Digital high bandwidth feedback controller

Masterarbeit in Physik

von

Florian Seidler,

angefertigt im

Institut für Angewandte Physik,

vorgelegt der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn.

Mai 2015

Angefertigt im
Institut für Angewandet Physik
der Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Gutachter: Prof. Dr. Dieter Meschede
2. Gutachter: Prof. Dr. Martin Weitz

Abstract

This work reports on a high bandwidth (>0.5 MHz) digital controller for intensity stabilisation. Important properties of the digital hardware are characterised. The noise characteristics of the digital to analogue converter limit the digital controller, making it inferior compared to a state-of-the-art analogue controller in this respect. However, similar bandwidths are achieved with both controllers in PI mode. In addition a proof of concept digital Internal Model Controller is presented showing an increased bandwidth of 2.4 MHz. The results and methods presented in this work may be used to improve intensity control for quantum optics experiments using digital controllers.

Contents

1. Introduction

Most experiments in physics rely on at least one parameter that can be controlled by the experimentalist. In some cases passive stability of the controlled system is sufficient to conduct the experiment. In many others however it is not. Then specialised devices are needed to actively stabilise or steer the parameter relying on measurements of the parameter to counteract any unpredicted behaviour of the system. Such devices are called feedback controllers [1] and are the main focus of this work.

Specifically versatile control devices are needed for the control of optically trapped neutral atoms which provide a flexible framework for the application and investigation of quantum physics. Manipulation of quantum states of localized neutral atoms was used for fundamental studies of quantum mechanics [2], [3], high precision metrology [4] and the implementation of quantum information [5], [6] and quantum simulation protocols [7].

Despite the advances of digital electronics, feedback controllers are still commonly build purely on analogue electronics. One goal of this thesis is to provide a comparison between a state of the art analogue controller and a digital general purpose device to examine viability of digital controllers for quantum optics experiments by the means of a test system stabilising intensity using an acusto-optical modulator. The other is to explore the possible benefits of digital controllers in practice.

Many advanced control schemes that provide increased performance including the one used in this work can be realized both in analogue or digital hardware. However programmable digital controllers offer easier and faster design of complex controllers. Also instead of manufacturing hardware just for one system that might change at a later point, the same hardware may be reused while changes and new developments are done in software [8].

Furthermore some advanced (adapting) schemes employing very complex logic (e.g. artificial intelligence) [1], [9] may only be practically realisable using the scalability of digital hardware and the ability to store and transmit data nearly indefinitely without degradation [10]. These properties of digital hardware could also be used to reduce the amount of time needed for maintenance of an experiment. The control hardware may provide automated recovery from failures or remote access to analysis that would otherwise require specialised hardware to be connected on site.

Outline

The remaining part of this thesis is structured as follows: In chapter 2 we analyse the control schemes applied later in time and frequency domain. Chapter 3 is focused on the

use of digital devices to build and evaluate control loops. In chapter 4 the hardware used to implement and test the digital controller is described. Finally, in chapter 5 we compare the digital controller with a state-of-the-art analogue controller and shows some benefits of an advanced control algorithm.

2. Continuous time control theory

2.1. The PID controller

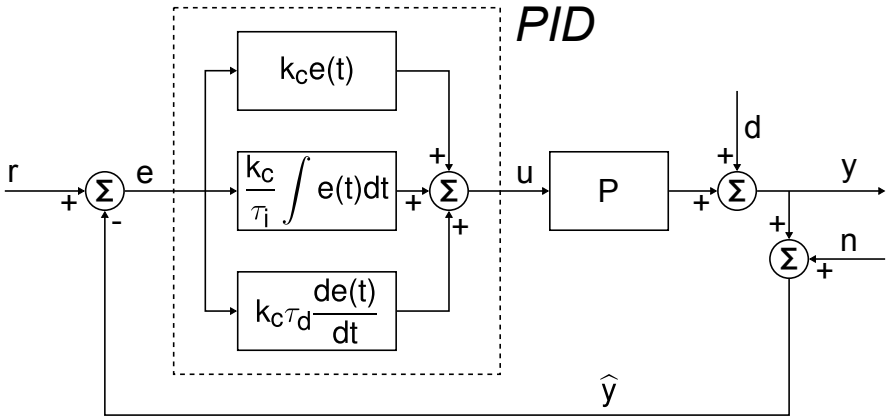


Figure 2.1.: *Block diagram of a PID feedback loop.* The user puts setpoint r. Output y of Process P is measured with a sensor adding noise n. The sum $\widehat{y}$ is then subtracted from r to get error e. This is fed to the PID controller. In response it takes control action u to obtain $y = r$ even if there is an unpredictable disturbance d in y.

