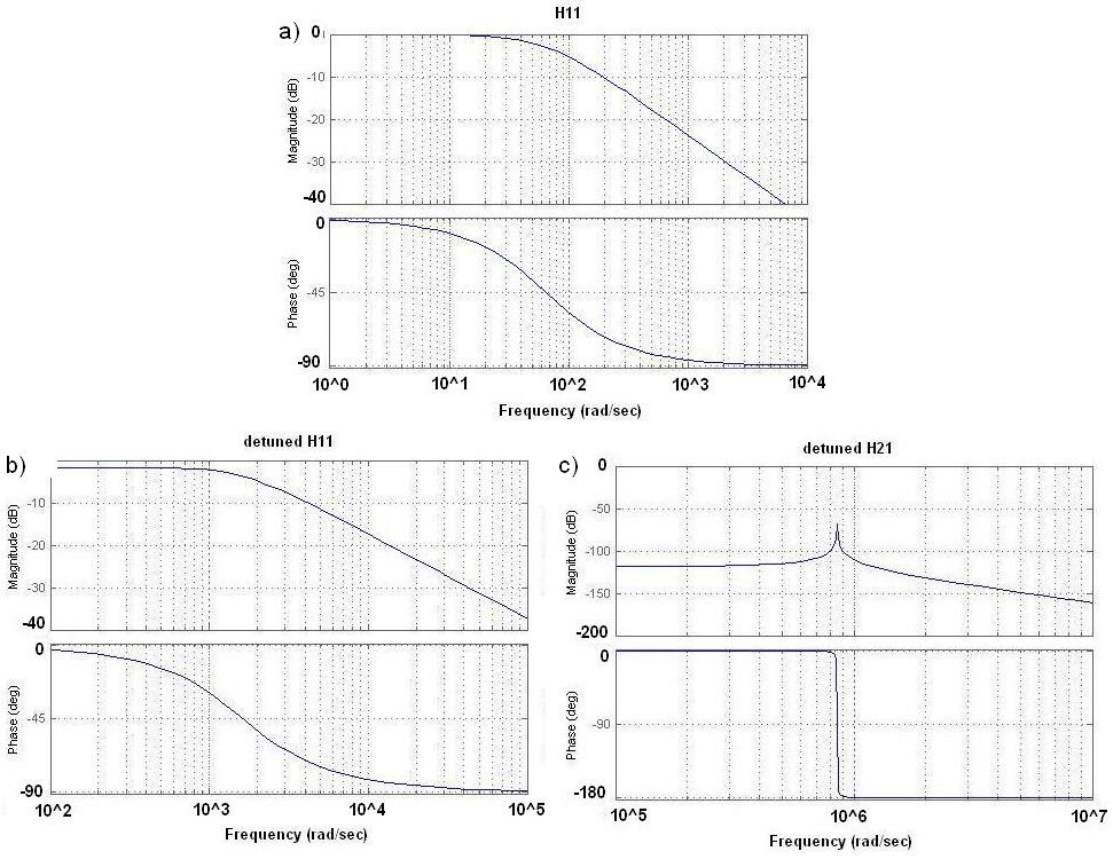


Figure 3.24: a) H_{11}, cavity on resonance. b) detuned H_{11}. c) detuned H_{21}. Only in a detuned cavity, H_{21} is non-zero. Bandwidth is 215 Hz for both plots, detuning by 100 Hz.

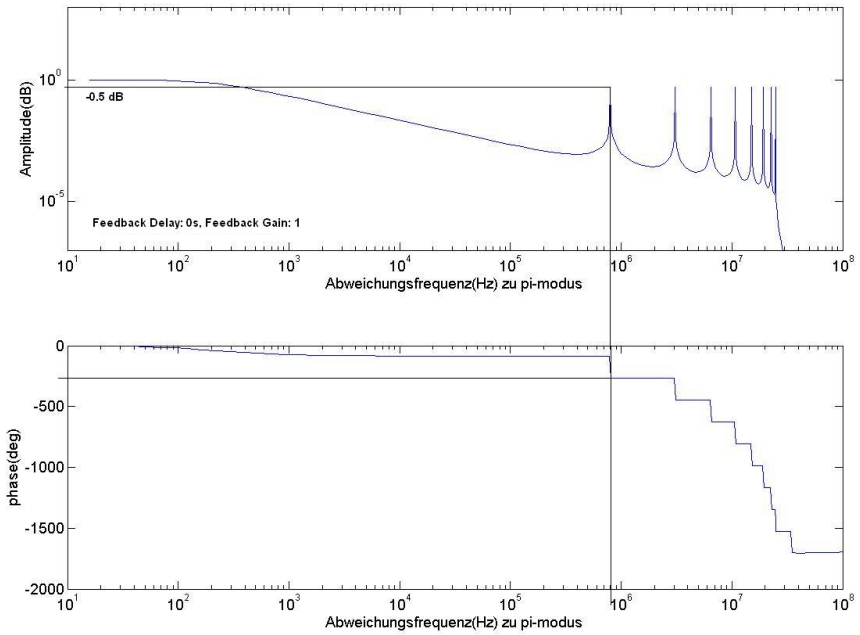


Figure 3.25: All passband modes, G=1, no loop delay. Nyquist-stability-criterion tells us, that if the amplitude at a certain frequency is higher than 0 dB at a phase shift of 180 degrees, the system can become unstable. This plot shows, that this could happen if we do not control the $\frac{8}{9}\pi$-mode.

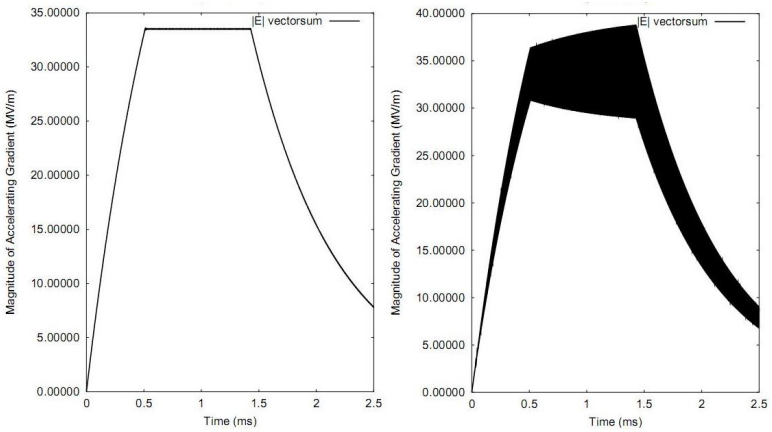


Figure 3.26: Left: no loop delay, right: delay is 2π. Strong oscillations on the Amplitude can be seen, which is the influence of the $\frac{8}{9}\pi$-mode. Therefore, considering loop-delay when planning your digital control is necessary. (see Fig. 3.27)

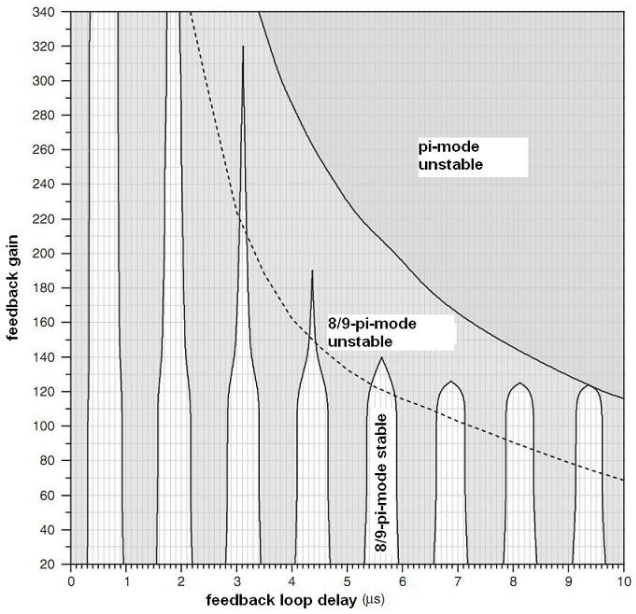


Figure 3.27: Stability chart for a single cavity. Simulation of a digital control loop. f_s =1 MHz, β_π- and $\beta_{\frac{8}{9}\pi}$-mode, $f_\pi - f_{\frac{8}{9}\pi} = 800$ kHz [37].

3.4 Driving multiple cavities with a single klystron

For the ILC, the Design Gradient is 31.5 MV/m. But for economical and industrial reasons, we will have a gradient spread within a single accelerator module[7]. This means different quenching limits inside a module. Since three modules at FLASH and ILC will be driven by a single klystron, we have to find a solution to maximize the vector sum along these modules without quenching single cavities. But besides a maximum vector sum without quenching we also want a flat vector sum during beam duration to have equal acceleration for each bunch. This is the problem, when controlling the vector sum instead of single cavities. We have to replace the single cavity voltage with the vector sum voltage in our ODE (3.32) and we will not consider individual properties and limits. This problem will be considered in this section and was under experimental investigation during the 9 mA Run at FLASH. The maximum operating gradients of the cavities before quenching in the modules are shown in Fig. 3.28 and the rf power distribution along the modules ACC 4, ACC 5 and ACC 6 in Fig. 3.29. If we decreased

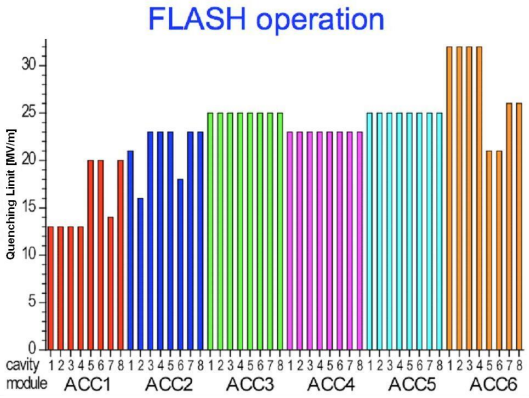


Figure 3.28: This picture shows the gradient spread at FLASH before the upgrade.

the driving power and matched the weakest cavity inside the module to avoid cavity quenches, which looks like the easiest solution, we run into several problems [6]. First, by decreasing the power we also decrease the gradient of every cavity. But this will increase the Q_L of the other cavities. A higher Q_L need more driving power to reach cavity equilibrium in the same time than to a lower Q_L. Fig. 3.30 depicts this situation. And even if we run the module with a higher klystron power, one still has problems.

[7] An accelerator module consists of 8 cavities with fixed power ratios. Usual nomenclature is ACC X where the X stands for the specific number of the module in the accelerator chain.

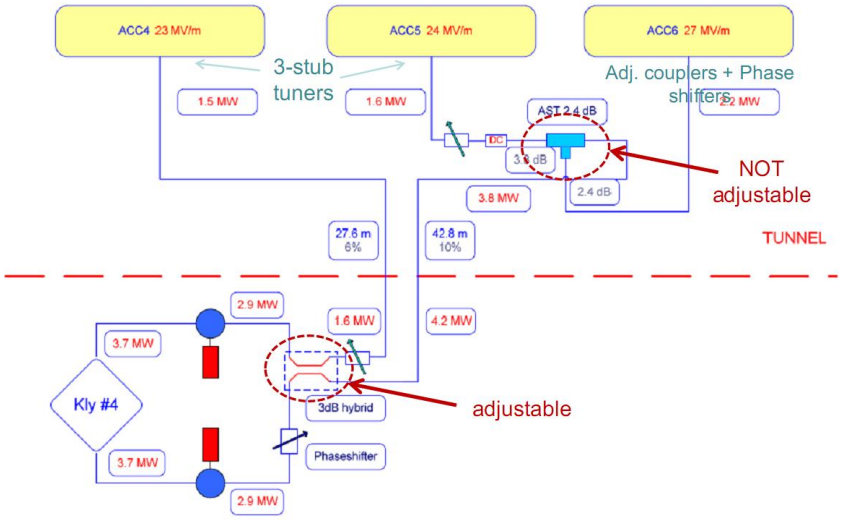


Figure 3.29: The accelerating modules 4,5,6 are fed by a single, two armed klystron. The distribution of the RF power is shown here.

An individual Q_L for each cavity implies that each cavity will have an individual time constant leading to individual filling and decay times (Fig. 3.31). This complicates the synchronization between beam injection time and the phase of the vector sum. A flat top will only be reached if the beam compensates the klystron power. If no beam is present, an increase in Q_L will lead to an increase of the cavity voltage slope. This will lead to a cavity quench in absence of the beam or if the beam is not matched (Fig. 3.32).

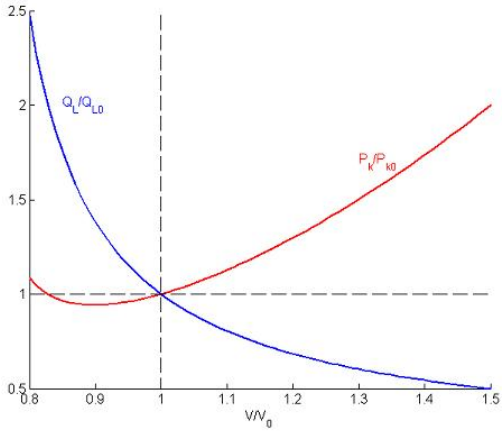


Figure 3.30: Lowering the gradient of a cavity by more than 15%, more klystron power is needed to drive this cavity [5].

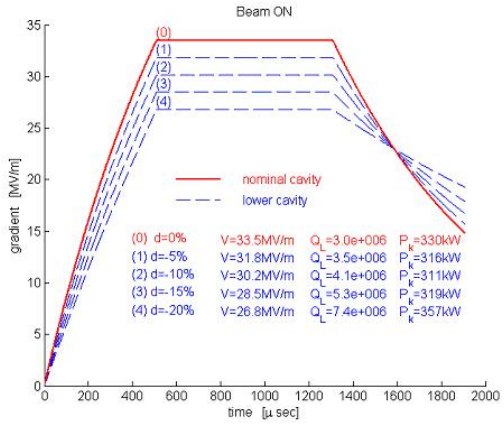


Figure 3.31: Adjusting individual Q_L, no quench in the presence of the beam will happen but different time constants will occur [5].

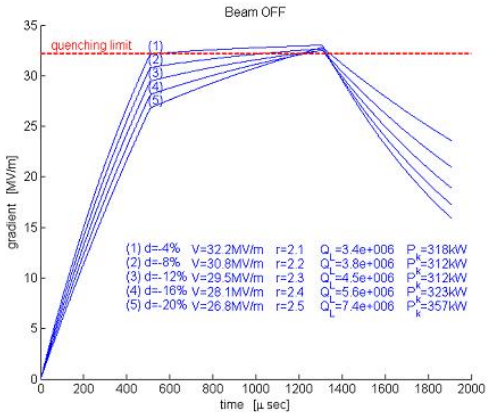


Figure 3.32: Without beam, a change in Q_L leads to different slopes during flattop so that certain cavities will quench during flattop [5].

The questions, which will be under investigation, are

1. How do we reach flattop for the vector sum without and with beam even if there is a gradient spread for multiple cavities?
2. How can we assure, that no cavity will quench and have a maximum vector sum?
3. If a cavity will quench, how can we detect this during acceleration?

The following assumptions will be used throughout the discussion:

- $Q_0 \approx 2 \cdot 10^{10}$ (FLASH configuration)
- $Q_L \approx 3 \cdot 10^6$ (FLASH configuration)
- $E_{acc,max} = 31.5$ MV/m (ILC Design gradient)
- $I_{beam} = 9$ mA - if present (ILC Design current)

3.4.1 Without Beam

Starting with the cavity ODE (3.32) and solving it with the assumptions above, we obtain for a single cavity

$$V_{cav}(t) = 2R_L I_g \left(1 - e^{-\frac{t}{\tau}}\right) - 2R_L I_b \left(1 - e^{-\frac{t-t_0}{\tau}}\right) \tag{3.36}$$

where t_0 is the beginning of the flattop which is also the arriving of the beam in the cavity (next section). To have flattop without beam, the driving power is reduced to make sure that the cavity field strength will not increase. The criterion for the flattop is found by setting the derivative of the cavity voltage with respect to the time equal zero. With no beam we obtain

$$\frac{dV_{cav}}{dt} = 0 \tag{3.37}$$

$$\Leftrightarrow e^{\frac{t_0}{\tau}} - 2 = 0 \tag{3.38}$$

$$\Leftrightarrow t_0 = \tau \ln(2). \tag{3.39}$$

You can assure that each cavity will have flattop when the beam enters the cavity or by decreasing the driving power, when the filling time is chosen that it fulfills the relation (3.39). With the equation $\tau = \frac{2Q_L}{\omega_0}$, the filling time can be controlled by adjusting the Q_L. For multiple cavities we sum up the accelerating fields to the vector sum and the flattop gradient which will be reached is calculated to

$$V_s = \frac{1}{N}\sum_i V_{c,i} = \frac{R_L}{N}\sum_i I_{c,i} = \frac{R_L I_g}{N}\sum_i \alpha_i$$

where the α_i is a coefficient which describes the ratio of the generator current that is distributed to the i-th cavity. These coefficients can be controlled with fixed or variable attenuators or tap-offs. These coefficients are crucial later on when we want to derive a method to prevent cavities from quenching.

3.4.2 With Beam

Now we want a criterion for flattop with beam. With beam, the vector sum of several cavities looks like:

$$V_S(t) = 2R_L \left(1 - e^{-\frac{t}{\tau}}\right) \frac{1}{N} \sum_i I_{c,i} - 2R_L I_b \left(1 - e^{-\frac{t-t_0}{\tau}}\right)$$

if we are only interested in the absolute value of the complex accelerating gradient. Again, taking the derivative and considering only the time when beam is present in the cavity, $t > t_0$, we get the condition for flattop

$$2R_L \frac{1}{\tau} e^{-\frac{t}{\tau}} \frac{1}{N} \sum_i I_{c,i} = 2R_L I_b \frac{1}{\tau} e^{-\frac{t-t_0}{\tau}} \tag{3.40}$$

$$\Leftrightarrow \frac{1}{N} \sum_i I_{c,i} = 2I_b. \tag{3.41}$$

This is the matching condition to assure that when the beam passes through the cavity, the beam current is compensated by increasing the cavity surface current in such a way that the accelerating field will stay constant over the time and will not increase or decrease. This condition can be rewritten to

$$\frac{1}{N} \sum_i \alpha_i = 1.$$

This criterion will be used to find the maximum vector sum. By now we addressed only the first of the three questions. By adjusting the Q_L of each cavity to be the same, every cavity has the same time constant. This means that the equilibrium for each cavity will be reached when the beam enters the cavity, and if the current matching condition is fulfilled, we will have flattop during beam. Now we need to guarantee that no cavity will quench during beam loading and still having a high accelerating gradient. We have not thought about the different quenching limits, but we have a possibility to introduce these problems in our calculations without any change necessary at the derived conditions. The key will be the current distribution coefficients α_i. Starting with considering three cases which are shown in Fig. 3.33. We see, that we have to check the gradients at two times:

- at the beginning of the pulse t_0, the injection time, for cavities with $\alpha_i < 1$

$$V_{c,i}^{max} = V_{c,i}(t = t_0)$$

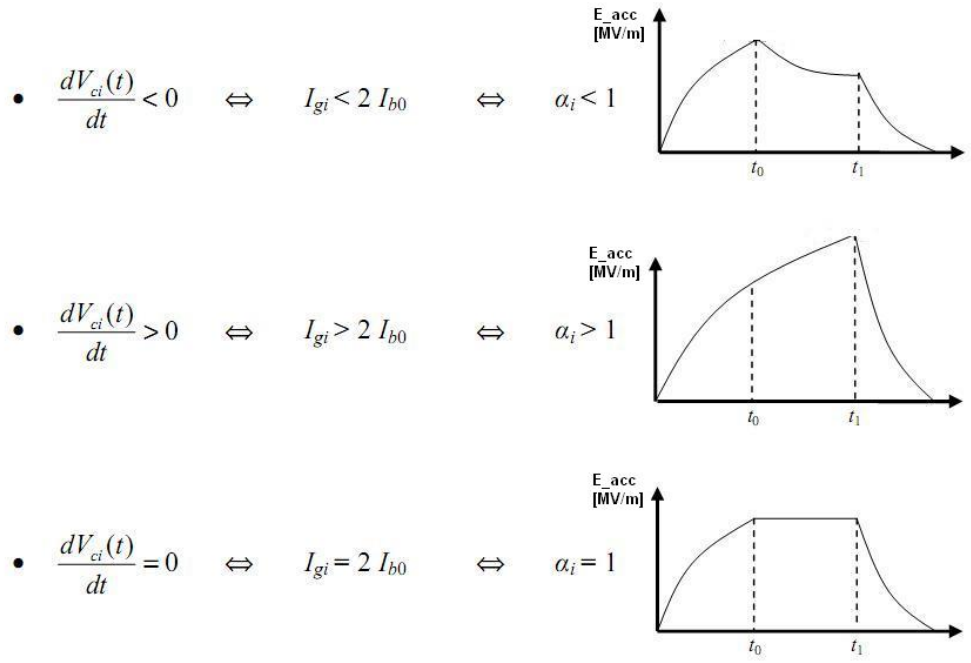


Figure 3.33: First Case: Generator current is too low to match beam load. Second Case: Generator current is too high to match beam load. Third Case: Generator current fits beam load [5].

- at the end of the rf pulse, at t_1 for cavities with $\alpha_i > 1$

$$V_{c,i}^{max} = V_{c,i}(t = t_1)$$

We introduce the abbreviation $\beta = e^{\frac{T_{FT}}{\tau}}$ where T_{FT} is the duration of the flattop[8]. When using the equation for the vector sum voltage with beam and inserting the relevant times, we get the solutions

- $V_{c,i}(t_0) = 2R_L I_b \alpha_i$
- $V_{c,i}(t_1) = 2R_L I_b \left(\alpha_i \left(2 - \frac{1}{\beta}\right) - \left(1 - \frac{1}{\beta}\right)\right)$

Keeping in mind that some cavities will quench below, others above the vector sum gradient we look at two cases (Fig. 3.34)

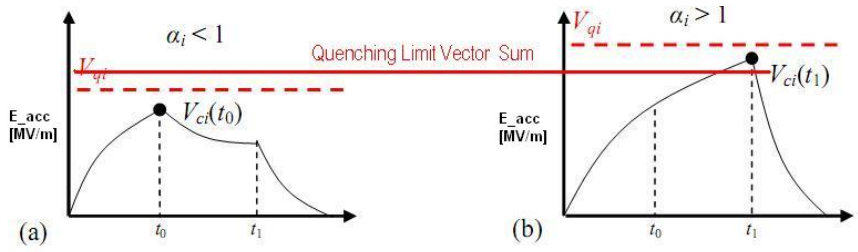


Figure 3.34: Individual cavity quenching limit below or above vector sum gradient [5].

- $2R_L I_b \alpha_i < V_{q,i}$ with quenching limit below vector sum $V_{q,i} < V_S$
- $2R_L I_b \left(\alpha_i \left(2 - \frac{1}{\beta}\right) - \left(1 - \frac{1}{\beta}\right)\right) < V_{q,i}$ with quenching limit above vector sum $V_{q,i} > V_S$

This means, that the critical vector sum gradient will be reached, when cavities with a maximal operating gradient below the vector sum will quench exactly at t_0 and those cavities with a maximal operating gradient above the vector sum will quench at t_1 (Fig. 3.35).

[8] This β has nothing to do with the coupling constant β.

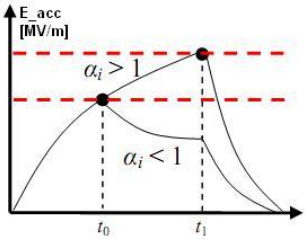


Figure 3.35: The critical vector sum gradient: cavities operating below V_S quench exactly at t_0; cavities operating above V_S quench at t_1 [5].