The PID (for Proportional Integral Differential) controller is the most prevalent controller type both in industry and in science. It owes this ubiquity to its simplicity and robustness [11]. The working principle is shown in fig. 2.1. The output value of the system to be controlled (process) is subtracted from a desired level of this value given by the user (setpoint). The result is then used to generate an input to the process (control action) influencing its output value (controlled variable). This constitutes a feedback loop.

Assume for example that at $t = 0$ either the setpoint r is stepped up or there is a step down in the controlled variable y because of a disturbance d. Disturbances are unpredictable changes in the unregulated process output. Both possibilities will result in a step up in the error signal i.e.:

$$e(t) = k\Theta(t) \tag{2.1}$$

with the step function $\Theta(t)$ and the step amplitude k.

The most intuitive approach is a controller only using proportional action (the P-part) whose output is proportional to its input. However the steady state output of the process will not

match the setpoint exactly for finite controller gain k_c but stay at $y = e \cdot k_c = r \cdot {}^{k_c}/_{k_c-1}$.

To counteract this we apply integral action (I-part). It will keep rising, increasing y and decreasing e, until (in steady state limit) $e = 0$. The drawback is that it cannot by its principle be as immediate as the proportional part. The speed at which it reacts is governed by the integral time constant τ_i. Therefore in most applications both P- and I-part are used together, constituting a PI-controller.

The Differential action (D-part) is less frequently used in real world applications for reasons that will become apparent later (see 2.5). It complements the integral action, since its contribution will be largest for steep slopes i.e. fast changes in e. Ideally it nearly compensates for the offset remaining from the proportional action and reduces as time goes on while integral action increases due to a small remaining error. The obvious downside is an overreaction (overshoot) to changes that are faster than the response time implied by the derivative time constant τ_d. The opposite would be less problematic since with an appropriately set time constant integral action will be sufficient to keep the error small on longer time-scales. Sources of the problematic steep rises can be step-like setpoint changes, random (also step-like) changes in process behaviour (disturbances) or sensor noise (high frequency).

The overall continuous time equation describing the ideal PID is given as:

$$C(t) = k_c \left(e(t) + \frac{1}{\tau_i} \int e(t)dt + \tau_d \frac{d}{dt} e(t) \right) \tag{2.2}$$

To better understand how we can avoid noise amplification by differential action and how to design more advanced control algorithms we will switch to another perspective. This perspective is the frequency domain.

2.2. System analysis by Laplace transformation

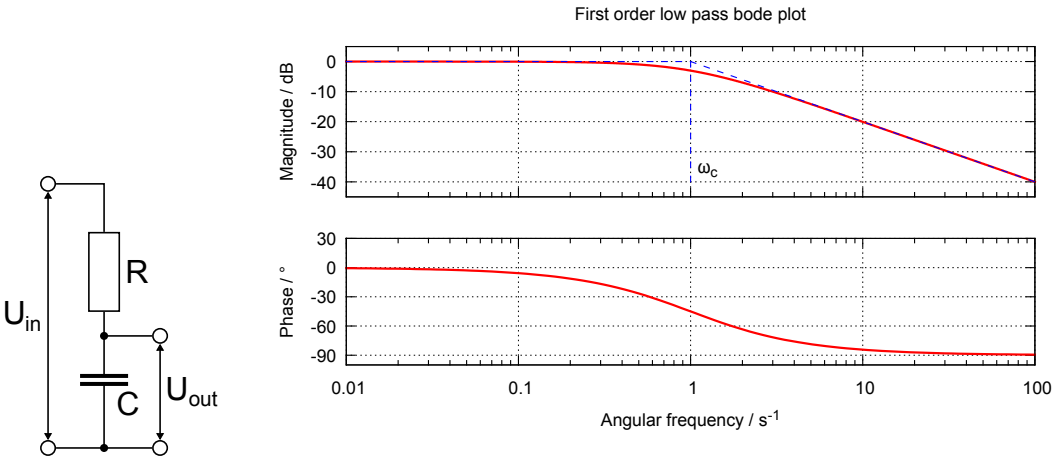
Instead of the widely used Fourier transform, in control theory we typically use the Laplace transform. The bilateral Laplace transform of a function $f(t)$ is defined as [12]:

$$F(s) = \mathcal{L}\{f(t)\} = \int_{-\infty}^{\infty} f(t) e^{-st} dt \tag{2.3}$$

for real t provided $f(t)$ is integrable. Assuming t is time, $s \in \mathbb{C}$ is a complex valued angular frequency. It can be written as $s = \sigma + i\omega$. $\omega \in \mathbb{R}$ represents angular frequency and $\sigma \in \mathbb{R}$ can be interpreted as coefficient of an exponentially damped sinusoid. This means functions for which Fourier transform does not converge like $f(t) = t$ or $f(t) = e^t$ can be transformed by the bilateral Laplace transform. We refer to the bilateral Laplace transform as the Laplace transform from here on.

Evaluating $F(s) \in \mathbb{C}$ for purely imaginary $s = i\omega$ will yield the Fourier transform:

$$F(s)|_{s=i\omega} = \int_{-\infty}^{\infty} f(t) e^{i\omega t} dt = F(i\omega) \tag{2.4}$$



(a) RC low pass circuit.

(b) RC low pass bode plot.

Figure 2.2.: *The RC low pass and its transfer function.* The critical angular frequency is $\omega_c = 1/\tau$ at which the magnitude of the transfer function is $|H(i\omega)| = 1/\sqrt{2} \approx -3\,dB$. Beyond it the magnitude decreases by 20 dB per decade. The high frequency phase of $-90°$ means the output lags behind the input by 1/4 period (for a sinusoid).

Its absolute value $|F(s)|$ represents an amplitude and its argument $\arg(F(s))$ a phase.

A very useful property of the Laplace transform is the fact that a differentiation in time will be transformed to a multiplication with s:

$$\mathcal{L}\{f'(t)\} = sF(s) \tag{2.5}$$

More generally we can find that the nth derivative in time can be transformed to:

$$\mathcal{L}\left\{f^{(n)}(t)\right\} = s^n F(s) \tag{2.6}$$

Therefore differential equations in time will be transformed to algebraic equations in s. As an example of the usefulness of this property let us evaluate a simple low pass filter consisting of a resistor and a capacitor (see fig. 2.2, more details in [13]).

We want to find the transfer function of the RC-filter as a function of s. This is defined as:

$$H(s) = \frac{y(s)}{x(s)} \tag{2.7}$$

$x(s)$ and $y(s)$ are the Laplace transformed input and output respectively. We first analyse the individual components. An ideal capacitor will fulfil:

$$I(t) = C\frac{dU(t)}{dt} \xrightarrow{\mathcal{L}} I(s) = CU(s)s \Rightarrow Z(s) = \frac{1}{Cs} \tag{2.8}$$

with the current I, voltage U, capacitance C and impedance Z. For the resistor $Z(s) = R$. Now we can combine this assuming $x(s)$ and $y(s)$ are voltage signals. By Kirchhoff's current law the current $I(s)$ through both components must be the same, so:

$$H(s) = \frac{y(s)}{x(s)} = \frac{I(s)Z_y(s)}{I(s)Z_x(s)} = \frac{\frac{1}{Cs}}{\frac{1}{Cs} + R} = \frac{1}{1 + RCs} \tag{2.9}$$

This is the representation of a low pass filter since the magnitude of this function is approximately 1 for $s \ll 1/RC$ and decreases as $1/i\omega$ for $s = i\omega \gg 1/RC$. The magnitude is also known as gain and is shown in double logarithmic representation in the upper part of what we call a bode plot [14] (see fig. 2.2). The lower part of the bode plot shows $\arg(H(s))$ or the phase in semi-logarithmic representation.

In general not only electrical circuits but all Linear Time Invariant (LTI) systems can be analysed using this method. Once we have obtained its transfer function we can also analyse the stability of a such an LTI system. It can be shown that if the transfer function has poles in the right half of the complex plane mapping s, the system is unstable [15]. Correspondingly it is stable if there is no pole in the right half-plane including the imaginary axis.

2.3. PID in frequency domain

Using the perspective of s-space we find the transfer function T of a closed feedback loop [16] (suppressing s dependency of all variables):

$$T = \frac{y}{r} = \frac{ePC}{r} = \frac{(r - y)PC}{r} = PC - TPC \Rightarrow T = \frac{PC}{1 + PC} \tag{2.10}$$

with all variables as introduced in sec. 2. PC is the open loop transfer function as obtained from the composition of the controller and process transfer function which is just the product in s-space.

More specifically the ideal PID open loop transfer function, whose bode plot is shown in fig. 2.3, is:

$$C(t) = k_c \left(1 + \frac{1}{\tau_i} \int e(t)dt + \tau_d \frac{d}{dt} e(t)\right) \xrightarrow{\mathcal{L}} C(s) = k_c \left(1 + \frac{e(s)}{\tau_i s} + \tau_d s e(s)\right) \tag{2.11}$$

As mentioned in sec. 2.2 we can use this description to get deeper insight in the behaviour of the control loop.