These criterion can be restated and will lead to useful boundary conditions of the current distribution coefficients

$$\alpha_i < \frac{V_{ql,i}}{2R_L I_b} = \alpha_i^- \tag{3.42}$$

$$\alpha_i < \left(\frac{V_{ql,i}}{2R_L I_b} + \left(1 - \frac{1}{\beta}\right)\right) \frac{\beta}{2\beta - 1} = \alpha_i^+ \tag{3.43}$$

Where the minus sign is for quenching below the vector sum and the plus for quenching above. These coefficients are plotted in Fig. 3.36. Now we need to remember that the averaged α_i need to be equal to unity to obtain flattop during beam. Choose

$$\alpha_i^{lim} = \min\left\{\alpha_i^-, \alpha_i^+\right\}$$

(see Fig. 3.36) and define

$$\bar{\alpha} = \frac{1}{N} \sum_i \alpha_i^{lim}$$

then we define our individual current distribution coefficients to

$$\alpha_i = \frac{\alpha_i^{lim}}{\bar{\alpha}}$$

and the criterion that the averaged sum α_i needs to be equal to unity will always be satisfied as long as $\bar{\alpha} \leq 1$. This is the vital point. Since

$$\bar{\alpha} = \frac{1}{N} \sum_i \alpha_i^{lim} = \frac{1}{N} \sum_i \min\left\{\alpha_i^-, \alpha_i^+\right\} \propto \frac{1}{R_L} \propto \frac{1}{Q_L}$$

a Q_L exist, for which $\bar{\alpha}$ becomes smaller than unity.

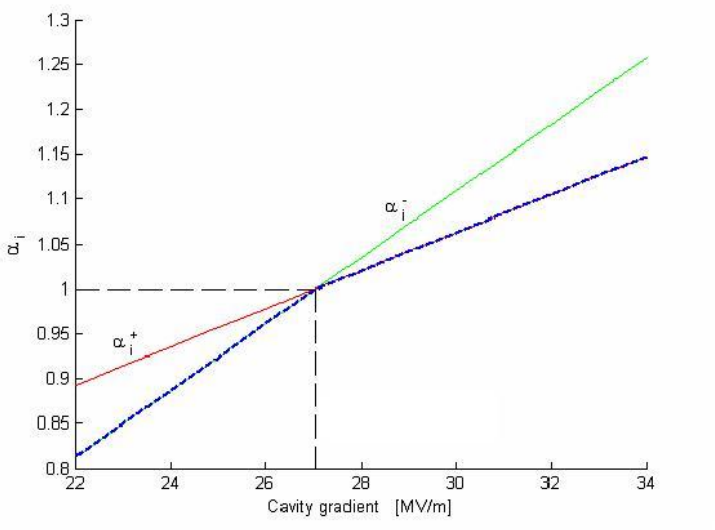


Figure 3.36: Plot of the values calculated for the current distribution parameters with the derived formulas. Blue dotted line is α_i^{min} [5].

With finding the Q_L where $\bar{\alpha}$ becomes equal to unity you find the maximum vector sum with which you can drive a module, since we have the relation

$$V_S^{FT} = 2R_L I_b = 2R\frac{Q_L}{Q_0} I_b$$

This problem can only be treated numerically and with given assumptions above, Fig. 3.37 was derived. This quench prevention model, which is implemented in MATLAB as a cavity Simulator, was tested during the 9 mA run, to validate it and to help predict future problems and make an easier and faster tuning of a machine possible. With these methods, we answered also the second question, how to prevent a cavity from quenching.

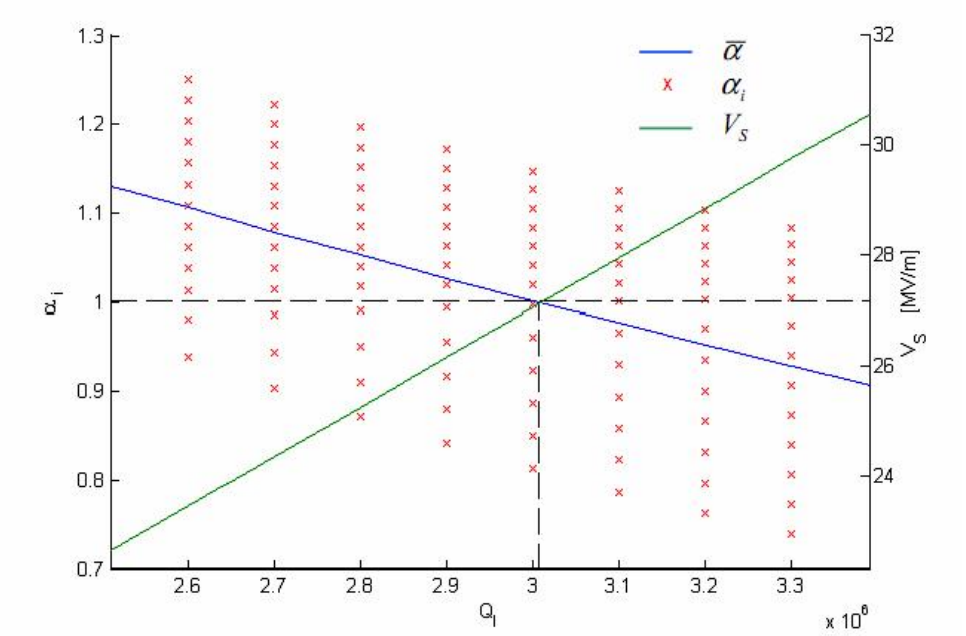


Figure 3.37: By using the MATLAB cavity Simulator, you can solve this problem numerically and derive the maximum vector sum for a given gradient spread [5].

3.4.3 Quench detection

Several reasons can lead to the quench of a cavity (Fig. 2.5), even problems with the cryo system or with heating of the input coupler. Now the question is, how to detect a quench. Two methods will be discussed. Exception handling, e.g. what to do if the quench is detected, is not the topic of this section. First we need to see the characteristics of a quench in the rf signals (see Fig. 3.8). The first method, which is used up to now,

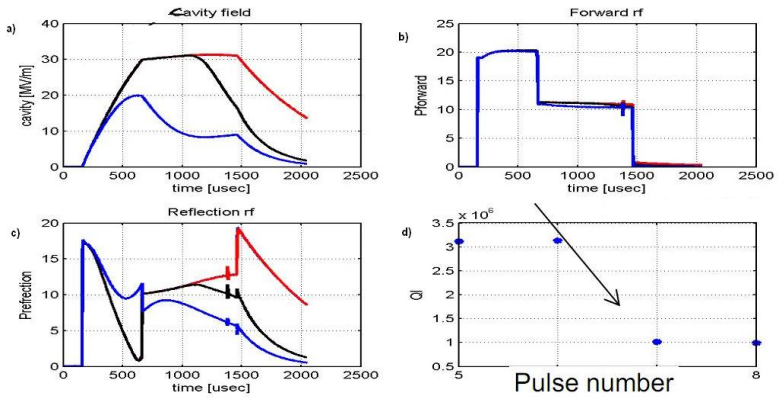


Figure 3.38: This plot shows the time development of a quench. In (a) the transmitted power is shown and in (c) the reflected power. (b) shows the driving signal and (d) the Q_L against the pulse number (cavity no. 2 at ACC6) pulse one is shown in red, the second in black and the third in blue [33].

is to detect a quench by fitting the decay time of the transmitted power at the end of the rf pulse. Since quenching means loosing the superconducting state, the Q_L will drop immediately and the cavity time constant will decrease. The accelerating field will decay faster, which can be measured. The disadvantage is, that the quench will be detected only at the end of the rf pulse and problems like instabilities in the cryosystem and an increased energy spread or even beam loss will occur. A second method, allowing intra pulse detection, could be possible when using the cavity ODE (Eqn.3.32). By solving the ODE for Q_L, and with every needed value provided via DAQ, real time quench detection could be possible. Even the beam current can be included when the toroid signals are scaled and used for calculation. This method is under development and only offline DAQ analysis is possible yet, since no FPGA code exist. The question under investigation is, if the second method is stable, fast and accurate to replace the existing method.

4 Implementation of LLRF control at FLASH

In general there are two possibilities to drive a cavity. The so called GDR (generator driven) mode and the self excited loop (SEL) mode. In the GDR mode (Fig. 4.1), the

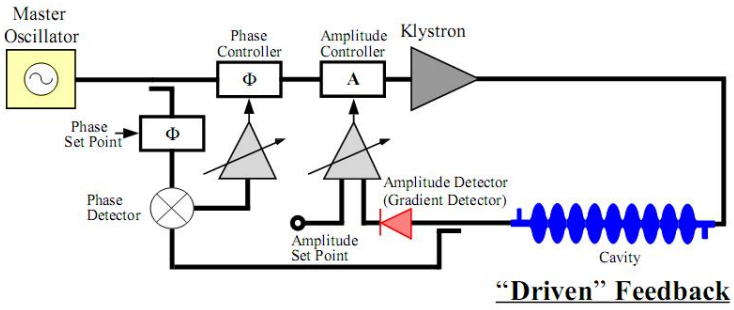


Figure 4.1: Sketch of GDR Model.

master oscillator gets controlled in amplitude and phase and is amplified by a klystron or a solid state amplifier and this signal is used to drive the cavity. A pick up antenna measures the field in the cavity and a phase and amplitude detector is used to derive an error signal in comparison with the setpoint. This error signal is used as input signal for the feedback control. The advantage of the GDR system is given, when fast lock up times are critical, i.e. pulsed systems and it is easily adaptable to I/Q domain for digital control. Disadvantages are, that the system needs tuning elements to keep the cavity close to the reference frequency. And machines with high Q_L and microphonic distributions and large Lorentz force detuning could become unstable and loose resonance lock. In the SEL mode (Fig. 4.2) the probe signal is bounded in amplitude by a limiter. Apart from that, the probe signal is directly fed back to the klystron, so that the accelerating field inside the cavity will start from noise if the loop phase is a multiple of 2π. This means that random noise gets amplified by the feedback if the noise is near the resonance frequency. The cavity frequency still must be locked close to the reference

frequency to avoid saturating the power amplifier. The advantage lies in operating high Q_L cavities and systems with Lorentz force detuning, since they follow any change in frequency and phase. The disadvantage is the slow lock up time. Basically, if the system

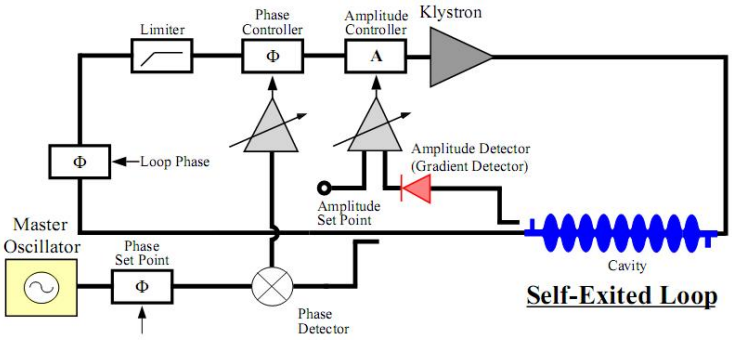


Figure 4.2: Sketch of SEL Mode.

is locked, these methods are equivalent. Digital SEL can follow a system only in a certain range which is limited by the digital filter. And the SEL must handle a spinning phasor when the system is not locked. This means, that the state vector rotates with an ωt term until the feedback locks the system onto the correct frequency and phase which is not the case for a GDR system. After choosing which mode will be used one has to derive an rf system model and derive the state space representation. This representation is discretized and implemented into the controller algorithm.

4.1 Requirements at FLASH

The LLRF control needs to maintain the accelerating field, e.g. amplitude and phase, within given tolerances for several cavities. As an example the next sections will show the LLRF control for the first accelerator module in the FLASH chain (ACC1), which is an accelerator module containing 8 cavities. The requirements which need to be met are [16]

- $\left(\frac{\Delta E}{E}\right)_{Beam} < 2.7 \cdot 10^{-4} \Longrightarrow \left(\frac{\Delta A_{V_S}}{A_{V_S}}\right) \leq \pm 10\%$ and $\left(\frac{\Delta \phi_{V_S}}{\phi_{V_S}}\right) \leq \pm 1^o$
- Lorentz force compensation, since LF detunes a cavity by ≈ 215 Hz
- Beam loading compensation, since $I_B = 8$ mA induces an amplitude drop of $\left(\frac{\Delta A}{A}\right) = 1.4 \cdot 10^{-3}$
- Drive multiple cavities with a single klystron: Not to exceed single cavity limits (quench)
- Reach maximal beam energy, which means reach maximal vector sum
- With new accelerator module ACC39, 3rd higher harmonic needs to be controlled
- finite state machine, describing the accelerator for complex procedures (e.g. turn on klystron)

The plant which needs to be controlled at FLASH is shown in Fig. 4.3.

4.2 Feedback

4.2.1 Principle layout of LLRF control

When designing the LLRF control system, you have to identify all available attenuators and actuators and pick up signals. The time constant τ_c of a superconducting cavity at FLASH is about 700 μs which is much longer than the sampling interval $T_s = 1$ μs. This allows a good system modeling which leads to a good 'fast noise' suppression. The latency of the feedback loop is about 4 μs, which is the sum of transport- (600 ns), calculation- ($2 - 3$ μs) and conversion delays (ADC 500 ns, DAC 200 ns) and an additional loop phase shift of approximately 100 degrees. A fast amplitude and phase control of multiple cavities needs to control three incident waves for each cavity. This applies 24 signals for a single accelerator module. To realize such a control system, the modulation of these

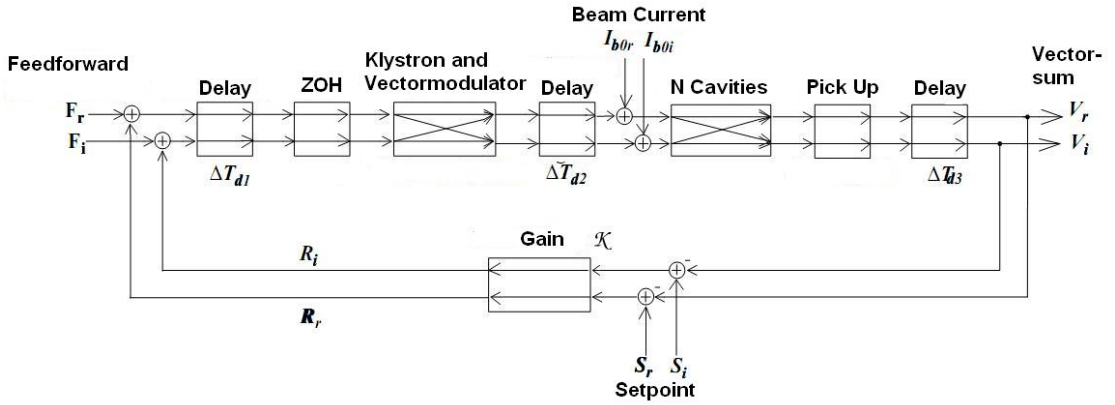


Figure 4.3: Principle layout of the closed loop system which needs to be controlled [29].

waves is chosen to be I/Q modulation, introduced in the chapter before. The sampled signals need to be scaled and rotated by a 2x2 matrix and the vector sum is calculated. In parallel, the transmitted power is analyzed to identify the transient beam loading for compensating the beam loading effects and also to derive the correct tables for the feedforward control system which is used to compensate the Lorentz force detuning and other repetitive, slowly changing errors. The signals are split up a third time and fed to the DAQ system[1]. Individual graphs can be plotted and saved for offline analysis, trouble shooting and manually operated exception handling. In Fig.4.4, the layout of the system is shown. The respective functional block diagram of the controller loop is shown in Fig.4.5. The control block and the algebraic model (derived in chapter 3) which is implemented as sketched in Fig.4.6. This algorithm uses the state-space representation derived in chapter 3. The algorithm runs in the FPGA, while signal conversion, summation, rotation and scaling is done in the DSPs. The algebraic model is used to derive the needed signals to steer a cavity to the desired state. This model runs simultaneously inside the FPGA to identify the cavity state and to predict its further development. It consist of successive steps, done for each cavity. The set point is compared with the actual state and error signals and feedforward tables are applied to the driving signal. The driving signal u_k is the forward power, the cavity phasor v_k the transmitted power of the k-th cavity. The transmitted power v_k is calibrated, rotated and scaled, the vector sum is calculated and compared to the set point value.

[1] At DESY,the DAQ and Controlsystem for the accelerators is DOOCS (Distributed Object Oriented Control System). It was developed on site and is under continous enhancement.

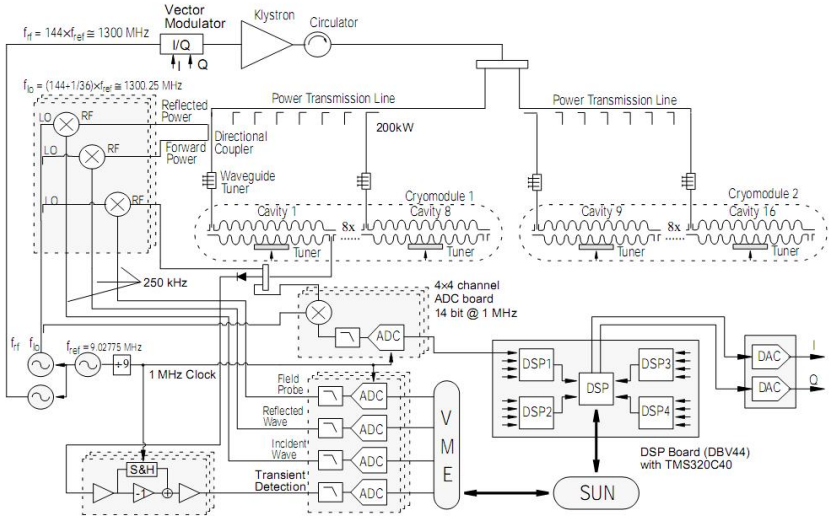


Figure 4.4: Layout of the LLRF Control of the ACC1 module at FLASH [30].

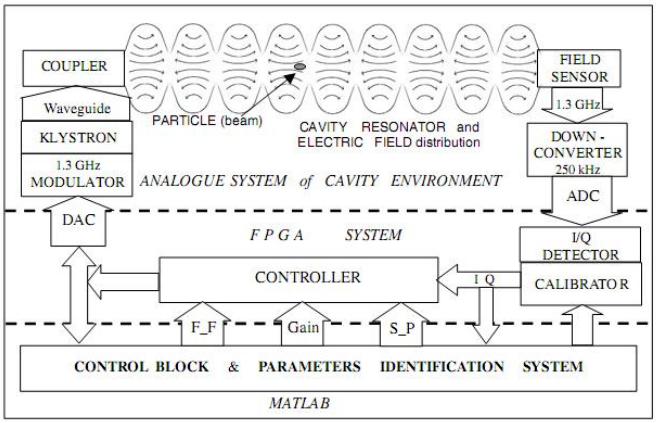


Figure 4.5: Functional block diagram of the LLRF control system for one cavity [31].

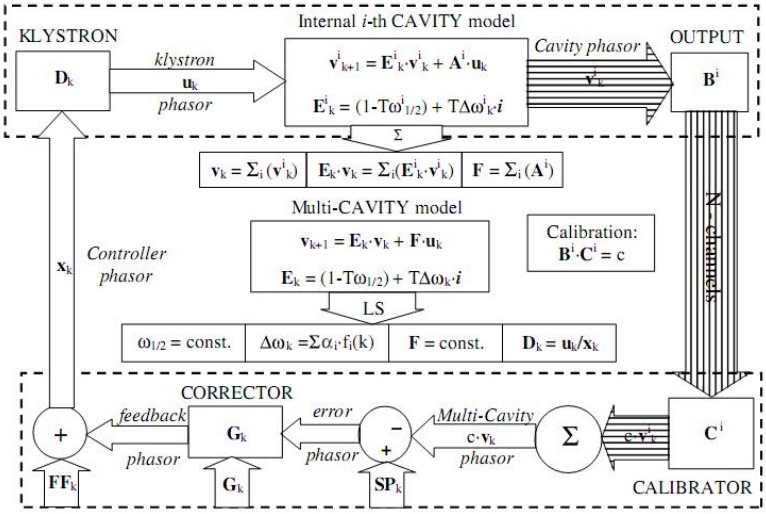


Figure 4.6: Algebraic model of the feedback and feedforward system [31].

The error value is fed into the correction matrix and the result, the feedback phasor, is added up with the feedforward phasor. This results in x_k, where this signal is the driving signal for the klystron, which itself generates the forward power u_k. More detailed descriptions can be found in [3], [11], [14], [16], [29], [30], [31].

4.2.2 Open loop, no beam operation

Some results of the work at ACC1, described in the section above, are shown here. The work has been carried out at at FLASH [14]. During the next cases, following parameters where held constant. The filling time with 500 μs. After that time, forward power was dropped to keep a nearly steady state value of the cavity. The flattop had a duration of about 800 μs. After 1.3 ms the rf power was turned off. In Fig. 4.7 the result in driving ACC1 without feedback, so called open loop operation, is shown. The upper plots include the amplitude and phase the individual cavities. The lower plot shows the vector sum of the cavities. Even though the vector sum shows a negligible slope, it can be seen that constant individual gradients are not necessary to reach a flat vector sum. The phase drift is due to detuning through Lorentz force.

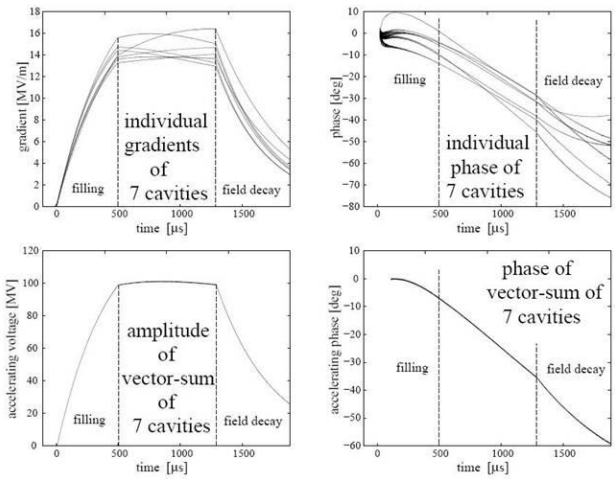


Figure 4.7: Open loop, no beam operation of ACC1. Even with no flat gradient for individual cavities a flat vector sum of the cavities can be achieved [14].

4.2.3 Closed loop, no beam

For closed loop operation, the gain of the I- and Q-component needs to be set. In this case, the gain was set to 30, the accelerating voltage for the whole module was set to 115 MV. The plots in Fig. 4.8. show the results of this operation. The RMS during the flattop for the vector sum is

$$\left(\frac{\Delta V}{V}\right)_{RMS} = 0.2\%; \Delta\phi_{rms} = 0.25^\circ$$

This time each cavity has a constant gradient during flattop. The vector sum of the cavities acts like a single cavity, driven by one klystron.

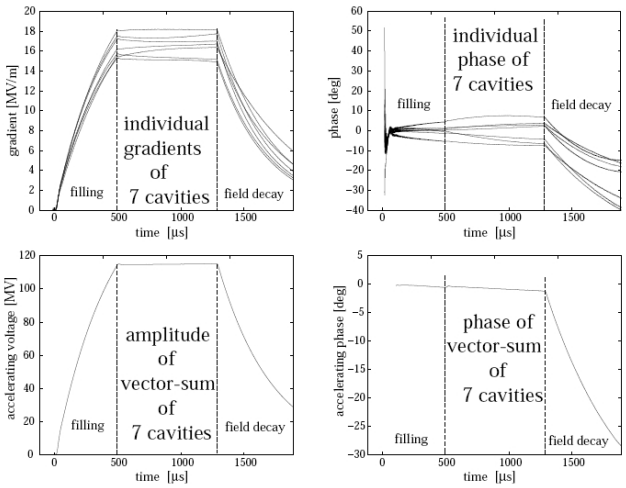


Figure 4.8: With closed feedback loop a higher vector sum gradient can be achieved, the cavity detuning is compensated and the gradients of individual cavities will not drift [14].

4.2.4 Closed loop, with beam

The acceleration of a beam current of 6mA, pulse length of 30 μs was tested at ACC1. Feedforward has been applied to the controller to reduce Lorentz force detuning. The RMS is calculated for the whole flattop. The stabilities achieved were:

- with feedback (gain = 70), without feedforward:

$$\left(\frac{\Delta V}{V}\right)_{RMS} = 5 \cdot 10^{-3}; \left(\frac{\Delta \phi}{\phi}\right)_{RMS} = 0.1^{\circ}$$

- with feedback (gain = 70), with feedforward:

$$\left(\frac{\Delta V}{V}\right)_{RMS} = 5 \cdot 10^{-4}; \left(\frac{\Delta \phi}{\phi}\right)_{RMS} = 0.03^{\circ}$$

When comparing the results achieved with the feedback system turned on, to the requirements in Section 4.1. concerning field and phase stability, the goal was reached within a single module. The idea to use this feedback system to drive a single cavity with a high Q_L is described in the next chapter.

5 Implementation of LLRF control at the vertical test stand

The vertical test stand at DESY is used to determine Q_L as a function of the accelerating gradient E_{acc}. This method gives us the possibility for several applications. The main goal is to test the cavity if it is capable to run an accelerator at high gradients. This includes a research of an optimized cavity treatment, like quality assurance of important steps in cavity surface treatment or the assembly of special parts to the cavity. A test is done between each step to assure the quality of the cavity and to understand the different procedures and their influence onto the cavity and understand the mechanisms happening inside the cavity (Fig. 5.1). Another point is the possibility to improve the

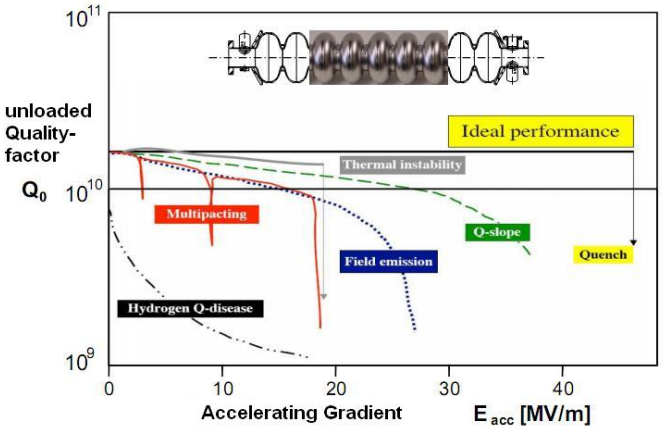


Figure 5.1: This plot sketches different types of Q-E-Plots and what informations we can derive about the mechanisms inside a cavity [18].

cavity parameters while testing when the cavity shows field emissions. This procedure is called processing. When reaching a certain rf power which leads to strong field emissions you stay at this power level to try to burst the impurity. This method is not always successful. Another possibility, introduced through this work, is the possibility to test several hardware / firmware components at the vertical test stand.

The LLRF control used at FLASH is capable to run several cavities. To show that this control system can run the vertical test stand, this work has been done. Even that it may look obvious that this should be possible some problems need to be solved. For example, exact and absolute values are needed to get good results while at FLASH, subsystems carry out some of these tasks and the LLRF control is working on error signal and relative values. Besides this, a higher Q_L increases the requirements onto the control algorithm speed. The layout of the construction can be seen in Fig. 5.2.

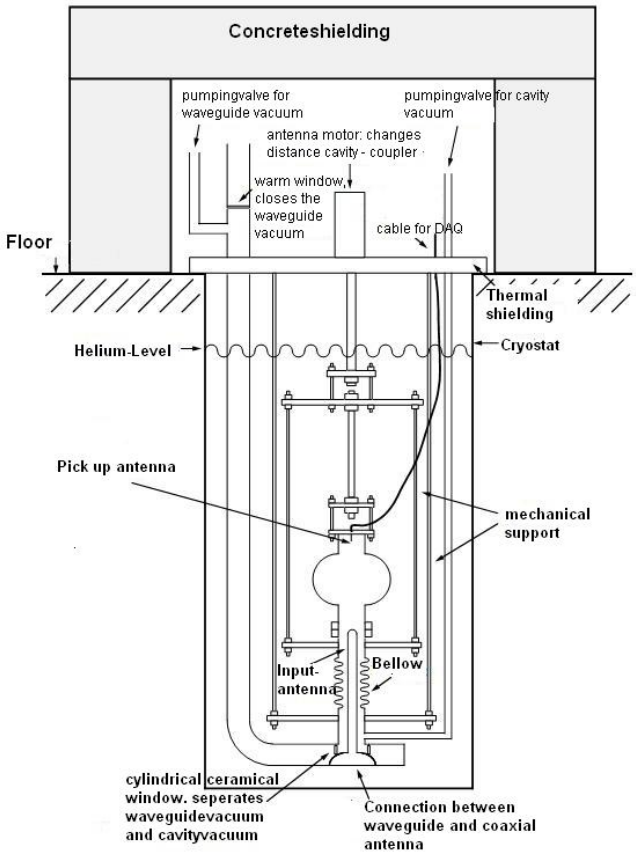


Figure 5.2: Engineering drawing of the vertical test stand at DESY [43].

5.1 Analog test stand

The frequency generator creates an rf signal, which is chopped to a pulsed signal via a pin diode, which itself is driven by a function generator (see Fig. 5.3). This pulsed driving signal is amplified by a solid state amplifier and is then fed to the cavity. Between amplifier and cavity a circulator exists to protect the amplifier against the reflected wave. Directional couplers pick up the forward and reflected waves to measure their power level and the signals are used to drive the Phase-Locked-Loop (PLL), to make sure that the system stays on resonance. Inside the PLL, the Phase of the forward and the transmitted Power is compared and the error signal is used to drive a phase shifter which shifts the driving signal phase.

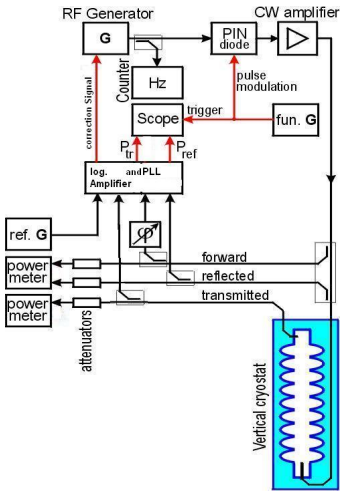


Figure 5.3: Layout of the signal path at the current testing system [43].

To derive the Q-E-Graph, one has to measure forward, reflected and transmitted Power. Once, this is done, following formulas are needed [1].

$$Q_0 = (1 + \beta) \cdot Q_L \tag{5.1}$$

$$Q_L = 2\pi \cdot \omega_{0,n} \cdot \tau \tag{5.2}$$

τ is calculated as an exponential fit to the decaying part of the transmitted power while the resonance frequency is read out of a spectrum analyzer.

[1] see equation 2.54. β_t can be neglected since we are using a nearly loss free antenna.

The coupling β is determined through the reflected Power. Depending on the duration of the rf driving signal, different formulas has been derived to calculate the coupling β [43]:

- $\beta = \frac{P_e}{P_i - P_r}$ for pulsed mode
- $\beta = \frac{1-\sqrt{\frac{P_r}{P_f}}}{1+\sqrt{\frac{P_r}{P_f}}}$ used when cavity reached steady state
- $\beta = \frac{1}{2\sqrt{\frac{P_f}{P_e}}-1}$ measuring when rf is turned off

The definitions of theP_x can be taken from Fig. 5.4. The calculation of the coupling is a vital point. Any errors arise during sampling the reflected power will have a large influence onto the Q_0 calculation. To derive the accelerating gradient E_{acc} following equation is used [2]

$$E_{acc} = \frac{\sqrt{\left(\frac{r}{Q}\right) \cdot Q_0 \cdot P_{diss}}}{n \cdot l} \tag{5.3}$$

Where n denotes the number of cells the cavity consist of and l the length of a single cell. $\left(\frac{r}{Q}\right)$ is the shunt impedance. The dissipated power can be calculated through

$$P_{diss} = \frac{4\beta}{1+\beta} \cdot P_f - P_r - P_t \tag{5.4}$$

when the cavity is in equilibrium, which means, the cavity is filled and flatop is achieved.

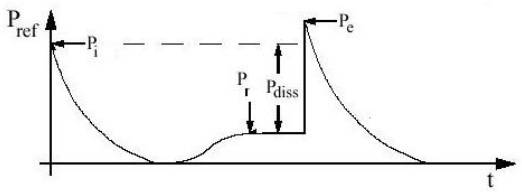


Figure 5.4: Relevant points of the reflected Power.

[2] see equation 2.24 with the electric field distribution of the TESLA cavity.

5.2 Digital test stand

5.2.1 Principle signal path

In Figure 5.5, a schematic view of the digital system is given. The frequency generator creates a 1.3 GHz sine signal near to the resonance frequency of the mode. This signal goes to the rf box (see next section). From the rf box a 1.3 GHz sine signal is fed to a vector modulator. In the vector modulator, this signal is split up into two signals which are phase delayed to each other by 90 degrees. Each is modulated by the feedback I/Q-signal from the DAC of the SimCon board in a mixer of the vector modulator. At the output of the vector modulator, these two signals are added up, amplified and then fed to the cavity. The forward and reflected power is coupled out and the transmitted power is picked up and fed to the SimCon Board. The ADCs can not handle the resonance frequency of about 1.3 GHz, so that we need to downconvert the signal to the intermediate frequency (IF) of 9 MHz. The IF is then fed to the ADC and sampled with a frequency of 4 times the IF, respectively 81 MHz which is delivered from the rf box. Inside the controller board, the control algorithm is applied. Both modes, GDR and SEL were under investigation.

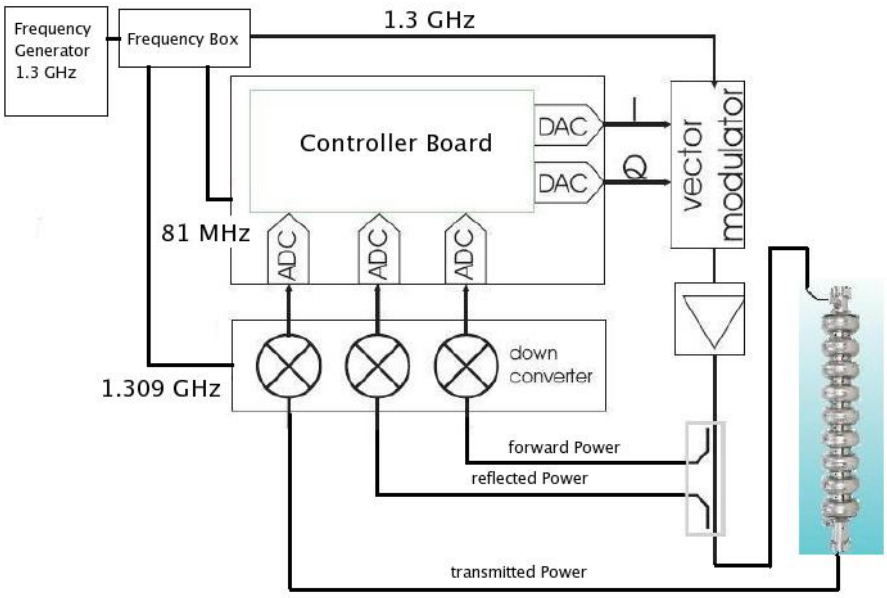


Figure 5.5: Layout of the digital testing system.

In principle, the signals have the same path like in the analog system. A driving signal is generated and fed to the cavity. The transmitted and driving signal are sampled and used to lock on, where the error signal is used to correct the driving signal. The only difference is in the controller algorithm. The analog system is fixed to the GDR mode while the digital system can switch between GDR and SEL and it is under investigation which is better to use.

5.2.2 Controller board and algorithm

The main part of the system is the SimCon-Board (Simulator and Controller Board), which has been developed at DESY. This board, a picture is given in Fig. 5.6, is also the controller board for the accelerator modules at FLASH. Different versions of this board exist and the latest one, the SimCon DSP, is used at the vertical test stand. The main challenge in controlling the test stand is the high Q_L, which can be about 10^4 higher than in an accelerator since the input-antenna attached at the test stand are high quality-factor antennas. This leads to a narrow resonance peak the system needs to lock on. Hence a faster feedback algorithm has to be developed. How this feedback algorithm is implemented depends on the mode the cavity is driven on, GDR or SEL. A similar approach has been done before at the horizontal test stand at DESY. The difference to the vertical test stand is that fully equipped cavities are tested under working conditions and the input and HOM couplers lowers Q_L to the nominal values of the accelerator [15].

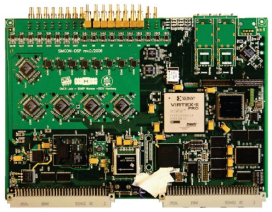


Figure 5.6: Picture of the used SimCon Board (courtesy of S. Simrock).

The SimCon Board consists of 8 14bit-ADCs and 8 16bit DACs, a Xilinx Virtex IV FPGA-Processor and a Tiger Shark DSP. It has also two optical ethernet connections and a VME interface for communicating and programming the board. The DSP is used at FLASH to calculate the vector sum and do the needed rotations and scaling while the FPGA is used for the controller algorithm. A more detailed description can be found in [14]. The characteristic curves of the hardware, e.g. calibration of the ADC, can be seen in Section 8.5, e.g. Fig.8.5.2-8.5.4.

They show a nonlinear behavior during the measurement at about 50% of their range. These nonlinearities will be considered later on. The ADCs are sensitive to a voltage of the range -1V to +1V. The ADC measures the voltage and not the power and will also be discussed later. This is an important feature since changing cables between the ADCs and the directional couplers will lead to a change in resistance and a new calibration need to be done. Another problem is that the range from -1V to +1 V is a small window to realize a full measurement of a cavity up to 50 MV/m which can mean a driving power of 200 W. The controller algorithm needs to handle a sampling frequency, which is also the internal clock, of 81 MHz. Due to conversion and calculation delay, the algorithm needed 31 clock cycles before an output is generated. The newest version, working as an SEL mode, needs 7 clock cycles and a tuning accuracy of about Δf = 0.5 Hz. This faster and more accurate algorithm used to drive high quality factor oscillating circuit together with a newer controller board can improve the stability of an accelerator and the possibility of a higher Q_L in the machine.

5.2.3 RF box

The frequencies, needed to drive the system are generated by an rf box (Fig. 5.7) which has been developed for this purpose. There is a fixed relation of the frequencies to each other.

$$IF = f_0 - f_{DWC} = f_0 - \left(f_0 + \frac{f_0}{144}\right)$$

$$f_S = \frac{f_0}{16}$$

$$\Longrightarrow f_S = 9 \cdot IF$$

With IF as intermediate frequency, f_0 as resonance frequency of the mode under investigation, f_{DWC} as the frequency used for down-conversion and f_S as the sampling frequency. These relations assure that no aliasing will occur since we fulfill the Nyquist-criterion. Besides this, the IQ-Sampling relation is also realized. The rf box guarantees synchronized frequencies with fixed phases to each other.

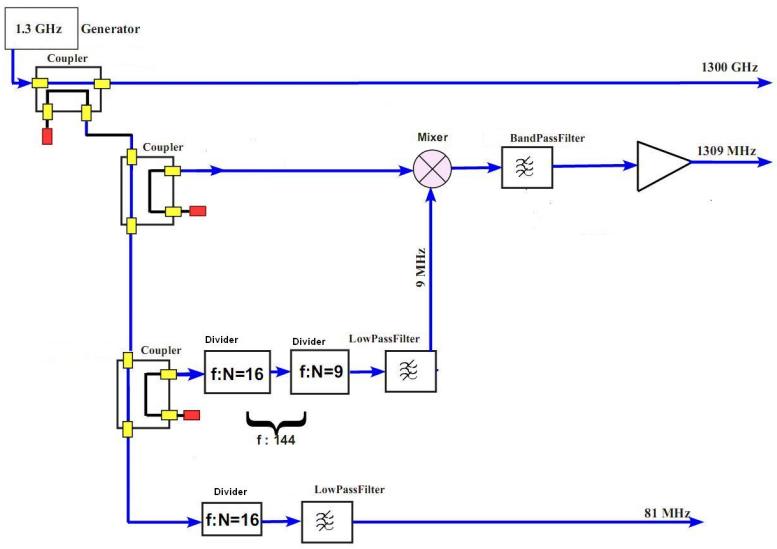


Figure 5.7: Technical drawing of the rf box used for generating the required frequencies. A picture of the hardware is shown in the section 8.5.

5.2.4 GUI and communication

The communication of the graphical interface with the SimCon Board, respectively the FPGA, is done via MATLAB and an internal interface server (Fig. 5.8). The algorithms, which control the registries of the FPGA, the frequency generator, the box and other parts of the system and the communication are shown in Appendix 1. Basically, the communication between MATLAB and the FPGA is done via *.mex-files or MATLAB executable files. They can be written in C/C++ or directly in MATLAB and are compiled onto the operating system on which MATLAB will be used. A more detailed description of the communication and the different layers can be found in [27], [26]. The principle of the communication is shown in Fig. 5.8. In Fig. 5.9 the recent graphical user interface is shown. This is the latest version of the developer environment but need to be changed when it should be operated by non experts. The automated measurement will make some of the shown panels and steering possibilities redundant.

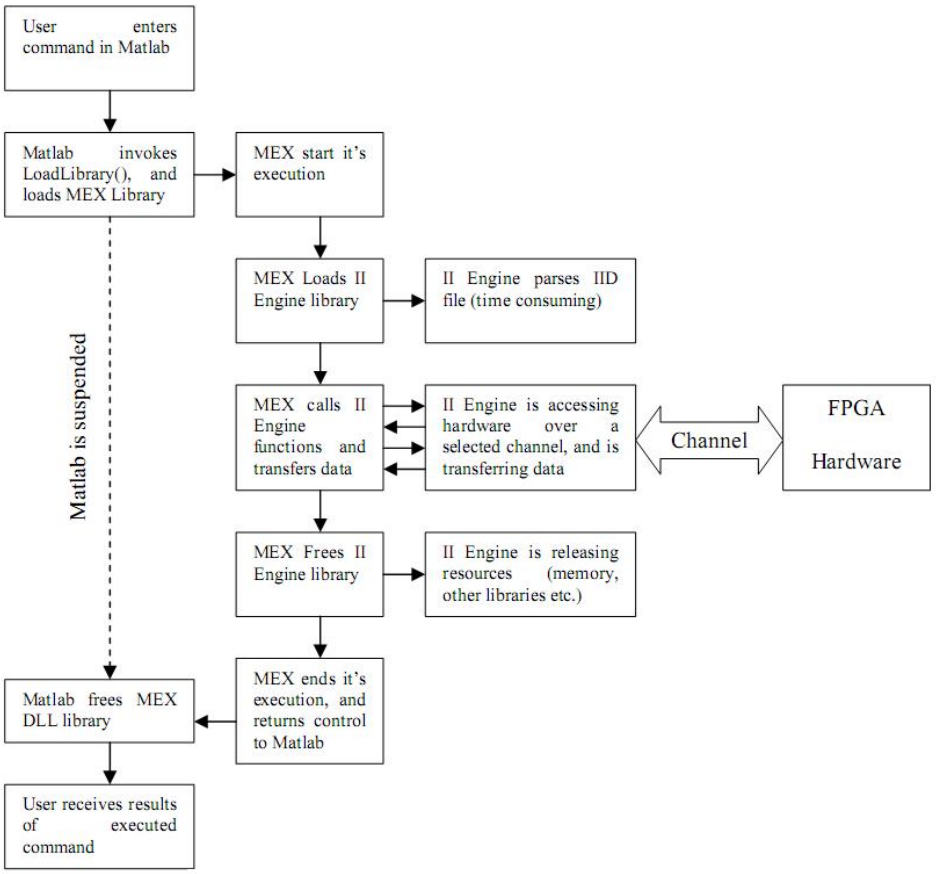


Figure 5.8: Drawing of the communication of MATLAB with the FPGA via TCP/IP and/or VME Bus [26].

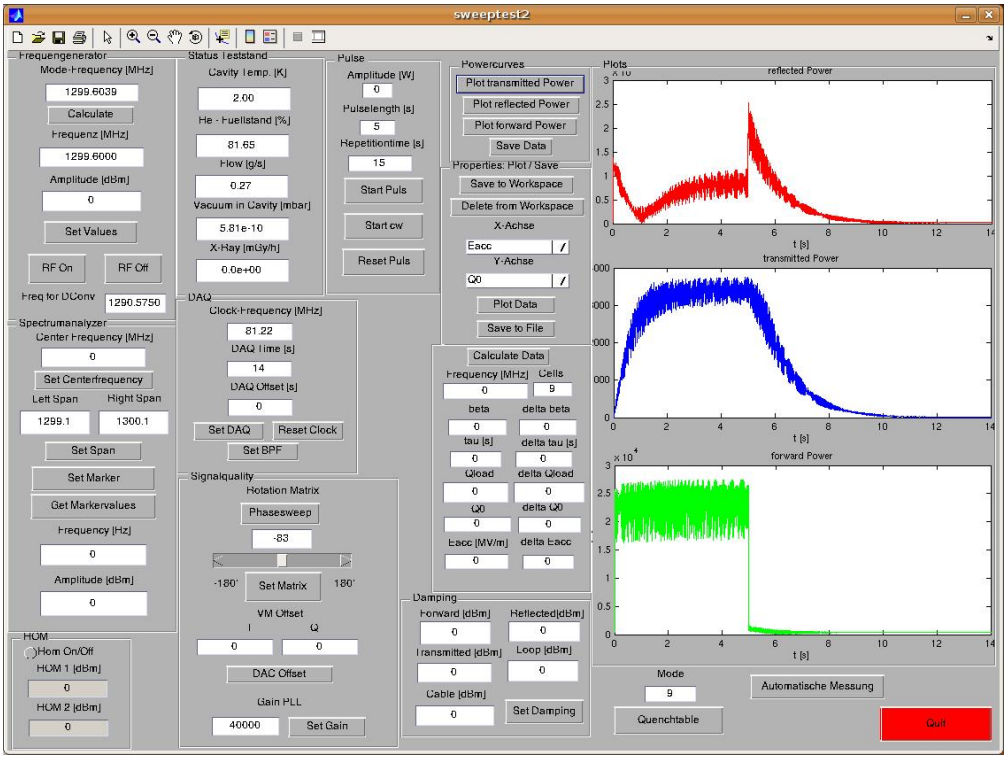


Figure 5.9: Latest version of the graphical user interface developed in MATLAB. The GUI is used to control the generator, spectrum analyzer and the FPGA and to monitor every needed value of the test stand. It is also used to plot every needed graph and calculates the needed and searched parameters (MATLAB Code in Section 8.1).