In steady state or $s \to 0$ the integral part goes to infinity dominating the behaviour of $C(s)$. Assuming a stable process ($P(s)$ has no right hand plane poles) this means for the closed loop system:

$$\lim_{s \to 0} T(s) = \lim_{s \to 0} \frac{PC}{1 + PC} = 1 \tag{2.12}$$

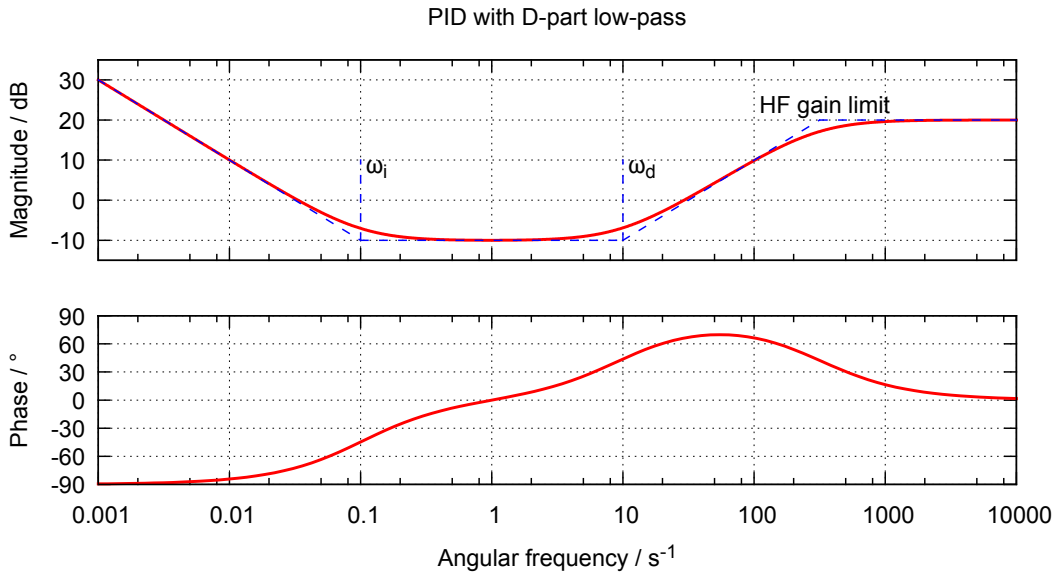


Figure 2.3.: *Bode plot of the open loop transfer function of a PID controller.* The transition frequencies are $\omega_i = 1/\tau_i$ (from I- to P-part) and $\omega_d = 1/\tau_i$ (from P- to D-part). For frequencies higher than the low-pass transition frequency, differentiator and low-pass balance each other out leading to an HF gain limit.

Therefore there is no steady state offset since $y = 1 \cdot r$.

Depending on the time constants for intermediate frequencies the proportional part may contribute the most while for high frequencies the differential part will be dominant. This leads back to the question of high frequency sensor noise amplification by differential action discussed at the end of sec. 2.1. For high frequencies we can assume $T \approx 1$ analogously to our steady state calculation. Also high frequency noise is usually white i.e. it has the same amplitude for all frequencies. The noise from the sensor distorts the feedback that the controller receives and is increasingly amplified by the differential action for increasing frequencies. This is undesirable since it will lead to wear of any control elements (especially mechanical ones) or even exceed their input range. Therefore the maximum amplification of the differential part is usually limited to 5 - 20.

To achieve this, differential action is filtered with a low-pass, yielding:

$$D(s)F(s) = \tau_d s \cdot \frac{1}{\alpha\tau_d s + 1} \tag{2.13}$$

with the D-part transfer function $D(s)$, the low pass filter transfer function $F(s)$ and the filter coefficient α. For high frequencies this will go to the value $\lim_{s\to\infty} D(s)F(s) = 1/\alpha$.

This approach also solves a more practical problem. A pure derivative cannot be physically realized because no physical system can provide gain at infinite frequencies [17]. Since we do not require a pure derivative we do not need to approximate it (e.g. with a backwards difference). Therefore we rewrite eq. 2.13 to:

$$\frac{\tau_d s}{\alpha\tau_d s + 1} = \frac{1}{\alpha}\left(\frac{\alpha\tau_d s}{\alpha\tau_d s + 1}\right) = \frac{1}{\alpha}\left(1 - \frac{1}{\alpha\tau_d s + 1}\right) \tag{2.14}$$

meaning only a low pass and an amplification by $1/\alpha$ is required.