5.2.5 Results of the digital test stand

In Fig. 5.10 and 5.11 two measurements done with the digital system in comparison with the analog system is given. In Fig. 5.10 an earlier version of the measurement

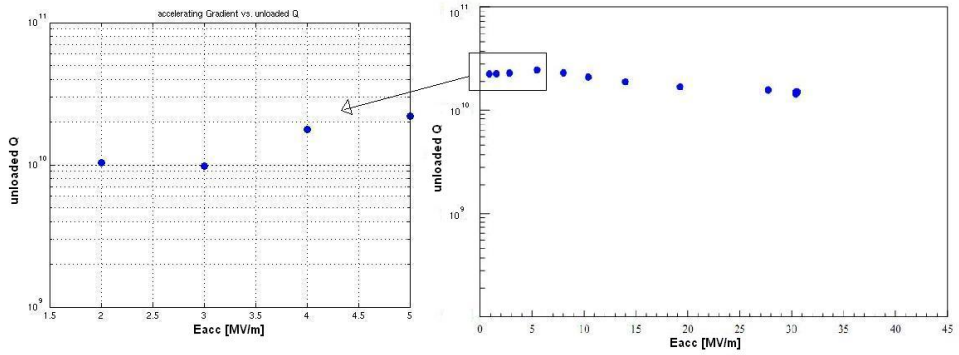


Figure 5.10: Cavity Z109. Left: Q-E-plot derived with the digital system. Right: Q-E-plot derived with the analog system.

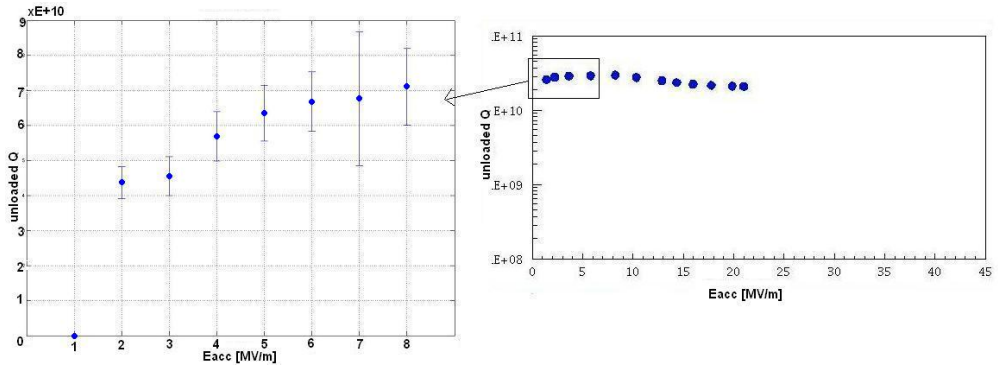


Figure 5.11: Cavity Z138 with HOM couplers. Left: Q-E-plot derived with the digital system. Right: Q-E-plot derived with the analog system.

algorithm is shown. Besides the fact that the y-scale, respectively the Q_0, has been changed, since it is favorable not to take the logarithmic scale at too low input power, an error calculation has been added which gives rise to a more accurate interpretation of the results (Fig. 5.11).

The general behavior of the cavity, respectively the Q-E-plot, could be reproduced by the digital system, but is still not in agreement with the absolute values of the analog systems. But an intrinsic error of about 10% of the analog system can be added due to limitations in the resolution of the powermeter. These plots proof the feasibility of the concept, driving a vertical test stand and a high Q_L cavity with a digital control. An automated measurement has been realized. The algorithm finds for each data point the correct phase of the driving signal within 248 seconds and this is reproducible without any external control. Some improvements have been done to make this phase sweep even faster but could not be tested up till now. The automated measurement takes 1113 seconds for finding the correct phase, evaluates and monitors every needed signal for 4 points. For a single point 286 seconds are needed. This makes it possible for an estimation of the time needed for a cavity measurement. The so called processing is still not analyzed and completely structured. Further investigations towards this are needed to see if an algorithm can be developed which mirrors the decision tree while treating a cavity this way. The automated search for the resonance modes of a cavity is not implemented in MATLAB but the algorithm is working correctly, which has been shown with LabView. When taken the error of about 10 % in the quality factor of the analog system into account and with the error of the digital system shown in Fig.5.11 the plots show an agreement.

5.2.6 Problems

Some Problems are still unsolved. Some because they were not completely understood, some just because the lack of time. As mentioned above, the forward power can only go up to several Watts, since the ADCs reach saturation too fast. This results in a limitation of the accelerating gradient, respectively the x-axis. The reason for this is, that at low signal levels, noise coming from the controller algorithm, the PLL or even thermal noise can lead to errors and thus it is not easy to reproduce the analog system results. The digital system is sensitive in a low power regime which will not be the problem at high power levels beyond several Watts. For example, one of the biggest problem is the calculation of β. In Fig. 5.12 the signal of the reflected power is shown. The oscillations seen there lead to an error of about 30% of the coupling. FFT analysis of the signal has been done (see Section 8.6) which has shown some sidebands at the higher harmonics of the rf box to down-converter signal. After low pass filters were installed we expected the oscillations to disappear. Unfortunately, the noise could still be seen. A more precise analysis of the data showed, that the oscillations are the result of two signals interfering with the 9MHz IF, which are at IF + 16 Hz and IF + 72 Hz. This small deviation is the reason, why they can not be seen in the FFT. The source

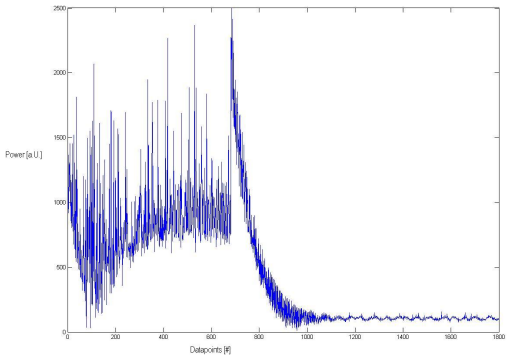


Figure 5.12: Oscillations in the reflected power lead to a huge error in the coupling (see Fig. 5.4).

of the noise is not found yet. Some tests showed that there are differences of the noise signal level and frequency in the forward and reflected Power during and after the pulse. That hint leads us to a possible source. The noise can be produced by non-linearities in the mixers of the down-converters, since they generate higher harmonics which can have a significant influence at input powers beyond the linear regime of the analog parts in the controller chain, respectively the down-converters and the ADCs. These higher harmonics are aliased either to DC respectively the Nyquist frequency (even higher harmonics) or the IF frequency (odd higher harmonics). A typical symptom of this error source is, that the distortion changes when carrier frequency and phase changes. This is the case, as mentioned, when the rf is turned off and the system looses the lock to the resonance frequency and phase and the driving signal frequency and phase is sampled. The odd higher harmonics mapped to the IF signal can not be distinguished from the carrier frequency by standard IQ sampling resulting in false data measurement. By applying non-IQ-sampling to the signal, we should verify the noise source as non-linearities within the system. Non-IQ-sampling means, that the ratio of sampling frequency to intermediate frequency is not an integer but rational [36] [45]. By changing to Non-IQ-sampling an improvement of the system accuracy should be possible. Another problem already mentioned is the limited power range, coming from the ADC limitation. One solution could be installing fixed attenuators to keep the signal level in the linear area and stay away from saturation. But this solution will increase the resolution and is only useful if the signal level can be estimated, which is in fact not known until the cavity has been measured. Another solution is to split up and connect

one signal to several ADCs with different attenuators in front and switch between the ADCs for different power ranges. This can be done but would need an ADC-Board for this setup. This would lead to a delay time and will result in a high latency which could be a problem for high Q_L - cavities. A favorable solution is the installation of digitally controlled attenuators. They will change the attenuation in dependence of the input signal level and thus will keep the signal in the linear regime of the ADC. The attenuation needs to be taken into account when calculating absolute values but this is no problem since they can be controlled and monitored via the MATLAB GUI. The calibration of the ADCs can be automated if the powermeters will be kept in the system and are read out via VISA/GPIB. A script to generate pulses, read out the ADCs and the power level at the powermeters exists.

5.2.7 Possible further developments

Besides these tasks which still need to be done, some general improvements can be done. The mode used to run the digital test stand during the feasibility study is the GDR mode. By implementing the SEL mode a faster measurement is possible since the phase sweep is not needed. A newest FPGA code was tested using SEL and showed some nice features. But due to hardware problems and time limitations no results can be shown. Another big improvement would be a completly new algorithm to derive the Q_L. This could be done by using the quench detection method derived above when using the cavity ODE. An example is shown in Fig.5.13. The advantage is, that Q_L can be derived without knowing the coupling and eliminate an error source. Also Q_L can be derived much faster, since the cavity does not need to reach equilibrium. Further investigations towards a combined use of SEL and quenchdetection-algorithm should be done. Fig.5.13 shows a fast convergence towards a constant, stable value of Q_L. This means, no use to wait for a flattop, which means a complete saturated cavity, to measure the decay time and calculate the Q_L. Well calibrated ADCs or fast powermeter can be used to realize this method.

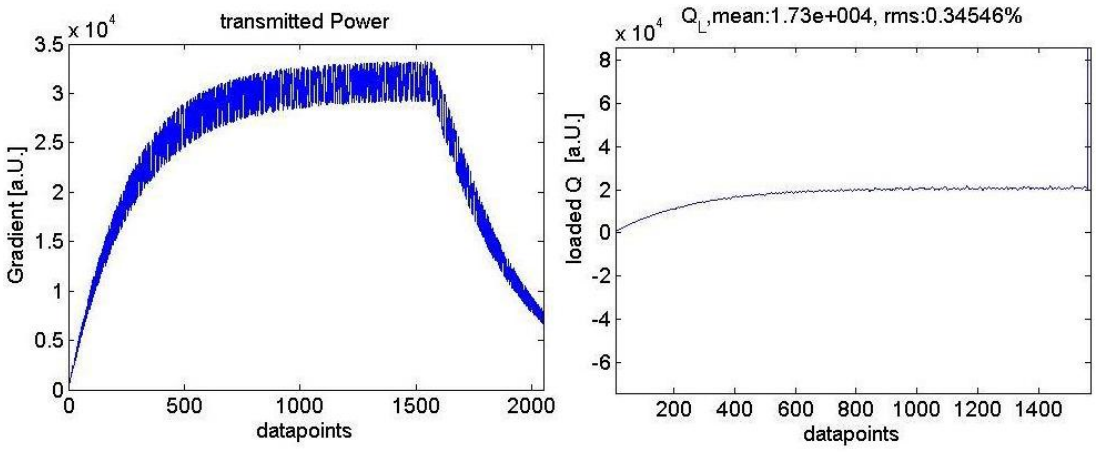


Figure 5.13: The quench detection algorithm used at the vertical test stand.

6 9 mA - run at FLASH

6.1 General interest and goals

The 9 mA run at FLASH investigates some key questions towards XFEL and ILC. First, the accelerator modules 4, 5 and 6 are driven by a single klystron. To understand and improve the control of this system is vital since similar schemes are suggested for future linacs. And a stable gradient of the superconducting cavities with high beam loading need to be achieved. The pulse will consist of 2400 bunches with 3 nC per bunch in a flattop duration of 800 μs, which means a bunch repetition rate of 3 MHz. The beam power should be about 1 GeV and the cavities will be driven close to their quenching limits. The lesson we can learn for XFEL is how to realize such an intense electron beam with a high repetition rate and have maximum flexibility to meet user requirements. This is not easy to achieve since we are working with different currents and bunch repetition rates during the 9 mA run. For ILC, high luminosity, through small beam emittance, high bunch repetition rate and high electron densities, with reliable and stable LLRF is crucial. This meets the key concept in superconducting cavities, that their huge advantage is the acceleration of long bunch trains, as outlined in the introduction.

		XFEL	ILC	FLASH design	FLASH 9 mA Run
Bunch charge	nC	1	3.2	1	3
# bunches		3250*	2625	7200*	2400
Pulse length	μs	650	970	800	800
Current	mA	5	9	9	9

Figure 6.1: Comparison of XFEL, ILC, FLASH and FLASH 9 mA.

The 9 mA run is carried out at FLASH, since by now it is the only machine which can provide the needed experimental and hardware setup. FLASH is an electron linac with design values close of the ILC (Fig. 6.1).

User requirements lead to a change over the years in these values during user operation and this lead to some changes in hardware and controlling. Nevertheless, DESY is a perfect environment for the ILC community, since it will construct and operate XFEL and is also a strong contributor to the TESLA-Collaboration. In section 6.2.1 I will show some results about the stability of the accelerating field along a single module and along FLASH together with a validation of the method to prevent quenches, derived in section 3.4.2. In section 6.2.2. I will concentrate on the method of quench detection as sketched in section 3.4.3.

6.2 Data analysis - Sep. 09 run

6.2.1 Gradient disparities

At ACC456 various hardware is used to steer the driving signal and the Q_L. At ACC 4 and 5 motorized 3-stub-tuners are used while at ACC 6 motorized input couplers (antennas) and phase shifters are the steering elements. Experience has shown, that the solution of ACC 6 can reach the desired state faster since changing the stub positions for a single cavity at ACC4 and 5 has influence onto the phase and Q_L of the other cavities. In Fig. 6.2. all settings at the modules 5 and 6 are given.

ACC5	22.6	MV/m		187	MeV		Max	231	Mev	Δ	44
Pin, MW	1.61		RF power	OK							
Qext	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0			
A, dB	9.67	9.64	9.61	9.53	9.34	9.35	9.38	9.39	measured		
A (klystron)	14.87	14.84	14.81	14.73	14.54	14.55	14.58	14.59			
Pcav, kW	173.5	174.7	175.9	179.2	187.2	186.8	185.5	185.1		1447.8	160
Ecav, MV/m	22.09	22.17	22.24	22.45	22.95	22.92	22.84	22.81		22.6	MV/m
Ecav, max	29	27	28	28	29	28	28	26		27.9	
	Cav 1	Cav 2	Cav 3	Cav 4	Cav 5	Cav 6	Cav 7	Cav 8			

ACC6	26.5	MV/m		220	MeV		Max	238	Mev	Δ	18
Pin, MW	2.18		RF power	OK							
Qext	2.95	2.97	3.00	2.98	3.00	2.98	2.99	2.98	11/21/2007		
A, dB	7.85	7.54	8.16	8.31	12.27	12.03	10.28	10.37	measured		
A, (klystron)	11.65	11.34	11.96	12.11	16.07	15.83	14.08	14.17			
Pcav, kW	357.6	384.0	332.9	321.6	129.2	136.6	204.3	200.2		2066.5	113
Ecav, MV/m	31.82	32.93	30.60	30.12	19.06	19.62	23.99	23.76		26.5	MV/m
Ecav, max	34	32	34	32	21	21	29	26		28.6	
	Cav 1	Cav 2	Cav 3	Cav 4	Cav 5	Cav 6	Cav 7	Cav 8			

Figure 6.2: RF distribution of Klystron 4 for ACC5 and ACC 6.

All cavities have been set to a Q_L of $3{\cdot}10^6$ to ensure, that the cavities have the same filling time and the individual powers have been adjusted in such a way, that the cavities are operating below their quenching limit. The simulations carried out are used to derive the best power distribution with and without beam in such a way that the highest accelerating gradient for the vector sum is reached. If the SASE-settings for FLASH in normal operating mode would be used, the cavities would have quenched, which made the changes necessary. In Fig.6.3 and 6.4 a real time analysis by Brian Chase and in Fig. 6.5 and offline DAQ analysis by myself are shown. Fig. 6.3 shows the behaviour of the

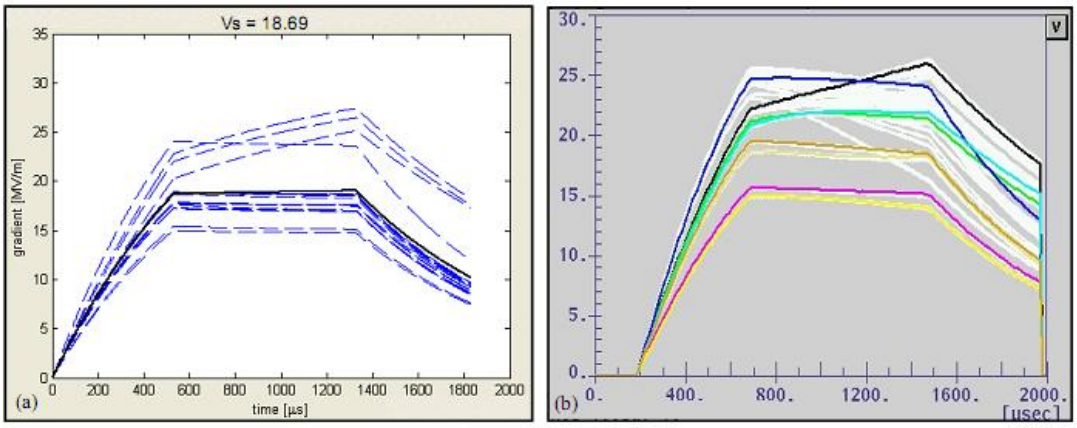


Figure 6.3: Left: Simulation of the cavity behavior with given parameters from the model. Right: real-time LLRF data from 9 mA - no beam run (ACC 6). The fact, that two cavities in the left plot show a positive tilt and not on the right is due to detuning, which was compensated by a resonance frequency shift [9].

cavities in ACC6 without beam. This was the first step in validating the model. Fig. 6.4 shows a direct comparison of real FLASH data with the simulated cavity behavior (21. Sep '09 - 2:50 am). The simulation describes the individual cavities with the ODE derived above with the parameters derived with the method sketched in Section 3.4.2. The beam current was 9 mA, klystron power was reduced for safe operation. The impact of detuning at the end of the pulse should be studied further. The different impacts which occure when we improve from an average vector sum of 20 MV/m to 31.5 MV/m need to be investigated.

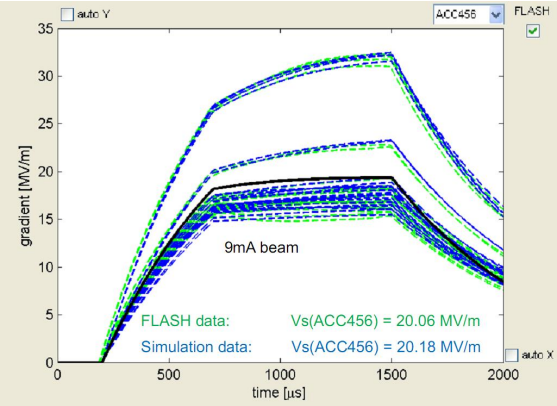


Figure 6.4: Comparison of simulation and FLASH data for rf distribution to reach high V_S and no quench with beam [34].

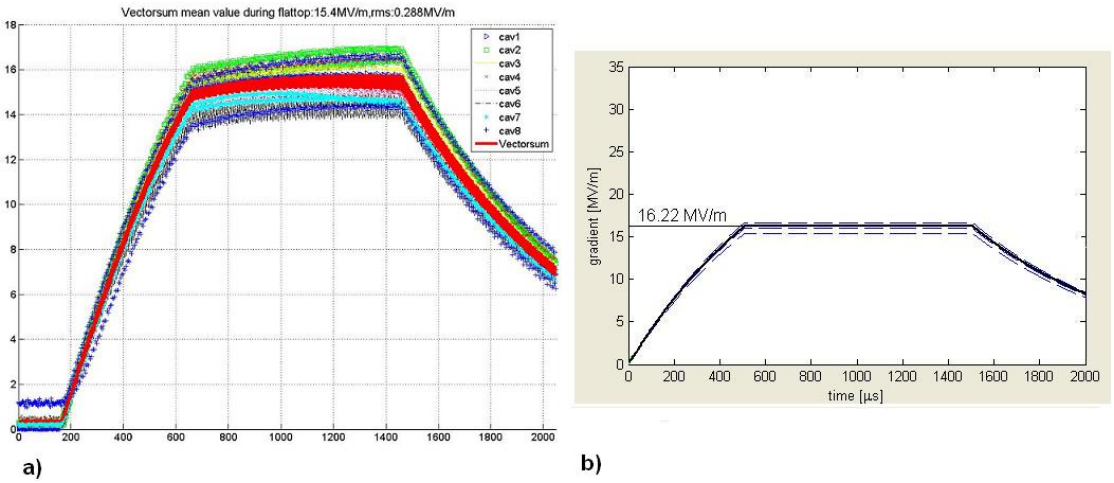


Figure 6.5: a) ACC5 - DAQ data from the 9 mA run. b) is a simulation of the cavity behavior with given parameters from the model - no beam run.

The offline DAQ data used to repeat this study showed some deviations to the real time comparison. The average vector sum during the 9 mA run (19. Sep '09 - 22:30) I consider is 15.4 MV/m with a RMS of 0.28 MV/m in comparison with the simulated vector sum of 16.22 MV/m. The deviations could occur due to a different sampling resolution and frequency for real time and DAQ data. The MATLAB Simulator used for the real time analysis was fine tuned during the run to meet the measured data and has some improvements compared to the Simulator I used. Basically, the model was validated and can be used for further predictions of possible quenching of single cavities and a better tuning of the machine. Higher vector sum gradients needs to be achieved and compared with the model to improve the algorithm. After the discussion of the field gradients of a single pulse, we will now come to the long time stability of the modules. Fig. 6.6 shows the vector sum and the RMS of each module and of FLASH for 422 pulses, each pulse with 9 mA. The deviations from the mean value for ACC 456 is shown in Fig. 6.7. A peak to peak field stability of only 1% was achieved during this run while the RMS is in the range of 0.1%.

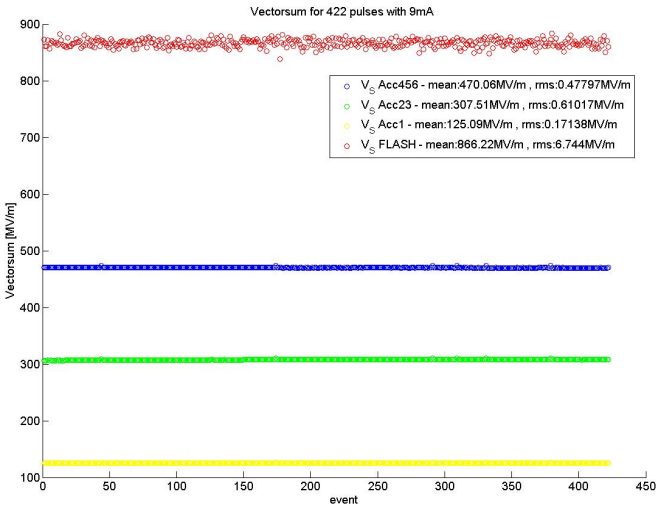


Figure 6.6: Vector sum for all modules plotted against 422 pulses each with 9 mA.

The peak to peak deviation could be explained with microphonics and beam current variations but needs to be investigated further. The vector sum deviation over several hours is shown in Fig. 6.8. The outlined trips were due to klystron problems. The change after $\approx$ 2.5 h is due to a longer bunch train which leads to a higher beam loading.

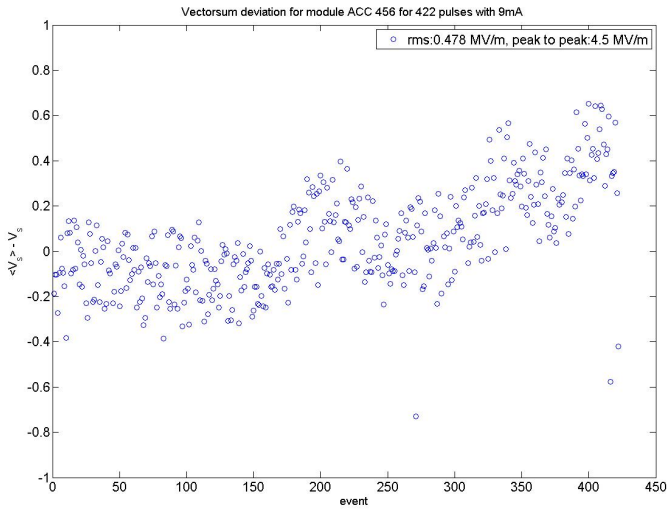


Figure 6.7: Vector sum deviation for module 456 plotted against 422 pulses each with 9 mA. The slope is due to a phase drift in the machine.

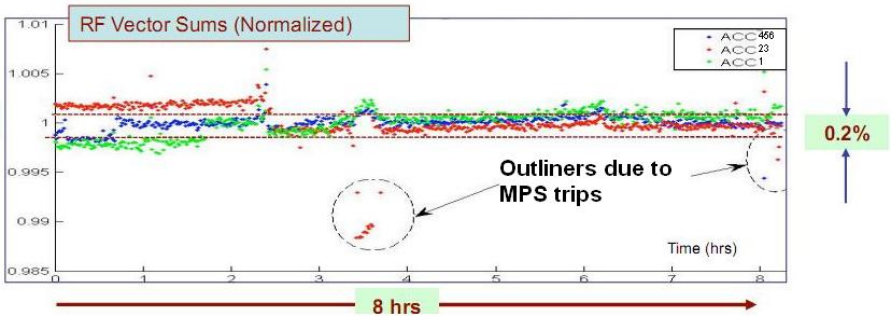


Figure 6.8: Vector sum deviation over longer period of the 9 mA run (each point: 3 mA, 800μs bunch trains) [7].

The correlation of the beam loading effect to the vector sum gradient is shown in Fig. 6.9.

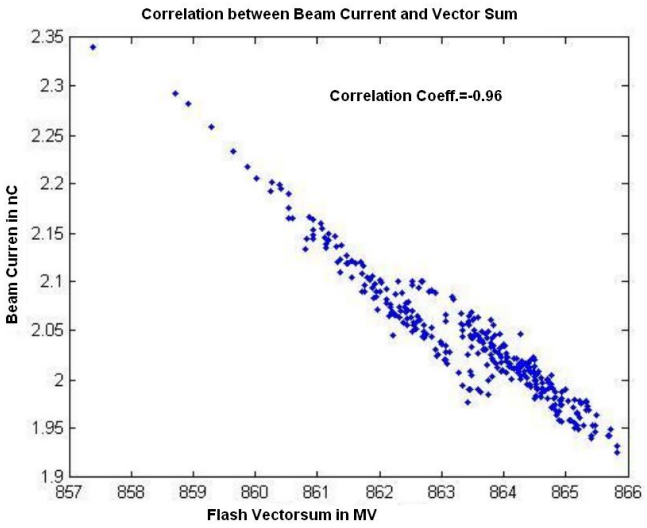


Figure 6.9: Correlation of the beam current to the accelerating gradient. Different working points of the machine correspond with the different linear relation [32].

6.2.2 Quench detection

The quench detection up to now is done via MATLAB scripts, reading out the DAQ and measuring the decay time at the end of the driving pulse. In Fig. 6.10 some results of this method are shown. It works reliable and stable and with an RMS up to 0.2%. A quench is detected when Q_L drops significantly below the design value of $3 \cdot 10^6$. But the algorithm works too slow, which means that a cavity can quench during the pulse. This causes the cryogenic system to react to a high dissipative energy and the pulse could be lost or at least will have a huge intrinsic energy spread. If the dissipated energy is too high, the cryogenic system can become unstable. A longer procedure to recover superconductivity is needed and the whole module needs to be taken out of the power supply chain. This decreases the final beam energy or one has to increase the gradient of the other modules. Exception handling will not be considered and we will focus now on quench detection. The method using the cavity ODE as described in section 3.4.3 can be done in realtime. In Fig. 6.11 a single event of cavity 2 in ACC 6 has been treatend.

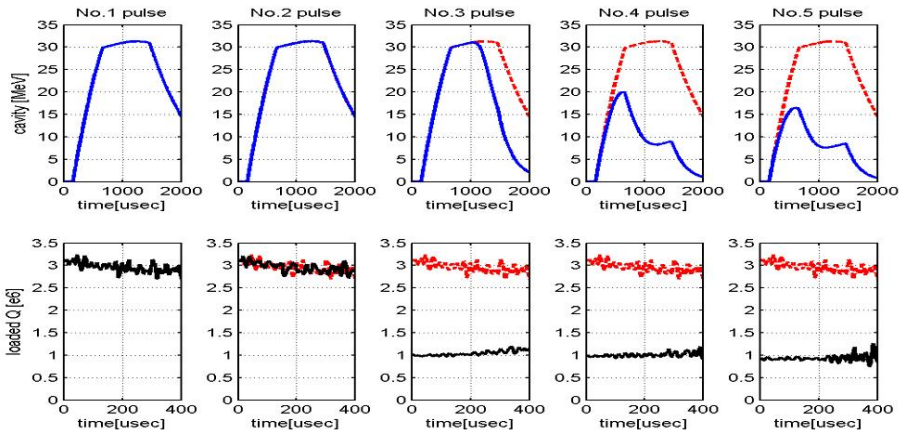


Figure 6.10: In both rows, the red dotted line is always the first graph as reference. The lower row shows the Q_L calculation after rf is turned ($t = 0$ is the point where the rf is turned off after 400μs). A single point in the lower row is a fit of 20μs duration. The dropping of the Q_L by a factor 3 during a quench can be seen [33].

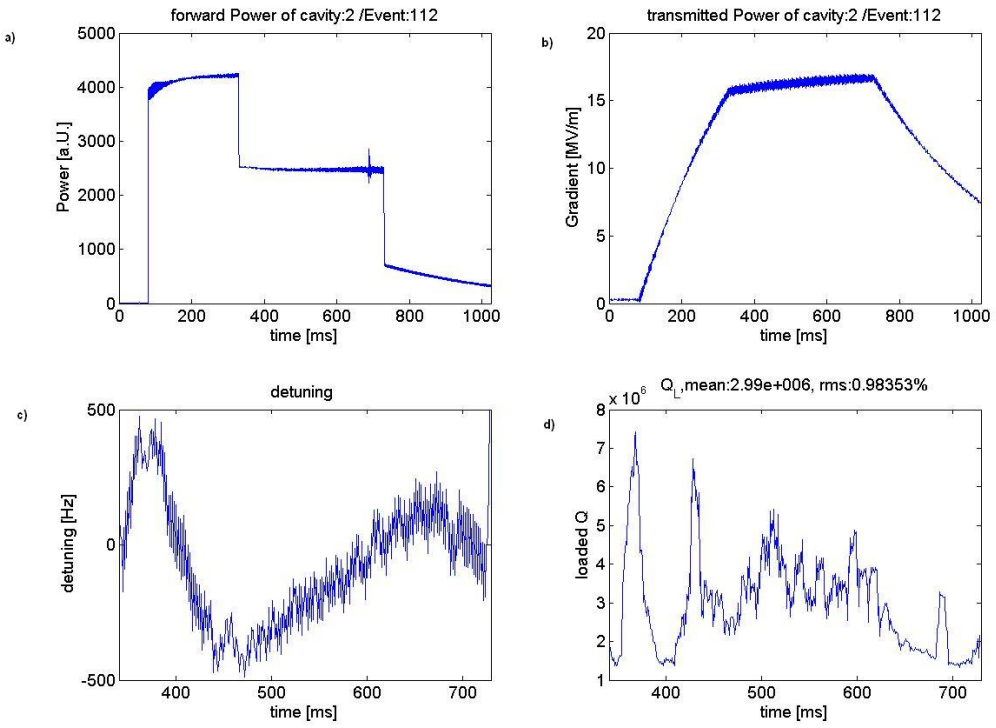


Figure 6.11: a) forward power, b) signal at the pick up antenna. c) detuning and d) Q_L during flattop.

Even that Q_L is noisy, an RMS up to 1% can be achieved and the feasibilty of the algorithm is shown. The huge peaks in Q_L at $\approx$ 200 ms and 420 ms correlate with the detuning, which still influences the calculation. After 420 ms flattop, a peak to peak deviation of $3 \cdot 10^6$ is still to large too identify a quench during flattop. Since this resolution is worse than that of the actual system and can not show a significant drop there is no real benefit. A reason for the noise in the Q_L is the quantization error which is amplified due to the division of the sampling time. The sampling noise can be improved when using higher resolution ADC's, but this is connected to a loss in sampling speed. An upgrade to a sampling frequency from 250 kHz to 54 MHz is planned and could bring significant improvements. Another error source is the calibration of the forward power used here, which is needed to subtract crosstalk in the downconverters. The method is described in [3] and has been developed in [39]. The quench detection method is quite sensitive to the calibration and other calibration methods are under development since the derived crosstalk parameters have large errors. The quench detection method shown here should be studied again when the planned upgrades are done and the calibration procedures are improved and automated. Furthermore, an implementation into an FPGA should be considered to proof the real time capability of this algorithm. Nevertheless, this method shows some nice features and can be studied and used at the vertical test stand.

7 Summary & Outlook

The topic of the diploma thesis are studies to optimize the accelerating gradient of superconducting cavities for the ILC. One part of the thesis took care of the optimization of the vertical test stand to control and assure high gradients during the cavity preparation and assembling. The other part was about obtaining high gradients during operation in the presence of high beam loading and multiple cavity control. A summary of these two parts will be given.

7.1 Discussion of the digital vertical test stand

The feasibility of the digital control system was shown. Furthermore, some plots could be derived and the system is well understood by now. Still, some problems need to be solved, which are more of technical nature than principle problems. The validation of the results and a measurement up to 31.5 MV/m still needs to be done. Principle changes like implementing the SEL mode and using the quench detection algorithm should improve the digital test stand to an easier and faster measurement device than the analog test stand. The digital test stand can be used as a sandbox for controller algorithms, since the environment is the same like in a controller. Lessons, relying on algorithm speed concerning the small cavity bandwith, have been learned and an SEL is the mode of choice when dealing with this. The higher sampling rate, and thus a higher data flow, of 81 MHz comparing to the 1 MHz at FLASH needed to be handled which could help at the planned upgrade at FLASH to a sampling rate of 54 MHz. The development of the rf box is also done and it behaves as it should and can be used for the test stand or at any other set up, where two or three phase stable, synchronized frequencies are used.

7.2 Discussion of the 9 mA run

The goal of the 9 mA run was long bunch trains with a current of 9 mA and a beam energy RMS of 0.1%. Resulting from the fact that a two week shutdown for major upgrades and hardware changes to carry out the run, were done before, normal start up problems arise and shortend the up time of the accelerator. The former goal to recover the results of the last run took much longer than planned due to different interlock trips and tuning procedures needed. This lead to the fact, that only minor goals could be considered. But nevertheless 2400 bunches, each with merely 3 nC and a repetition rate of 3 MHz, was achieved during the nearly last shift (see Fig. 7.1). But the demonstration

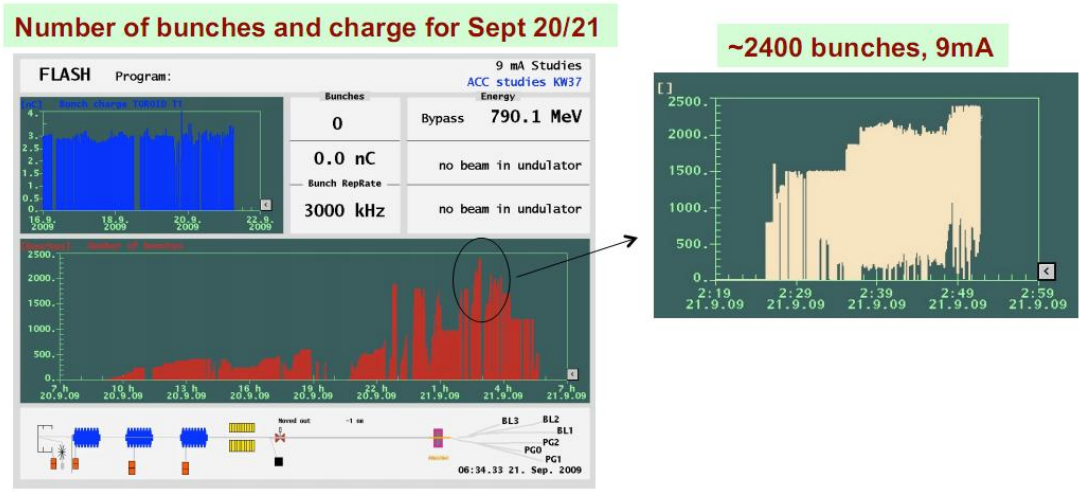


Figure 7.1: Main goal achieved: 2400 Bunche at 3 MHz, each 3nC. [7]

of the field stability up to 0.1%, operation close the the quenching limits, integrating the 3.9 GHz module and several other goals could not be achieved and need to be done in future studies carried out. The 9 mA run is an important accelerator study for the accelerator community since it addressed many challenging machine problems dealing with intense beam, high gradients, stable beam energy and high repetition rates. This is needed for future linear colliders and FELs no matter what the design will be.

Acknowledgements

First of all I like to thank Prof. Quadt and Prof. Elsen who made this work possible. Then I have to say a big 'Thank You' to Sven Karstensen and Wojciech Jalmuzna. Without their work, support and discussions this thesis would have never ended or even started. I also thank Torsten Külper for his technical support and the construction of anything I could wish. Of course everyone of the MHF-sl group and from Hall III, especially Jürgen Eschke who supported me with helium whenever he could and the people from the MSK group.

8 Appendix

8.1 sweeptest2.m

```
function varargout = sweeptest2(varargin)
% Designed to control digital System at Vertical Teststand II
% Delevoped by:
% Marc Wenskat @ DESY ; University of Goettingen
% Needs special files located in folder.
% See dependency report -> 'Tools'
% -> 'Save and show dependency report'
% tcpip_ and ii_*_*.mexsol for communication with GPIB-Devices, DOOCS-Server
  % or with FPGA and several m-script
  % (uni_daq.m,uni_daq_once.m,uni_gen.m,sweep_function.m).
  %
  % Edit the above text to modify the response to help sweeptest2
  % Last Modified by GUIDE v2.5 12-Aug-2009 15:26:26
  % Begin initialization code - DO NOT EDIT
  gui_Singleton = 1;
  gui_State = struct('gui_Name', mfilename, ...
  'gui_Singleton', gui_Singleton, ...
  'gui_OpeningFcn', @sweeptest2_OpeningFcn, ...
  'gui_OutputFcn', @sweeptest2_OutputFcn, ...
  'gui_LayoutFcn', [] , ...
  'gui_Callback', []);
  if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
  end
  if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
  else
    gui_mainfcn(gui_State, varargin{:});
  end
```

```
   function sweeptest2_OpeningFcn(hObject, eventdata, handles, varargin)
% Choose default command line output for sweeptest2
   handles.output = hObject;
   % Update handles structure
   guidata(hObject, handles);
   t=timer('TimerFcn',{@read_Callback,handles},'Period',2.0,'TasksToExecute',Inf,...
   'ExecutionMode','fixedRate');
   start(t);
   set(handles.AW1,'String',0);
   set(handles.ps1,'String',0);
   set(handles.eacc,'String',0);
   set(handles.qzero,'String',0);
   set(handles.beta,'String',0);
   set(handles.tau,'String',0);
   set(handles.resonancefrequency,'String',0);
   set(handles.ptop,'String',0);
   set(handles.clockfreq,'String',0);
   set(hObject,'toolbar','figure');
   set(handles.dpf,'String',0);
   set(handles.dpt,'String',0);
   set(handles.dpr,'String',0);
   set(handles.dloop,'String',0);
   set(handles.dcable,'String',0);
   set(handles.ctemp,'String',0);
   set(handles.feldem,'String',0);
   setappdata(0 , 'dpr' , 0);
   setappdata(0 , 'dpt' , 0);
   setappdata(0 , 'dpf' , 0);
   setappdata(0 , 'phase' , 0);
   setappdata(0 ,'dhom1' , 0);
   setappdata(0 ,'dhom2' , 0);
   quenchtable=zeros(9,9);
   setappdata(0,'quenchtable',quenchtable);
   function varargout = sweeptest2_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function unigen_Callback(hObject, eventdata, handles)
a1=str2double(get(handles.AW1,'String'));
if (a1>0) %&& a1<3
```

```
   a12=a1;%3680.3465+2498.168*log(a1-0.28722); %[W] to IQ
else a12=0;
end
ptop=str2double(get(handles.ptop,'String'));
adcfreq=str2double(get(handles.clockfreq,'String'));
p_length=str2double(get(handles.ps1,'String'));
div = double(ii_get_word('WORD_TIMING_FREQ_STR')) ;
time_base = 1/(div/(adcfreq*1e6))/512 ;
clk_counts = floor(ptop*adcfreq*1e6/9)*9+8 ;
ff_i = [ 0 a12 0] ;
ff_q = [ 0 0 0]*0000 ;
gp_i = [ 0 0 0]*0 ;
prog = [1 p_length*time_base 2^18-1] ;
gp_q = [ 0 -1 0]*256 ;
mode = [0 1 0 0 0];
ii_set_word('WORD_TIMING_FREQ_TRG',uint32(clk_counts));
ii_set_area('AREA_FF_I',uint32(0),uint32(length(ff_i)),uint32(ff_i)) ;
ii_set_area('AREA_FF_Q',uint32(0),uint32(length(ff_q)),uint32(ff_q)) ;
ii_set_area('AREA_GP_I',uint32(0),uint32(length(gp_i)),uint32(gp_i)) ;
ii_set_area('AREA_GP_Q',uint32(0),uint32(length(gp_q)),uint32(U22Dec(gp_q,18))) ;
ii_set_area('AREA_PROG',uint32(0),uint32(length(prog)),uint32(prog)) ;
ii_set_area('AREA_MODE',uint32(0),uint32(length(mode)),uint32(mode)) ;
function AW1_Callback(hObject, eventdata, handles)
function AW1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function ps1_Callback(hObject, eventdata, handles)
function ps1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function resetpuls_Callback(hObject, eventdata, handles)
ff_i = [ 0 0 0 ]*10000 ;
ff_q = [ 0 0 0 ]*0000 ;
gp_i = [ 0 0 0 ]*0 ;
prog = [0 0 0 2^18-1] ;
ii_set_area('AREA_FF_I',uint32(0),uint32(length(ff_i)),uint32(ff_i)) ;
```

```
ii_set_area('AREA_FF_Q',uint32(0),uint32(length(ff_q)),uint32(ff_q)) ;
ii_set_area('AREA_GP_I',uint32(0),uint32(length(gp_i)),uint32(gp_i)) ;
ii_set_area('AREA_GP_Q',uint32(0),uint32(length(gp_i)),uint32(gp_i)) ;
ii_set_area('AREA_PROG',uint32(0),uint32(length(prog)),uint32(prog)) ;
function quit_Callback(hObject, eventdata, handles)
rmappdata(0,'cavity');
rmappdata(0,'dpf');
rmappdata(0,'dpr');
rmappdata(0,'dpt');
rmappdata(0 , 'phase');
rmappdata(0 ,'dhom1');
rmappdata(0 ,'dhom2');
delete(handles.figure1)
function ctemp_Callback(hObject, eventdata, handles)
function ctemp_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function hef_Callback(hObject, eventdata, handles)
function hef_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function flow_Callback(hObject, eventdata, handles)
function flow_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function vcav_Callback(hObject, eventdata, handles)
function vcav_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function read_Callback(hObject, eventdata, handles)
[data2,err]=ttfr('TTF.KRYO/TTFKEITHLEY1/1023/VALUE'); % Fullstand
if (data2<80),
   set(handles.hef,'BackgroundColor','red');
else set(handles.hef,'BackgroundColor','white');
```

```
end
data2=num2str(data2,'%4.2f');
set(handles.hef,'String',data2);
[data,err]=ttfr('TTF.KRYO/CALC/V_2.TEMPERATURE/CALC'); % Cavity Temp
data=num2str(data,'%4.2f');
set(handles.ctemp,'String',data);
[data3,err]=ttfr('TTF.KRYO/CALCULATOR/V_2.FLOW/CALC'); % Flow
if (data3>9),
   set(handles.flow,'BackgroundColor','red');
else set(handles.flow,'BackgroundColor','white');
end
data3=num2str(data3,'%3.2f');
set(handles.flow,'String',data3);
[data4,err]=ttfr('TTF.VAC/PENNING/V_2.8/P'); % Vacuum in Cavity
if (data4>10^-6),
   set(handles.vcav,'BackgroundColor','red');
else set(handles.vcav,'BackgroundColor','white');
end
data4=num2str(data4,'%3.2e');
set(handles.vcav,'String',data4);
[data,err]=ttfr('TTF.RF/UNIDOS/V2/X_RAY'); % Feldemission
data=num2str(data,'%2.1e');
set(handles.feldem,'String',data);
function resonancefrequency_Callback(hObject, eventdata, handles)
function resonancefrequency_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function beta_Callback(hObject, eventdata, handles)
function beta_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function tau_Callback(hObject, eventdata, handles)
function tau_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
```

```
function sliderValue_editText_Callback(hObject, eventdata, handles)
sliderValue = get(handles.sliderValue_editText,'String');
sliderValue = str2num(sliderValue);
if (isempty(sliderValue) —— sliderValue < -180 —— sliderValue > 180)
   set(handles.slider,'Value',0);
   set(handles.slider_editText,'String',0);
else
   set(handles.slider,'Value',sliderValue);
end
function sliderValue_editText_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function slider_Callback(hObject, eventdata, handles)
sliderValue = get(handles.slider,'Value');
set(handles.sliderValue_editText,'String', num2str(sliderValue));
guidata(hObject, handles);
function slider_CreateFcn(hObject, eventdata, handles)
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor',[.9 .9 .9]);
end
function setmat_Callback(hObject, eventdata, handles)
angle=str2num(get(handles.sliderValue_editText,'String'));
I = 1*cos(angle*2*pi/360)*65535 ;
Q = 1*sin(angle*2*pi/360)*65535 ;
ii_set_area('WORD_ROT_RE',uint32(0),uint32(1),uint32(U22Dec(I,18))) ;
ii_set_area('WORD_ROT_IM',uint32(0),uint32(1),uint32(U22Dec(Q,18))) ;
function ptop_Callback(hObject, eventdata, handles)
function ptop_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function savedata_Callback(hObject, eventdata, handles)
ptop=str2double(get(handles.ptop,'String'));
pause(ptop)
a= uni_daq([0 1 2 8 9 10],1);
pt=abs(complex(a(:,1),a(:,4)));
pr = abs(complex(a(:,2),a(:,5)));
```

```
pf = abs(complex(a(:,3),a(:,6)));
d=datestr(now,'mmddHHMM');
savefile1=strcat('pt',d,'.m');
savefile2=strcat('pr',d,'.m');
savefile3=strcat('pf',d,'.m');
save(savefile1,'pt','-ascii','-tabs') ;
save(savefile2,'pr','-ascii','-tabs') ;
save(savefile3,'pf','-ascii','-tabs') ;
function eacc_Callback(hObject, eventdata, handles)
function eacc_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function qzero_Callback(hObject, eventdata, handles)
function qzero_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function measure_Callback(hObject, eventdata, handles)
ptop=str2double(get(handles.ptop,'String'));
fid=tcpip_open('ttfvert2.desy.de',5555);
s='0;10;0;:CALC:MARK:MAX \';
tcpip_write(fid,s);
tcpip_close(fid);
fid=tcpip_open('ttfvert2.desy.de',5555);
p='0;10;1;:CALC:MARK:X? \';
tcpip_write(fid,p);
pause(0.1);
datax=tcpip_readln(fid,50);
tcpip_close(fid);
set(handles.xvalue,'String',datax);
pause(ptop)
a= uni_daq([0 1 2 8 9 10],1);
ptadc = abs(complex(a(:,1),a(:,4)));
pradc = abs(complex(a(:,2),a(:,5)));
pfadc = abs(complex(a(:,3),a(:,6)));
%pf=exp(1.73257e-9.*pfadc.^2+3.66329e-5.*pfadc-2.12135);
%pr=exp(-9.24249e-7.*pradc.^2+0.00399.*pradc-7.54021);
```

```
%pt=exp(8.70431e-9.*ptadc.^2-2.57523e-4.*ptadc-4.71769);
dpt=getappdata(0,'dpt');
dpr=getappdata(0,'dpr');
dpf=getappdata(0,'dpf');
pt=ptadc+dpt;
pr=pradc+dpr;
pf=pfadc+dpf;
daqtime=str2double(get(handles.daqtime,'String'));
daqoffset=str2double(get(handles.daqoffset,'String'));
datapoints=double(ii_get_word('WORD_DAQ_LIMIT'));
deltat=daqtime/(datapoints-1);
pulselength=str2double(get(handles.ps1,'String'));
n=round((pulselength/3-daqoffset)/deltat); %obere Grenze erster Peak bei pr
mo=round((pulselength+1-daqoffset)/deltat); %obere Grenze zweiter Peak bei pr
o=round((pulselength-daqoffset)/deltat - 100); %untere Grenze zweiter Peak bei pr
q=round((2*pulselength/3-daqoffset)/deltat); %beginn des Flat tops bei pt
r=round((pulselength-daqoffset)/deltat - 100); %ende des flat tops bei pt
v=round((2*pulselength/3-daqoffset)/deltat); %beginn des flattops bei pr
w=round((pulselength-daqoffset)/deltat - 100); %ende des flattops bei pr
u=floor((pulselength-daqoffset)/deltat ) ; %beginn decay fuer tau
ptt=mean(pt(q:r));
deltaptt=std(ptt);
ptexp=mean(pt(u-10:u+10))/exp(1);
stoptau=round((pulselength+pulselength/3)/deltat);
[rte,index]=min(abs(pt(u:stoptau)-ptexp));
d=index+u-1; %ende decay fuer tau
indexx=d-u+1;
p1=max(pr(1:n)); %erster peak pr
pe=max(pr(o:mo)); %zweiter peak pr
prt=mean(pr(v:w)); %flattop pr
errorprt=std(pr(v:w));
pft=mean(pf(v:w));
errorpft=std(pf(v:w));
x(1:indexx)=log(pt(u:d)); %zweiter peak, bestimmen von tau
beta2=pe/(p1-prt) ; % bei cw
beta3=1/((1-sqrt(prt/pft))/(1+sqrt(prt/pft)));
deltabeta2=(pe/(p1-prt)^2)*errorprt;
deltabeta3=sqrt(((sqrt(prt/pft)+1)/(2*pft*sqrt(prt/pft)*(1-sqrt(prt/pft))^2)+
```

```
1/(2*pft*sqrt(prt/pft)*(1-sqrt(prt/pft))))^2*errorprt^2+((prt*(1+sqrt(prt/pft)))/(2*pft^2*sqrt(prt
*(1-sqrt(prt/pft))^2)-prt/(2*pft^2*sqrt(prt/pft)*(1-sqrt(prt/pft))))^2*errorpft^2);
vecbeta=[beta2 beta3];
beta=mean(vecbeta);
deltabeta=sqrt((deltabeta2/2)^2+(deltabeta3/2)^2);
topt=(length(x)-1)*deltat;
t=[0:deltat:topt];
[p,S]=polyfit(t,x,1);
m=p(1);
R=S.R;
d2=(R'*R)\eye(1+1);
d2=diag(d2);
MSE=(S.normr^2)/S.df;
se=sqrt(MSE*d2);
deltam=se(1);
tau=-1/m;
deltatau=deltam/m^2;
resfreq=str2double(get(handles.xvalue,'String'));
qload=2*pi*resfreq*tau;
deltaqload=2*pi*resfreq*deltatau;
mpf=mean(pf);
deltampf=std(mpf);
pd=(4*beta/(1+beta)^2)*mpf-prt-ptt;
deltapd=sqrt(deltaptt^2+((mpf*16*(beta-1)^2)/(beta+1)^6)^
2*deltabeta^2+((16*beta^2)/(beta+1)^4)^2*deltampf^2);
q0=qload*(1+beta);
deltaqo=sqrt(deltaqload^2*(1+beta)^2+deltabeta^2*qload^2);
n=get(handles.nrcells,'String');
l=0.1154;
Eacc=sqrt(1030*q0*pd)/(n*l);
deltaeacc=0.5*sqrt(1030)*sqrt((pd^2*deltaqo^2+q0^2*deltapd^2)/(q0*1.035^2*pd)) ;
set(handles.deltaqload,'String',num2str(deltaqload,'%5.3e'));
set(handles.deltaqzero,'String',num2str(deltaqo,'%5.3e'));
set(handles.deltaeacc,'String',num2str(deltaeacc/1e6));
set(handles.deltatau,'String',num2str(deltatau));
set(handles.resonancefrequency,'String',num2str(resfreq/1e6));
set(handles.qload,'String',num2str(qload,'%5.3e'));
set(handles.qzero,'String',num2str(q0,'%5.3e'));
```

```
set(handles.eacc,'String',num2str(Eacc/1e6));
set(handles.beta,'String',num2str(beta));
set(handles.tau,'String',num2str(tau));
set(handles.betadelta,'String',deltabeta);
%% QPS wird ab hier getestet!!!
z_pt=a(:,1)+a(:,4)*j;
z_pf=a(:,3)+a(:,6)*j;
pfphase=(phase(complex(a(:,3),a(:,6)))*(180/pi));
ptphase=(phase(complex(a(:,1),a(:,4)))*(180/pi));
pt=abs(complex(a(:,1),a(:,4)));
pf=abs(complex(a(:,3),a(:,6)));
omega_0=1.3e9;
f_sample=(str2double(get(handles.clockfreq,'String')));
daqtime=str2double(get(handles.daqtime,'String'));
delta=daqtime/2047;
t=0:delta:daqtime;
%% Ableitung bestimmen und averaging Window anwenden
dt=1/(f_sample*1e6);
dv(1:2047)=(pt(2:2048)-pt(1:2047));
dphiprobe=(ptphase(2:2048)-ptphase(1:2047));
dvdt=dv/dt;
dphiprobedt=dphiprobe;
meandvdt=slidingavg(dvdt,16);
%% Detuning berechnen
for i=1:2047
   detuning(i)=(1/(2*pi))*(dphiprobedt(i)-(1.3e9/3e6)*sin(pfphase(i)-ptphase(i))*...
   (abs(pf(i))/abs(pt(i))));
end
%% DGL nach loaded Q aufloesen/berechnen/plotten
for i=1:2047
   z_loaded_Q(i)=-0.5*(omega_0*(2*z_pf(i)-z_pt(i)))/(z_pt(i)*detuning(i)*j-meandvdt(i));
end
loaded_Q=abs(z_loaded_Q);
testqpsplot=figure+1;
subplot(2,3,1)
plot(t,pf)
title('forward Power')
xlabel('time [s]');
```

```
ylabel('Gradient ')
subplot(2,3,2)
plot(t,pt)
title('transmitted Power')
xlabel('time [s]');
ylabel('Gradient ')
subplot(2,3,3)
plot(t(1:2047),meandvdt')
title('dV_c/dt - Sliding Average')
xlabel('time [s]');
subplot(2,3,4)
plot(t(1:2047),slidingavg(detuning,10)')
ylim([-100 100]);
title('detuning ')
xlabel('time [s]');
ylabel('detuning [Hz]')
subplot(2,3,5)
mittelwert=mean(loaded_Q);
error=std(loaded_Q);
error_ql=error/mittelwert;
plot(t(1:2047),slidingavg(loaded_Q,10))
Title=strcat('loaded Q ','mean Value: ',num2str(mittelwert,'%0.3g'),' rms is : ',num2str(error_ql),'%'
title(Title)
ylabel('loaded Q');
xlabel('time [s]');
function plotAxes2_pushbutton_Callback(hObject, eventdata, handles)
ii_set_bits('BIT_DAQ_ONCE',uint32(1))
ptop=str2double(get(handles.ptop,'String'));
axes(handles.axes2);
pr=uni_daq([1 9],1);
pradc=abs(complex(pr(:,1),pr(:,2)));
daqtime=str2double(get(handles.daqtime,'String'));
delta=daqtime/2047;
t=0:delta:daqtime;
dpr=getappdata(0,'dpr');
pr=exp(-9.24249e-7.*pradc.^2+0.00399.*pradc-7.54021);
pr=pr+dpr;
plot(t,pradc,'r') %pr
```

```
title('reflected Power');
xlabel('t [s]');
ylabel('Power [W]')
ii_set_bits('BIT_DAQ_ONCE',uint32(0))
guidata(hObject, handles);
function plotAxes3_pushbutton_Callback(hObject, eventdata, handles)
ptop=str2double(get(handles.ptop,'String'));
pause(ptop)
axes(handles.axes3);
pf = uni_daq([2 10],1) ;
pfadc =abs(complex(pf(:,1),pf(:,2)));
daqtime=str2double(get(handles.daqtime,'String'));
delta=daqtime/2047;
t=0:delta:daqtime;
dpf=getappdata(0,'dpf');
pf=exp(1.73257e-9.*pfadc.^2+3.66329e-5.*pfadc-2.12135);
pf=pf+dpf;
plot(t,pfadc,'g')
title('forward Power');
xlabel('t [s]');
ylabel('Power [W]')
guidata(hObject, handles);
function plotAxes1_pushbutton_Callback(hObject, eventdata, handles)
ii_set_bits('BIT_DAQ_ONCE',uint32(1))
ptop=str2double(get(handles.ptop,'String'));
a= uni_daq([0 1 2 8 9 10],1);
ptadc = abs(complex(a(:,1),a(:,4)));
pradc = abs(complex(a(:,2),a(:,5)));
pfadc = abs(complex(a(:,3),a(:,6)));
daqtime=str2double(get(handles.daqtime,'String'));
delta=daqtime/2047;
t=0:delta:daqtime;
delta=daqtime/2047;
axes(handles.axes1);
plot(t,ptadc);
title('transmitted Power');
xlabel('t [s]');
ylabel('Power');
```

```
axes(handles.axes3);
plot(t,pfadc,'g') ;
title('forward Power');
xlabel('t [s]');
ylabel('Power [W]');
axes(handles.axes2);
plot(t,pradc,'r') ;
title('reflected Power');
xlabel('t [s]');
ylabel('Power [W]') ;
ii_set_bits('BIT_DAQ_ONCE',uint32(0))
function clockfreq_Callback(hObject, eventdata, handles)
function clockfreq_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function daqtime_Callback(hObject, eventdata, handles)
function daqtime_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function daqoffset_Callback(hObject, eventdata, handles)
function daqoffset_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function feldem_Callback(hObject, eventdata, handles)
function feldem_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function betadelta_Callback(hObject, eventdata, handles)
function betadelta_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function dpr_Callback(hObject, eventdata, handles)
function dpr_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function dpf_Callback(hObject, eventdata, handles)
function dpf_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function dpt_Callback(hObject, eventdata, handles)
function dpt_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function dloop_Callback(hObject, eventdata, handles)
function dloop_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function dcable_Callback(hObject, eventdata, handles)
function dcable_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function dodamping_Callback(hObject, eventdata, handles)
dforward=str2double(get(handles.dpf,'String'));
dreflected=str2double(get(handles.dpr,'String'));
dtransmitted=str2double(get(handles.dpt,'String'));
dloop=str2double(get(handles.dloop,'String'));
dcable=str2double(get(handles.dcable,'String'));
dloopr=(dcable-dloop)/2;
ddforwardr=20-dforward-dcable-0.61;
ddreflectedr=20-dreflected-dcable-0.61;
ddtransmittedr=20-dtransmitted-dcable-dloopr-0.61;
%fpf=str2double(get(handles.fpf,'String'));
%fpr=str2double(get(handles.fpr,'String'));
%fpt=str2double(get(handles.fpt,'String'));
k=get(handles.homyesno,'Value');
if k==1
```

```
   dhom1=str2double(get(handles.hom1damping,'String'));
   dhom2=str2double(get(handles.hom2damping,'String'));
   ddhom1=20-dhom1-dcable-0.61;
   ddhom2=20-dhom2-dcable-dloopr-0.61;
   dhom_1=(10^(ddhom1/10))*10^(-3); % Unit is [W]
   dhom_2=(10^(ddhom2/10))*10^(-3); % Unit is [W]
   setappdata(0,'dhom1',dhom_1);
   setappdata(0,'dhom1',dhom_2);
end
if ddforwardr==0 %&& fpf ==0
   dpf=0;
else dpf=(10^(ddforwardr/10))*10^(-3)%+10^(fpf/10); % Unit is [W]
end
if ddreflectedr==0 %&& fpr ==0
   dpr=0;
else dpr=(10^(ddreflectedr/10))*10^(-3)%+10^(fpr/10); % Unit is [W]
end
if ddtransmittedr==0 %&& fpt ==0
   dpt=0;
else dpt=(10^(ddtransmittedr/10))*10^(-3)%+10^(fpt/10); % Unit is [W]
end
setappdata(0,'dpf',dpf);
setappdata(0,'dpr',dpr);
setappdata(0,'dpt',dpt);
function fpr_Callback(hObject, eventdata, handles)
function fpr_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function fpf_Callback(hObject, eventdata, handles)
function fpf_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function fpt_Callback(hObject, eventdata, handles)
function fpt_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
```

```
end
function setdaq_Callback(hObject, eventdata, handles)
adcfreq=str2double(get(handles.clockfreq,'String')); %DAQ-Timing
div = double(ii_get_word('WORD_TIMING_FREQ_STR')) ;
daqtime=str2double(get(handles.daqtime,'String'));
daqoffset=str2double(get(handles.daqoffset,'String'));
datapoints=double(ii_get_word('WORD_DAQ_LIMIT'));
daq_freq=uint32(daqtime*adcfreq*1000000/div/datapoints);
daq_offs=uint32(daqoffset*adcfreq*1000000/div/daq_freq);
ii_set_word('WORD_DAQ_FREQ',daq_freq) ;
ii_set_word('WORD_DAQ_OFFSET',daq_offs) ;
function freq_gen_Callback(hObject, eventdata, handles)
function freq_gen_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function amp_gen_Callback(hObject, eventdata, handles)
function amp_gen_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function setfreqgen_Callback(hObject, eventdata, handles)
fid=tcpip_open('ttfvert2.desy.de',5555);
adresse='0;1;0;';
freq=get(handles.freq_gen,'String');
amp=get(handles.amp_gen,'String');
freqz=[':FREQ' ' ' freq];
ampz=[';:POW:AMPL' ' ' amp];
s=strcat(adresse,freqz, ' MHz', ampz ,' dBm \ ');
tcpip_write(fid,s);
tcpip_close(fid);
function rfon_Callback(hObject, eventdata, handles)
fid=tcpip_open('ttfvert2.desy.de',5555);
s='0;1;0;:OUTP:STAT ON \';
tcpip_write(fid,s);
tcpip_close(fid);
function rfoff_Callback(hObject, eventdata, handles)
fid=tcpip_open('ttfvert2.desy.de',5555);
```

```
s='0;1;0;:OUTP:STAT OFF \';
tcpip_write(fid,s);
tcpip_close(fid);
function modefreq_Callback(hObject, eventdata, handles)
function modefreq_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function calculate_Callback(hObject, eventdata, handles)
f_res=str2num(get(handles.modefreq,'String'));
ifreq = double(uint32(f_res/144*10000))*100;
f1 = ifreq*9/1000000 ;
f2 = ifreq*144/1000000 ;
f3 = ifreq*143/1000000 ;
f1=num2str(f1,'%4.2f');
f2=num2str(f2,'%8.4f');
f3=num2str(f3,'%8.4f');
set(handles.freq_gen,'String',f2);
set(handles.clockfreq,'String',f1);
set(handles.convfreq,'String',f3);
freq=get(handles.freq_gen,'String');
leftspan=str2double(freq)-0.5 ;
rightspan=str2double(freq)+0.5 ;
set(handles.leftspan,'String',num2str(leftspan));
set(handles.rightspan,'String',num2str(rightspan));
function adcfreq_Callback(hObject, eventdata, handles)
function adcfreq_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function convfreq_Callback(hObject, eventdata, handles)
function convfreq_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function setmarker_Callback(hObject, eventdata, handles)
fid=tcpip_open('ttfvert2.desy.de',5555);
s='0;10;0;:CALC:MARK:MAX \';
```

```
tcpip_write(fid,s);
tcpip_close(fid);
function xvalue_Callback(hObject, eventdata, handles)
function xvalue_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function yvalue_Callback(hObject, eventdata, handles)
function yvalue_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function getmarker_Callback(hObject, eventdata, handles)
fid=tcpip_open('ttfvert2.desy.de',5555);
p='0;10;1;:CALC:MARK:X? \';
tcpip_write(fid,p);
pause(0.1);
datax=tcpip_readln(fid,50);
tcpip_close(fid);
set(handles.xvalue,'String',datax);
fid=tcpip_open('ttfvert2.desy.de',5555);
q='0;10;1;:CALC:MARK:Y? \';
tcpip_write(fid,q);
pause(0.1);
datay=tcpip_readln(fid,1024);
set(handles.yvalue,'String',datay);
tcpip_close(fid);
function centerfreq_Callback(hObject, eventdata, handles)
function centerfreq_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function leftspan_Callback(hObject, eventdata, handles)
function leftspan_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function rightspan_Callback(hObject, eventdata, handles)
```

```
function rightspan_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function setcenter_Callback(hObject, eventdata, handles)
fid=tcpip_open('ttfvert2.desy.de',5555);
centfreq=get(handles.centerfreq,'String');
centfreqz=[':FREQ:CENT' ' ' centfreq];
s=strcat('0;10;0;',centfreqz, ' MHz \');
tcpip_write(fid,s);
tcpip_close(fid);
function setspan_Callback(hObject, eventdata, handles)
fid=tcpip_open('ttfvert2.desy.de',5555);
leftspan=get(handles.leftspan,'String');
rightspan=get(handles.rightspan,'String');
leftspanz=[':FREQ:STAR' ' ' leftspan];
rightspanz=[';:FREQ:STOP' ' ' rightspan];
s=strcat('0;10;0;',leftspanz , ' MHz',rightspanz , ' MHz \');
tcpip_write(fid,s);
tcpip_close(fid);
function freqlimit_Callback(hObject, eventdata, handles)
function freqlimit_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function setlimit_Callback(hObject, eventdata, handles)
val = str2num(get(handles.freqlimit,'String'));
ii_set_word('WORD_FREQ_LIMIT',uint32(val));
function qload_Callback(hObject, eventdata, handles)
function qload_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function homyesno_Callback(hObject, eventdata, handles)
k=get(handles.homyesno,'Value');
if k==1
   set(handles.hom1damping,'BackgroundColor','white');
   set(handles.hom2damping,'BackgroundColor','white');
```

```
end
if k==0
   set(handles.hom1damping,'BackgroundColor','black');
   set(handles.hom2damping,'BackgroundColor','black');
end
function hom2damping_Callback(hObject, eventdata, handles)
function hom2damping_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','grey');
end
function hom1damping_Callback(hObject, eventdata, handles)
function hom1damping_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','grey');
end
function savemeasure_Callback(hObject, eventdata, handles)
new1=zeros(1,9);
new=new1';
new(1)=str2double(get(handles.beta,'String'));
new(2)=str2double(get(handles.tau,'String'));
new(3)=str2double(get(handles.qzero,'String'));
new(4)=str2double(get(handles.qload,'String'));
new(5)=str2double(get(handles.ctemp,'String'));
new(6)=str2double(get(handles.feldem,'String'));
new(7)=str2double(get(handles.resonancefrequency,'String'));
new(8)=str2double(get(handles.eacc,'String'));
new(9)=str2double(get(handles.AW1,'String'));
old=getappdata(0,'cavity');
cavity=[old,new];
setappdata(0,'cavity',cavity);
function xaxis_Callback(hObject, eventdata, handles)
function xaxis_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function yaxis_Callback(hObject, eventdata, handles)
function yaxis_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
```

```
   set(hObject,'BackgroundColor','white');
end
function plotdata_Callback(hObject, eventdata, handles)
cavity=getappdata(0,'cavity');
a=get(handles.xaxis,'Value');
b=get(handles.yaxis,'Value');
setappdata(0,'log',0);
switch a
case 1
   x=cavity(8,:);
   namex='Eacc [MV/m]';
case 2
   x=cavity(9,:);
   namex='forward Power [W]';
case 3
   x=cavity(1,:);
   namex='Coupling';
case 4
   x=cavity(2,:);
   namex='decay Time [sec]';
case 5
   x=cavity(4,:);
   namex='loaded Q';

case 6
   x=cavity(3,:);
   namex='unloaded Q';

case 7
   x=cavity(7,:);
   namex='resonance Frequency [Hz]';
case 8
   x=cavity(5,:);
   namex='Cavity Temperatur [K]';
end
switch b
case 1
   y=cavity(3,:);
```

```
   namey='unloaded Q';
   log=2;
   setappdata(0,'log',2)
case 2
   y=cavity(4,:);
   namey='loaded Q';
   log=2;
   setappdata(0,'log',2)
case 3
   y=cavity(1,:);
   namey='Coupling';
   log=1;
case 4
   y=cavity(2,:);
   namey='decay Time [sec]';
   log=1;
case 5
   y=cavity(6,:);
   namey='Field Emission [mG/h]';
   log=1;
case 6
   y=cavity(5,:);
   namey='Cavity Temperatur [K]';
   log=1;
case 7
   y=cavity(9,:);
   namey='forward Power [W]';
   log=1;
end
switch log
case 2
   figure(2);
   semilogy(x,y,'o','MarkerFaceColor','b','MarkerSize',7)
   a=strcat(namey,' vs. ',namex);
   title(a);
   xlabel(namex);
   ylabel(namey);
   grid
```

```
otherwise
   figure(2);
   plot(x,y,'o','MarkerFaceColor','b','MarkerSize',7)
   a=strcat(namey,' vs. ',namex);
   title(a);
   xlabel(namex);
   ylabel(namey);
   grid
end
function deltatau_Callback(hObject, eventdata, handles)
function deltatau_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function deltaqload_Callback(hObject, eventdata, handles)
function deltaqload_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function deltaqzero_Callback(hObject, eventdata, handles)
function deltaqzero_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function deltaeacc_Callback(hObject, eventdata, handles)
function deltaeacc_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function clockreset_Callback(hObject, eventdata, handles)
ii_set_bits('BIT_EXT_CLK_ENA',uint32(0))
ii_set_bits('BIT_DCM_RST',uint32(1))
ii_set_bits('BIT_DCM_RST',uint32(0))
ii_set_bits('BIT_EXT_CLK_ENA',uint32(1))
function nrcells_Callback(hObject, eventdata, handles)
function nrcells_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
```

```
end
function cwmode_Callback(hObject, eventdata, handles)
uni_gen(81e6,0.1,0.09,10000,'cw_vco',1);
ii_set_word('WORD_DAQ_FREQ',uint32(64)) ;
function phasesweep_Callback(hObject, eventdata, handles)
set(handles.daqtime,'String',num2str(14));
set(handles.ptop,'String',num2str(15));
set(handles.ps1,'String',num2str(5));
adcfreq=(str2double(get(handles.clockfreq,'String')));
uni_gen(adcfreq*1e6,0.1,0.09,50000,'cw_vco',1);
ii_set_word('WORD_DAQ_FREQ',uint32(64));
pause(2) ;
[ phase char1 ] = sweep_function2(-180:60:180,0.5);
uni_loadrot(1,(phase-45))
pause(3)
[ phase char2 ] = sweep_function( (phase-45):2:(phase+45),2) ;
uni_loadrot(1,phase)
ii_set_word('WORD_DAQ_FREQ',uint32(2048));
uni_loadrot(1,phase);
uni_gen(adcfreq*1e6,4,2.5,50000,'sweep',1);
pause(2) ;
counter = 1 ;
range = (phase-10):1:(phase+10) ;
amplitudes = zeros(length(range),1) ;
for I = range,
   uni_loadrot(1,I) ;
   disp(I);
   pause(0.5)
   a = uni_daq([1 9],1) ;
   amplitudes(counter) = min(abs(complex(a(:,1),a(:,2)))) ;
   counter = counter + 1 ;
end
[val, ind] = min(amplitudes) ;
uni_loadrot(1,range(ind))
uni_gen(adcfreq*1e6,15,5,100000,'sweep',1)
set(handles.sliderValue_editText,'String',num2str(range(ind)));
div = double(ii_get_word('WORD_TIMING_FREQ_STR')) ;
daqtime=str2double(get(handles.daqtime,'String'));
```

```
daqoffset=str2double(get(handles.daqoffset,'String'));
datapoints=double(ii_get_word('WORD_DAQ_LIMIT'));
daq_freq=uint32(daqtime*adcfreq*1000000/div/datapoints);
daq_offs=uint32(daqoffset*adcfreq*1000000/div/daq_freq);
ii_set_word('WORD_DAQ_FREQ',daq_freq) ;
ii_set_word('WORD_DAQ_OFFSET',daq_offs) ;
function stebpf_Callback(hObject, eventdata, handles)
f=str2double(get(handles.freq_gen,'String'));
fbpf=num2str(f*(1-1/144)-30);
fid=tcpip_open('ttfvert2.desy.de',5555);
s=strcat('0;26;0;', 'F', fbpf , '\');
tcpip_write(fid,s);
tcpip_close(fid);
function deletemeasure_Callback(hObject, eventdata, handles)
cavity1=zeros(1,9);
cavity=cavity1';
setappdata(0,'cavity',cavity);
function savemeasurefile_Callback(hObject, eventdata, handles)
cavity=getappdata(0,'cavity');
d=datestr(now,'mmddHHMM');
save2file=strcat('cavity',d,'.m');
save(save2file,'cavity','-ascii','-tabs') ;
function offset_I_Callback(hObject, eventdata, handles)
function offset_I_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function offset_Q_Callback(hObject, eventdata, handles)
function offset_Q_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function dacoffset_Callback(hObject, eventdata, handles)
offsetI=str2double(get(handles.offset_I,'String'));
offsetQ=str2double(get(handles.offset_Q,'String'));
ii_set_area('WORD_DAC_OFFSET',uint32(0),uint32(2),uint32(U22Dec([offsetI offsetQ],18)))
;
function quenchtable_Callback(hObject, eventdata, handles)
```

```
quenchtable=getappdata(0,'quenchtable');
eacc=str2num(get(handles.eacc,'String'));
mode=str2num(get(handles.mode,'String'));
% m ist die mode, von links nach rechts steigt die frequenz, ganz rechts is pi
% mode
% n ist die nummer der zelle , von oben nach unten steigt die zellennummer
quenchtablecalc=zeros(9,9);
for m=9:-1:1
   for n=1:9
      quenchtablecalc(n,m)=floor(abs(sin((n-0.5)*m/9*pi)/sin(m/18*pi)));
   end
end
quenchtable(:,mode)=eacc*quenchtablecalc(:,mode);
setappdata(0,'quench',quenchtable);
quenchtableplot
function mode_Callback(hObject, eventdata, handles)
function mode_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
function automatic_Callback(hObject, eventdata, handles)
%% FREQUENZEN BERECHNEN und alle relevanten Werte setzen
%-------------------------------------------------
f_res=str2double(get(handles.modefreq,'String'));
ifreq = double(uint32(f_res/144*10000))*100;
f1 = ifreq*9/1000000 ;
f2 = ifreq*144/1000000 ;
f3 = ifreq*143/1000000 ;
f1=num2str(f1,'%4.2f');
f2=num2str(f2,'%8.4f');
f3=num2str(f3,'%8.4f');
set(handles.freq_gen,'String',f2);
set(handles.clockfreq,'String',f1);
set(handles.convfreq,'String',f3);
set(handles.daqtime,'String',15);
set(handles.ptop,'String',15);
set(handles.ps1,'String',5);
set(handles.daqtime,'String',num2str(14));
```

```
set(handles.ptop,'String',num2str(15));
leftspan=str2double(get(handles.freq_gen,'String'))-0.5 ;
rightspan=str2double(get(handles.freq_gen,'String'))+0.5 ;
set(handles.leftspan,'String',num2str(leftspan));
set(handles.rightspan,'String',num2str(rightspan));
fid=tcpip_open('ttfvert2.desy.de',5555);
leftspan=get(handles.leftspan,'String');
rightspan=get(handles.rightspan,'String');
leftspanz=[':FREQ:STAR' ' ' leftspan];
rightspanz=[';:FREQ:STOP' ' ' rightspan];
s=strcat('0;10;0;',leftspanz , ' MHz',rightspanz , ' MHz \');
tcpip_write(fid,s);
tcpip_close(fid);
h=waitbar(0,'Please wait during testing...','CreateCancelBtn','setappdata(gcbf,''cancelling",1)');
maxlevel=5;
minlevel=2;
progress=1;
stepsize=0.5;
wait_step=(2*maxlevel+1)/stepsize;
cavity1=zeros(1,10);
cavity=cavity1';
setappdata(0,'cavity',cavity);
adcfreq=(str2double(get(handles.clockfreq,'String')));
% Messschleife beginnt hier
%-------------------------------------------------
for steps=minlevel:stepsize:maxlevel
   waitbar(progress/wait_step,h,sprintf('Phase wird gesucht...'));
   if getappdata(h,'cancelling')
      break
      delete(h)
   end
   % PHASE FINDEN FUER DAS LEVEL
   %-------------------------------------------------
   uni_gen(adcfreq*1e6,0.1,0.09,steps*10000,'cw_vco',1);
   ii_set_word('WORD_DAQ_FREQ',uint32(64));
   pause(2) ;
   [ phase char1 ] = sweep_function2(-180:60:180,0.5);
   uni_loadrot(1,(phase-45))
```

```
pause(3)
[ phase char2 ] = sweep_function( (phase-45):2:(phase+45),2) ;
uni_loadrot(1,phase)
ii_set_word('WORD_DAQ_FREQ',uint32(2048));
uni_loadrot(1,phase);
uni_gen(adcfreq*1e6,4,2.5,steps*10000,'sweep',1);
pause(2) ;
counter = 1 ;
range = (phase-10):1:(phase+10) ;
amplitudes = zeros(length(range),1) ;
for I = range,
   uni_loadrot(1,I) ;
   disp(I);
   pause(0.5)
   a = uni_daq_once([1 9],1,4) ;
   amplitudes(counter) = min(abs(complex(a(:,1),a(:,2)))) ;
   counter = counter + 1 ;
end
[val, ind] = min(amplitudes) ;
uni_loadrot(1,range(ind))
uni_gen(adcfreq*1e6,15,5,steps*10000,'sweep',1)
set(handles.sliderValue_editText,'String',num2str(range(ind)));
% LAENGE DER PULSE
%-----------------------------------------------------------
p_length=5;
ptop=3*p_length;
div = double(ii_get_word('WORD_TIMING_FREQ_STR')) ;
time_base = 1/(div/(adcfreq*1e6))/512 ;
clk_counts = floor(ptop*adcfreq*1e6/9)*9+8 ;
prog = [1 p_length*time_base 2^18-1] ;
ii_set_word('WORD_TIMING_FREQ_TRG',uint32(clk_counts));
ii_set_area('AREA_PROG',uint32(0),uint32(length(prog)),uint32(prog)) ;
%DAQ-Timing
```

8.2 QLnoBeam.m

```
 function QLnoBeam(cav,evt)
%% Daten Laden
load acc5nobeam.mat;
modul=acc5nobeam;
%% Daten vorbereiten / Event auswaehlen
cavity=cav;
event=evt;
switch cavity
case 1
   pfor=2;
   probe=3;
   pref=4;
case 2
   pfor=5;
   probe=6;
   pref=7;
case 3
   pfor=8;
   probe=9;
   pref=10;
case 4
   pfor=11;
   probe=12;
   pref=13;
case 5
   pfor=14;
   probe=15;
   pref=16;
case 6
   pfor=17;
   probe=18;
   pref=19;
case 7
   pfor=20;
   probe=21;
   pref=22;
```

```
case 8
   pfor=23;
   probe=24;
   pref=25;
end
a(:,1)=1e6*modul{pfor}(event,:,2);
a(:,2)=1e6*modul{pfor}(event,:,3);
b(:,1)=1e6*modul{probe}(event,:,2);
b(:,2)=1e6*modul{probe}(event,:,3);
c(:,1)=1e6*modul{pref}(event,:,2);
c(:,2)=1e6*modul{pref}(event,:,3);
%klystron_window1=modul{26}(event,:,5);
%klystron_window2=modul{26}(event,:,4);
pf=abs(complex(a(:,1),a(:,2)));
pfphase=(phase(complex(a(:,1),a(:,2)))*(180/pi));
pt=abs(complex(b(:,1),b(:,2)));
ptphase=(phase(complex(b(:,1),b(:,2)))*(180/pi));
pr=abs(complex(c(:,1),c(:,2)));
pt=slidingavg(pt,10);
pr=slidingavg(pr,10);
pf=slidingavg(pf,10);
z_pt=b(:,1)+b(:,2)*j;
z_pf=a(:,1)+a(:,1)*j;
pt=pt';
pr=pr';
pf=pf';
omega_0=1.3e9;
f_sample=1e6;
t=1/f_sample*1:1024;
cavitystr=num2str(cavity);
eventstr=num2str(event);
%% Ableitung bestimmen und averaging Window anwenden
dt=1e-6;
dv(1:1023)=(pt(2:1024)-pt(1:1023));
dphiprobe=(ptphase(2:1024)-ptphase(1:1023));
dvdt=dv/dt;
dphiprobedt=dphiprobe;
meandvdt=slidingavg(dvdt,16);
```

```
%% Detuning berechnen
for i=1:1023
   detuning_pf(i)=(1/(2*pi))*(dphiprobedt(i)-(1.3e9/3e6)*sin(phase(z_pf(i))
-phase(z_pt(i)))*(abs(z_pf(i))/abs(z_pt(i))));
end
%% DGL nach loaded Q aufloesen/berechnen/plotten
for i=1:1023
   z_loaded_Q_pf(i)=-0.5*(omega_0*(2*z_pf(i)-z_pt(i)))/(z_pt(i)*detuning_pf(i)*j-meandvdt(i));
end
loaded_Q_pf=abs(z_loaded_Q_pf);
subplot(3,2,1)
plot(t,pf)
Title= strcat('forward Power of cavity: ',cavitystr,' / ','Event: ',eventstr);
title(Title)
xlabel('time [ms]');
ylabel('Gradient [V/m]')
subplot(3,2,2)
plot(t,pt)
Title=strcat('transmitted Power of cavity: ',cavitystr,' / ','Event: ',eventstr);
title(Title)
xlabel('time [ms]');
ylabel('Gradient [V/m]')
subplot(3,2,3)
plot(t(1:1023),meandvdt)
title('dV_c/dt - Sliding Average')
xlabel('time [ms]');
subplot(3,2,4)
plot(t(1:1023),slidingavg(detuning_pf,10))
title('detuning ')
xlabel('time [ms]');
ylabel('detuning [Hz]')
subplot(3,2,5)
mittelwert=mean(loaded_Q_pf);
error=std(loaded_Q_pf);
error_ql=error/mittelwert;
line(1:1023)=mittelwert;
hold on
plot(t(1:1023),slidingavg(loaded_Q_pf,10))
```

```
Title=strcat('Q_L, ','mean Value: ',num2str(mittelwert,'%0.3g'),', rms:
',num2str(error_ql),'%');
title(Title)
ylabel('loaded Q');
xlabel('time [ms]');
plot(line,'r','LineWidth',2)
hold off
```

8.3 qps.m

```
 function data=qps();% start time
name(1,1) = cellstr('-TStart 2009-09-17T15:30:00');
end time
name(2) = cellstr('-TStop 2009-09-17T15:31:00');
% DAQ details - just use it as it is
name(3) = cellstr('-Exp linac');
name(4) = cellstr('-DDir /daq_data/ilc/LINAC');
name(5) = cellstr('-CDir /daq/ttf2/admtemp');
chan={...
'TOROID/1TCOL'...
'LLRF/C1.ACC5.PFOR',...
'LLRF/C1.ACC5.PROBE',...
'LLRF/C1.ACC5.PREFL',...
'LLRF/C2.ACC5.PFOR',...
'LLRF/C2.ACC5.PROBE',...
'LLRF/C2.ACC5.PREFL',...
'LLRF/C3.ACC5.PFOR',...
'LLRF/C3.ACC5.PROBE',...
'LLRF/C3.ACC5.PREFL',...
'LLRF/C4.ACC5.PFOR',...
'LLRF/C4.ACC5.PROBE',...
'LLRF/C4.ACC5.PREFL',...
'LLRF/C5.ACC5.PFOR',...
'LLRF/C5.ACC5.PROBE',...
'LLRF/C5.ACC5.PREFL',...
'LLRF/C6.ACC5.PFOR',...
```

```
'LLRF/C6.ACC5.PROBE',...
'LLRF/C6.ACC5.PREFL',...
'LLRF/C7.ACC5.PFOR',...
'LLRF/C7.ACC5.PROBE',...
'LLRF/C7.ACC5.PREFL',...
'LLRF/C8.ACC5.PFOR',...
'LLRF/C8.ACC5.PROBE',...
'LLRF/C8.ACC5.PREFL',...
'LLRF.ML/ACC5_AMPL',...
'LLRF.ML/ACC5_PHASE',...
'LLRF/ACC5.TOTAL'};
% DAQ channel name
for i=1:length(chan)
name(5+i)=cellstr(sprintf('-Chan %s',chan{i}));
end
[raw,err]=daq_read_svr(name);
if err
err
return
end
% convert struct to cell array
raw=struct2cell(raw);
% assume we get at least 1 event
labels=fieldnames(raw{1});
data=cell(length(labels),1);
for n=1:length(raw)
raw{n}=struct2cell(raw{n});
end
numevt=length(raw);
if numevt==0
error('DAQ read returned no events');
end
%allocate space for all data in advance
for cc=1:length(raw{1})
   d=raw{1}{cc};
   disp(sprintf('A %d',cc))
   if  isfield(d,'dtype')
      % This is the time stamp
```

```
      data{cc}=zeros(numevt,2);
   elseif isfield(d,'ndata') && isfield(d,'nchan')
      dmax=0;
      for nn=1:length(raw)
         dmax=max(dmax, raw{nn}{cc}.ndata);
      end
      disp(sprintf('X %d %d %d',numevt,dmax,d.nchan));
      %data{cc}=zeros(numevt,dmax,d.nchan+1);
      % initialize to NaN
      data{cc}=repmat(NaN,[numevt,dmax,d.nchan+1]);
   else
      data{cc}=cell(numevt,1);
   end
end
disp('Formatting')
for ee=1:length(raw)
   if mod(ee,100)==0
      disp(sprintf('> %f',100*ee/length(raw)));
   end

   for cc=1:length(labels)
      % disp(sprintf('C %d %s',cc));

      d=raw{ee}{cc};

      if  isfield(d,'dtype')
         % time stamp
         data{cc}(ee,1)=d(1,1)+d(2,1)*1e-6;
         data{cc}(ee,2)=d(3,1);
         continue
      end

      dt=d.dtype;
      if isfield(d,'ndata') && isfield(d,'nchan')
         data{cc}(ee,1:d.ndata,2:(d.nchan+1))=...
         reshape(d.data,[1 d.ndata d.nchan]);
         % construct timebase
         t=0:(d.ndata-1);
```

```
        t=d.start + d.inc*t;
        data{cc}(ee,1:d.ndata,1)=reshape(t,[1 d.ndata 1]);
      else
        data{cc}{ee}=d;
      end
    end
end
```

8.4 ubertragungsfunktion.m

```
% for uebertragungsfunktiontest
  clear all

  % general resonanzfrequenz -> f,w
  f0 = 1.3e9;
  w0 = 2*pi*f0;

  QL9 = 3e6;

  % resonanzfrequenz bei jedem modus -> f
  f(9) = 1300444000;
  f(8) = f(9)-800000;
  f(7) = f(9)-3053000;
  f(6) = f(9)-6501000;
  f(5) = f(9)-10694000;
  f(4) = f(9)-15122000;
  f(3) = f(9)-19237000;
  f(2) = f(9)-22594000;
  f(1) = f(9)-24773000;

  vec=1:9;

  % kopplung zwischen jeden zwei cavitycells
  kopplung = 2 * sin(pi.*vec/18).^2;
  kopplung(9)=1;
```

```
% halbe bandbreite –> w12 von jedem FM
w12 = kopplung * pi.* f / QL9;

% idealen Verstimuung –> dw von jedem FM
dw = 2 * pi.* (f - f(9));

% uebertragungsfunktion von jedem cavitycell
h_cav_mode = [tf([w12(1) w12(1)^2], [1 2*w12(1) w12(1)^2+dw(1)^2]), ...
-1 * tf([w12(2) w12(2)^2], [1 2*w12(2) w12(2)^2+dw(2)^2]), ...
tf([w12(3) w12(3)^2], [1 2*w12(3) w12(3)^2+dw(3)^2]), ...
-1 * tf([w12(4) w12(4)^2], [1 2*w12(4) w12(4)^2+dw(4)^2]), ...
tf([w12(5) w12(5)^2], [1 2*w12(5) w12(5)^2+dw(5)^2]), ...
-1 * tf([w12(6) w12(6)^2], [1 2*w12(6) w12(6)^2+dw(6)^2]), ...
tf([w12(7) w12(7)^2], [1 2*w12(7) w12(7)^2+dw(7)^2]), ...
-1 * tf([w12(8) w12(8)^2], [1 2*w12(8) w12(8)^2+dw(8)^2]), ...
tf(1, [1/w12(9) 1])];

% diskrete uebertragungsfunktion von jedem cavitycel l
h_cav_mode_dis1 = c2d(h_cav_mode(1), 0.1, 'tustin');
h_cav_mode_dis2 = c2d(h_cav_mode(2), 0.1, 'tustin');
h_cav_mode_dis3 = c2d(h_cav_mode(3), 0.1, 'tustin');
h_cav_mode_dis4 = c2d(h_cav_mode(4), 0.1, 'tustin');
h_cav_mode_dis5 = c2d(h_cav_mode(5), 0.1, 'tustin');
h_cav_mode_dis6 = c2d(h_cav_mode(6), 0.1, 'tustin');
h_cav_mode_dis7 = c2d(h_cav_mode(7), 0.1, 'tustin');
h_cav_mode_dis8 = c2d(h_cav_mode(8), 0.1, 'tustin');
h_cav_mode_dis9 = c2d(h_cav_mode(9), 0.1, 'tustin');
% uebertragungsfunktion von dem ganzen cavity
% hcav = sum(hcavcell);
h_cav = h_cav_mode(1) + h_cav_mode(2) + h_cav_mode(3) + h_cav_mode(4) ...
+ h_cav_mode(5) + h_cav_mode(6) + h_cav_mode(7) + h_cav_mode(8) ...
+ h_cav_mode(9);

% diskrete uebertragungsfunktion von dem ganzen cavity
h_cav_dis = c2d(h_cav, 0.1, 'tustin');

% uebertragungsfunktion von klystron
h_kly = tf(1, [1/2/pi/3e6 1]);
```

```
% uebertragungsfunktion von feedback delay
delay = 100e-6; %%%%%%%%————————— delay veraenderen —————————%%%%%%%%
s = tf('s');
FB_Delay = exp(-1 * s * delay);

% feedback gain
FB_Gain = 500; %%%%%%%%————————— gain veraenderen —————————%%%%%%%%

parameter=strcat('Feedback Delay:',num2str(delay),'s, Feedback Gain:',num2str(FB_Gain));
% transformationsfunktion von dem system bei open loop
h_tf = FB_Gain * FB_Delay * h_cav;

% diskrete transformationsfunktion von dem system
h_tf_dis = c2d(h_tf, 0.1, 'tustin');

% plot die transformationsfunktion in Bode
%hold on
[mag,pha,w]= bode(h_tf, {100, 10e7*2*pi});
subplot(2,1,1)
loglog(w./(2*pi),reshape(mag,1,length(mag)),'b');
xlabel('Abweichungsfrequenz(Hz) zu pi-modus');
ylabel('Amplitude(dB)');
xlim([10 10^8]);
ylim([10^-7 1000]);
text(2*10^1,10^-6,parameter)
subplot(2,1,2)
semilogx(w./(2*pi),reshape(pha,1,length(mag))-pha(1),'b');
xlabel('Abweichungsfrequenz(Hz) zu pi-modus');
ylabel('phase(deg)');
xlim([10 10^8]);
ylim([-2000 0]);
%hold off
```

8.5 Characteristic curves of test stand hardware

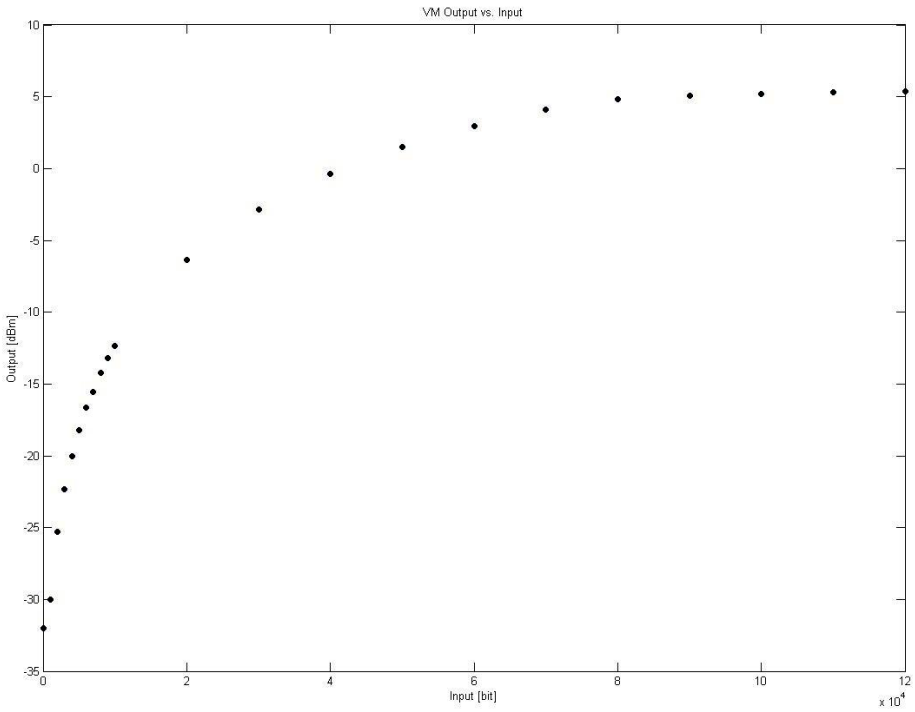


Figure 8.1: output - input relation of the vectormodulator. nonlinear regime / saturation is reached.

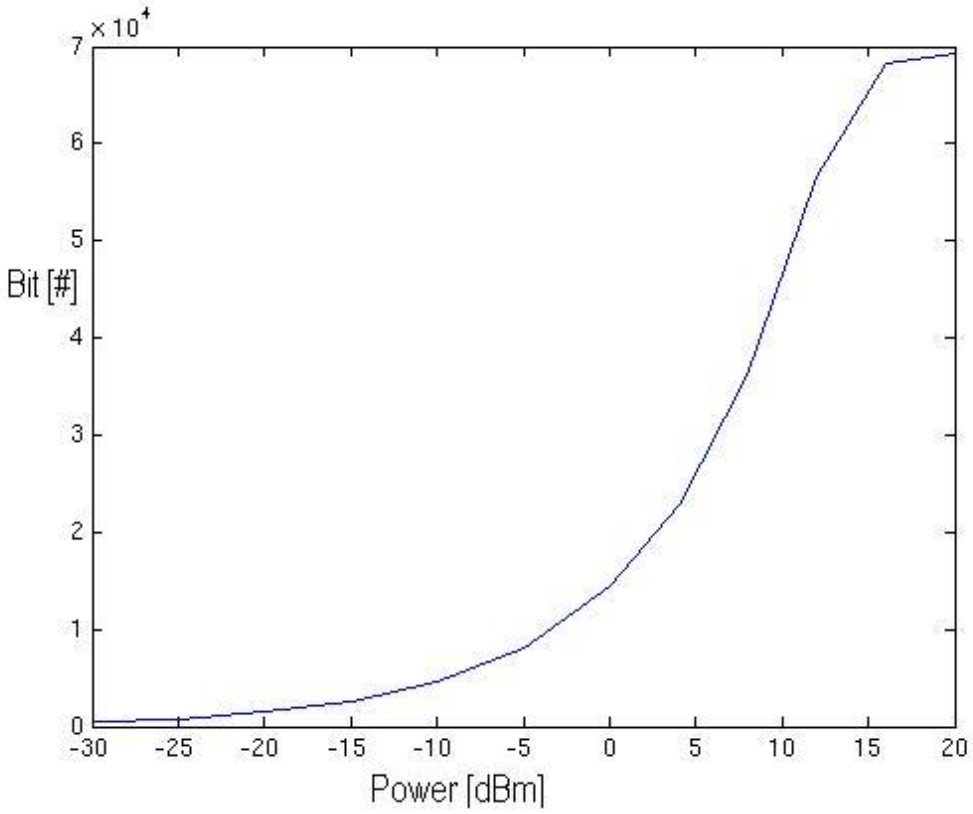


Figure 8.2: ADC-signal vs. powerinput.

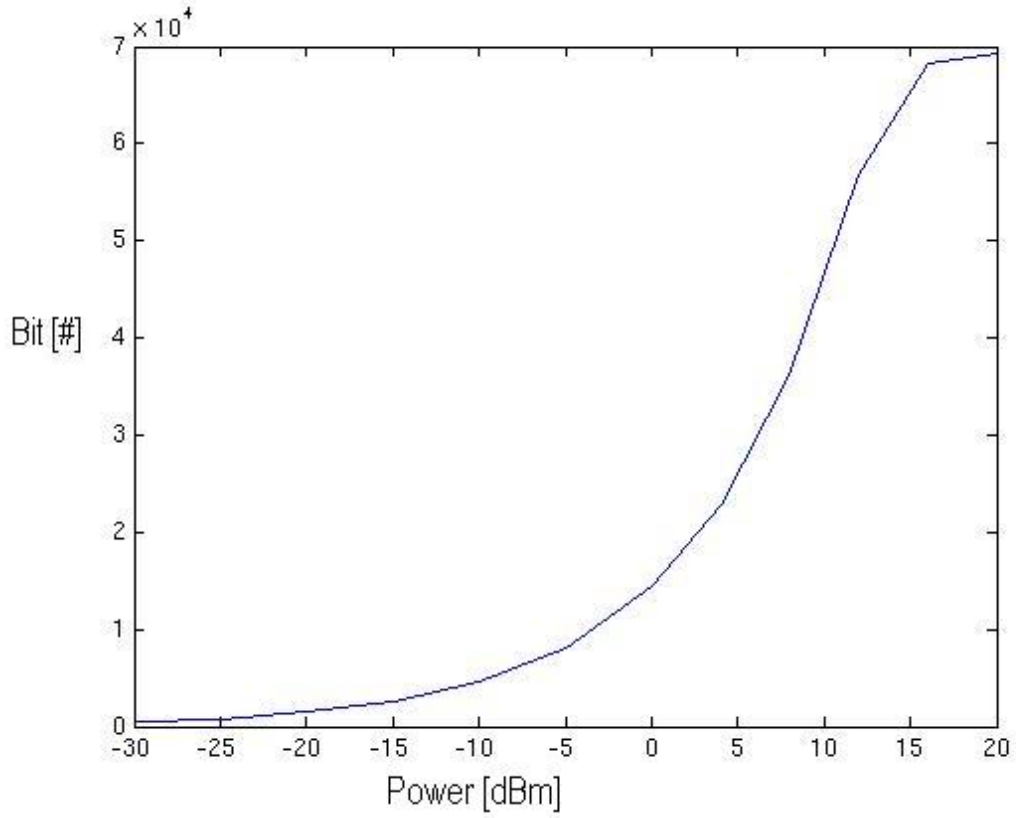


Figure 8.3: ADC-signal vs. powerinput.

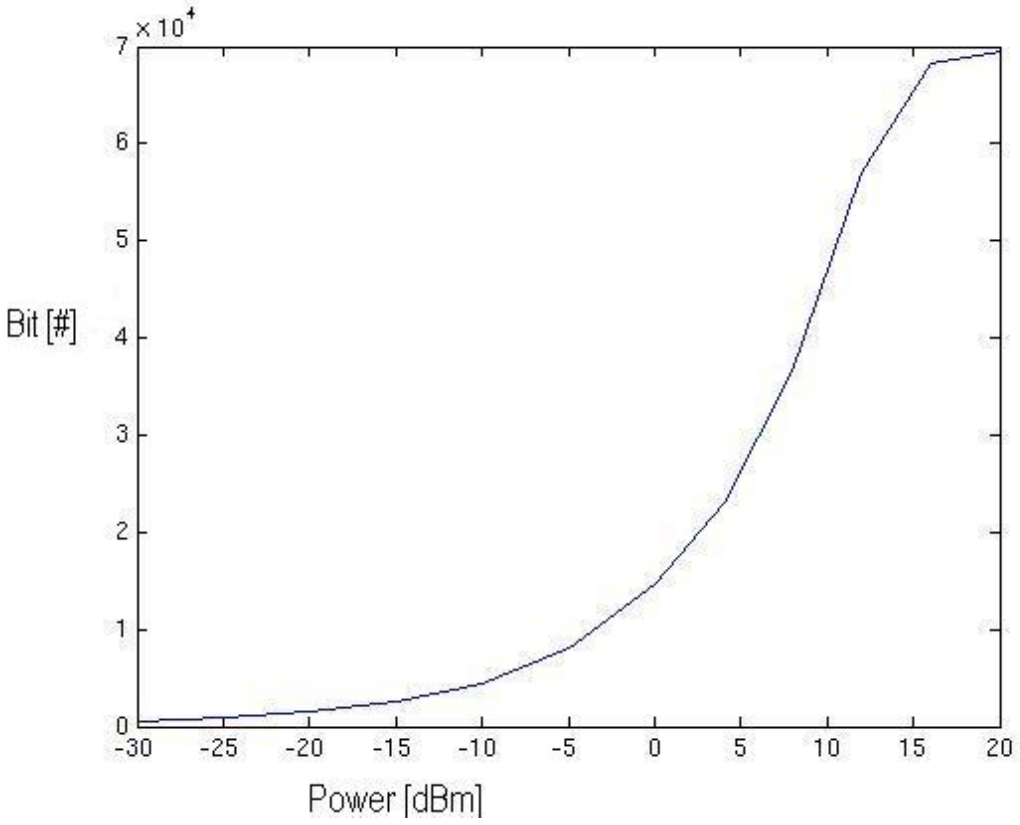


Figure 8.4: ADC-signal vs. powerinput.

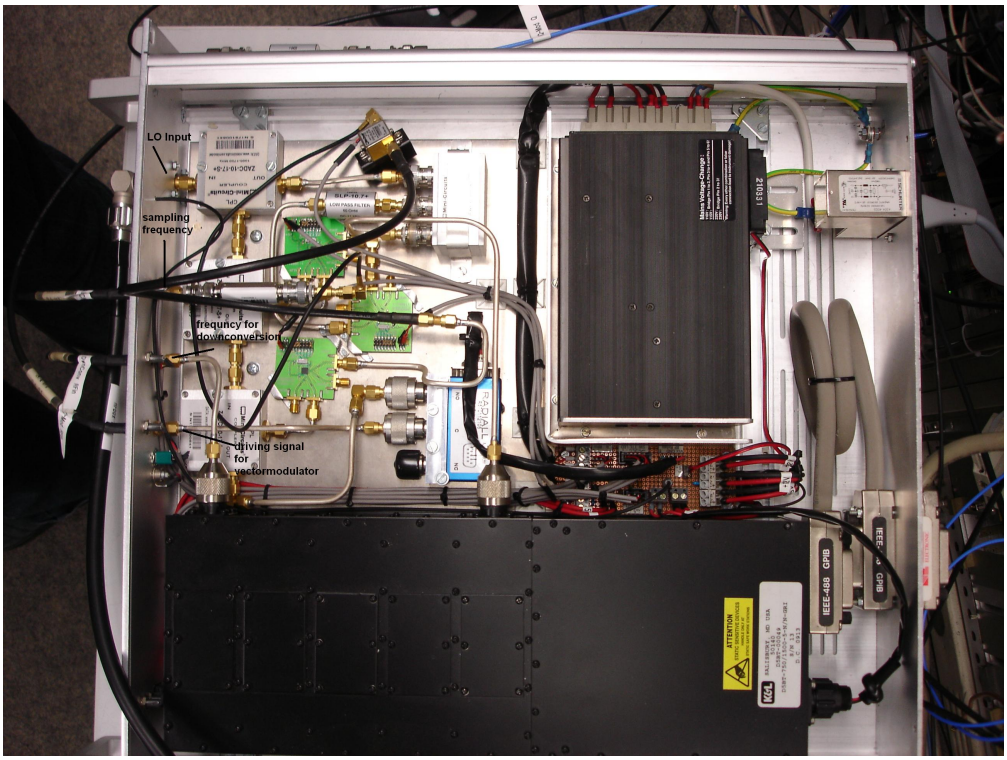


Figure 8.5: Picture of the RF Box, taken from above.

8.6 Fouriertransformation of reflected power to identify the noise

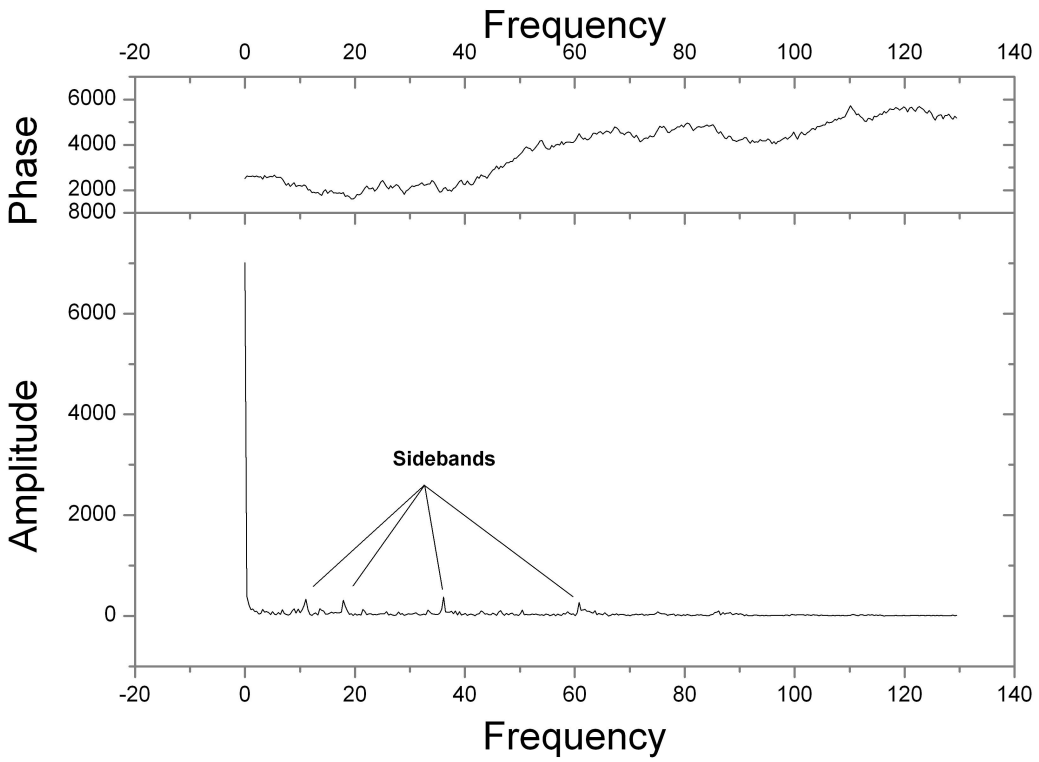


Figure 8.6: FFT of reflected power, sidebands generated by downconverter can be seen at IF+ [9, 18, 36, 63] MHz.

The oscillations are at IF+ $\approx$ 16 and 72 Hz.

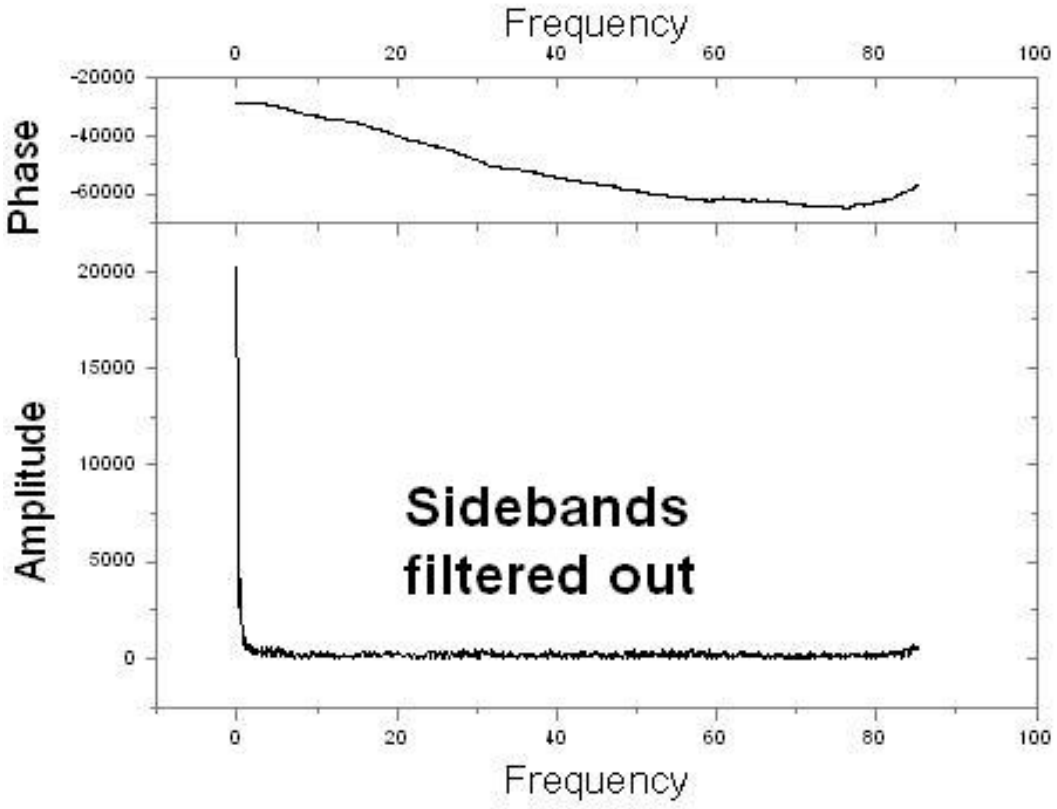


Figure 8.7: FFT of reflected power, sidebands filtered out by low-pass. oscillations still seen in reflected power.

Bibliography

[1] Ashcroft, Neil, Mermin, David: 'Solid State Physics',
Philadelphia: Saunders College, 1976

[2] Arnold, Ned; 'Accessing DAQ Data using MATLAB', DAQ tutorial available at ILC WIKI

[3] Brandt, Alexander ; PhD Thesis: 'Development of a finite State Machine for the automated Operation of the LLRF Control at FLASH', University Hamburg, 2007

[4] Branlard, Julien: MATLAB cavity Simulator, 2007, ILC-doc-481-v1

[5] Branlard, Julien: 'Operational Solution to obtaining a Flat Vector Sum from multiple Cavities with Gradient Disparities', 2007, Beams-doc-3443-v1

[6] Branlard, Julien et. al: 'Coupling Adjustments for Gradient Disparities', 2006, Beams-doc-3442

[7] Carwardine, John: 'Progress Report on FLASH 9 mA Studies', held at FLASH Seminar - DESY on 24. November 2009

[8] Chase, Brian: 'Lessons Learned from the 9 mA Test', Proceedings of the 14th International Conference on RF Superconductivity, Dresden 2009

[9] Chase,Brian; Branlard, Julien; Cancelo, Gustavo : '9 mA beam current study at FLASH - DESY', 2009, Beams-doc-3444-v1

[10] Geng, Zheqiao; Ayvazyan, Valeri; Simrock, Stefan: 'Progress on LLRF applications development', held at FLASH Seminar - DESY on 7. April 2009

[11] Liepe, Matthias; Doktorarbeit: 'Regelung supraleitender Resonatoren mit Strahlbelastung am Tesla-Test-Beschleuniger', University Hamburg 1998

[12] Oppenheim, Alan; Willsky, Alan: 'Signale und Systeme', VCH Verlagsgesellschaft 1989

[13] Padamsee, Hasan; Knobloch, Jens; Hays, Tom: 'RF Superconductivity for Accelerators', John Wiley & Sons, Inc. 1998

[14] Schilcher, Thomas; PhD Thesis: 'Vector Sum Control in Lorentz Force Detuned Cavities', University Hamburg 1998

[15] Simrock, Stefan, Czarski, Thomasz; Koprek, Waldemar: 'CHECHIA cavity driving with FPGA controller' , TESLA-Report 2005

[16] Simrock, Stefan et al.: 'Experience with the Control of the Vector sum at the Tesla Test Facility', Proceedings of the 24th Linear Accelerator Conference, Victoria 2008

[17] Vogel, Elmar: 'Considerations on the third Harmonic rf of the European XFEL', Proceedings of the 13th International Conference on RF Superconductivity, Beijing 2007

[18] Weise, Hans: 'Neuentwicklungen in der Beschleunigertechnologie', Proceedings DPG Tagung, Freiburg 2008

[19] Wille, Klaus: 'Physik der Teilchenbeschleuniger und Synchrotronstrahlungsquellen', B.G. Teubner Verlag 1992

[20] Yun, Xiangyun; private communication

[21] CERN Accelerator School Proceedings 2008, 'Digital Signal Processing', CERN

[22] Norem, Jim; Pellin, Mike: 'Gradient Limits and SCRF Performance', Proceedings of the 13th International Conference on RF Superconductivity, Beijing 2007

[23] Shemelin, Valery ; Padamsee, Hasan: 'Magnetic Field Enhancement at Pits and Bumps on the Surface of Superconducting Cavities', TTC-Report2008-07

[24] Buckel, Werner: 'Supraleitung: Grundlagen und Anwendungen', VCH Verlagsgesellschaft 2004

[25] Lilje, Lutz: 'Superconducting Radiofrequency Accelerating Structures', Terascale Accelerator School, Hamburg 2008

[26] Koprek, Waldemar et al.: 'Software layer for FPGA-based TESLA cavity control system', TESLA-Report 2004

[27] Szewinski, Jaroslaw et al.: 'Software for development and communication with FPGA based hardware', TESLA-Report 2005

[28] Jalmuzna, Wojciech: 'Performance of 24 Cavity vector sum controller with distributed architecture', Proceedings of the 6th European Particle Accelerator Conference, Stockholm 1998

[29] Simrock, Stefan; Liepe, Matthias: 'Adaptive Feed Forward for the digital RF Control System at the Tesla Test Facility', Proceedings of the 6th European Particle Accelerator Conference, Stockholm 1998

[30] Simrock, Stefan et al.: 'Design of the digital LLRF control System for the Tesla Test Facility', Proceedings of the 5th European Particle Accelerator Conference Barcelona 1996

[31] Tomasz Czarski 2007 *Meas. Sci. Technol.* **18** 2328

[32] Cancelo, Gustavo: 'LLRF 9 mA operations and data analysis', Held at 'Workshop on Long Bunch Trains', DESY 2009

[33] Michizono, Shin: 'Results of High Gradient Study', Held at 'Workshop on Long Bunch Trains', DESY 2009

[34] Branlard, Julien: 'High Gradient with 9 mA beam', Held at 'Workshop on Long Bunch Trains', DESY 2009

[35] Hovater, Curt 'RF Controls and Tuning System', Proceedings of the 14th International Conference on RF Superconductivity, Dresden 2009

[36] Schilcher, 'RF applications in digital Signal Processing', Proceedings Cern Accelerator School 2007

[37] Vogel, Elmar: 'High gain proportional rf control stability at TESLA cavities', Phys. Rev. ST Accel. Beams **10**, 052001 (2007)

[38] Keung, Justin: 'Application Note: The Effects of Latency on the Feedback Stability of the 8/9 pi mode', Penn Virtual Cavity Documentations 2007

[39] Huehning, Markus; Diplomarbeit: 'Selbstoptimierende Parametersteuerung der Hochfrequenz des supraleitenden Linearbeschleunigers TESLA Test Facility', RWTH Aachen, 1998

[40] Pawlik, Pawel; PhD Thesis: 'Accelerating electric field strength amplitude and phase calibration based on particle beam induced electric field strength transient detection in linear particle accelerators', Technical University of Lodz 2007

[41] Lilje, Lutz; Simrock, Stefan; Kostin, Denis: 'Characteristics of a fast piezo-tuning mechanism for superconducting cavities', Proceedings of the 8th European Particle Accelerator Conference, Paris 2002

[42] Nagle, Darragh; Knapp, Edward: 'Coupled Resonator Model for Standing Wave Accelerator Tanks', The Review of Scientific Instruments Volume 38, Number 11, November 1967

[43] Homepage of the TTF-Group at DESY
http://www-mhf.desy.de/desy/ttfadm/

[44] Navitski, Aliaksandr; Reschke, Detlef: Surface roughness and correlated enhanced field emission investigations of electropolished niobium samples, Proceedings of the 14th International Conference on RF Superconductivity, Dresden 2009

[45] Doolittle, Ma, 'Digital Low-Level RF Control using Non-IQ Sampling', Proceedings of the 2006 Linear Accelerator Conference, Knoxville 2006