With increasing frequencies time delay becomes increasingly important. Until now we did not include this in our considerations. In s-space a time delay is represented as [18]:

$$\mathcal{L}\{f(t-\theta)\} = e^{-\theta s} f(s) \tag{2.15}$$

with the time delay θ.

The next section will show how to use s-space analysis and modelling to optimize a control solution for a given process.

2.4. Internal model control

The basic idea of Internal Model Control (IMC) is to use the ability to model the process to construct a controller specifically for it. The IMC design method described in the following is closely based on [19].

The easiest way to use the model to our advantage would be to invert transfer function of the process and use it as the controller without any feedback as shown in fig. 2.4:

$$T = PQ = PP^{-1} = 1 \tag{2.16}$$

This is a basic feedforward. The most obvious drawback is the inability to counteract disturbances. Also there are usually reasons why the inverse cannot be realised in a physical system. For example inverting a process that behaves like an integrator will give a pure derivative. Even if the inversion is possible the model may not be exact i.e. the parameters

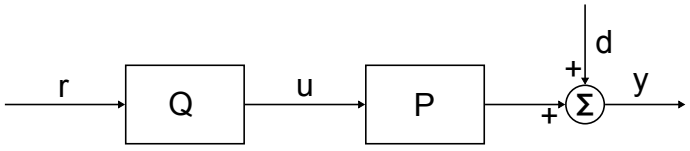


Figure 2.4.: *Block diagram of a feedforward control.* The setpoint r is used by the controller Q to produce a control action u suitable to achieve an output of $y = r$ of the process P. However y is also influenced by disturbances d that are not accounted for.

might not be known precisely enough or there might even be some behaviour that is not modelled at all (typically at higher frequencies) [20].

Let us nevertheless attempt this, assuming the process behaves like a first order low pass:

$$\tilde{P} = \frac{k_p}{\tau s + 1} \rightarrow \tilde{P}^{-1} = \frac{\tau s + 1}{k_p} \tag{2.17}$$

where $\tilde{P}$ denotes the process model, k_p the steady state process gain and τ low pass time constant. The inverse looks like an ideal PID controller without integral action. The parameters can be read off the equation to be $k_c = {}^1/_{k_p}$ and $\tau_d = \tau$.

Besides the fact that the pure derivative action is problematic, most processes also contain some time delay which is commonly referred to as the dead time θ. Including it in the model of the process yields the so called First Order Plus Dead Time (FOPDT) model. Despite its simplicity this model is sufficient for many real-world applications [21]. Yet inverting it reveals a problem:

$$\tilde{P} = \frac{k_p}{\tau s + 1} e^{-\theta s} \rightarrow \tilde{P}^{-1} = \frac{\tau s + 1}{k_p} e^{\theta s} \tag{2.18}$$

The factor $e^{\theta s}$ represents a time *advance*. That means the present output of the system would be based on future inputs which would violate causality.

To obtain a realizable controller the model is factored into a realizable $\tilde{P}_-$ and an unrealisable $\tilde{P}_+$ component. $\tilde{P}_+$ contains all factors whose inversion would violate causality (time advance) or lead to instability (poles in the right half plane). As far as the feedforward controller Q is concerned we simply ignore $\tilde{P}_+$ (see [19] for more details).

We also add a low pass filter F_n of suitable order n to make sure $\lim_{s\to\infty} Q < \infty$ i.e. Q is semi-proper. For the FOPDT model this means:

$$Q = \tilde{P}_-^{-1} F_n = \tilde{P}_-^{-1} \frac{1}{(\lambda s + 1)^n} = \frac{\tau s + 1}{k_p} \frac{1}{(\lambda s + 1)^1} \tag{2.19}$$

Note that the form of the filter is just a simple example with the smallest possible value for n. A good first estimate of the filter parameter for low-pass-like systems is $\lambda = {}^\tau/_2$ [20].

To accommodate for disturbances and the effects of $\tilde{P}_+$ we add feedback using the difference between the output of the model and the output of the process to make corrections to the setpoint, similar to a trim (see fig. 2.5). This feedback is the estimated disturbance $\widehat{d} = (P - \tilde{P})u + d$. It is the sum of the true disturbance and the mismatch between model and process. The controller action can be written as:

$$u = Q(r - \widehat{d}) = Q(r - (P - \tilde{P})u - d) \Rightarrow u = \frac{Q(r - d)}{1 + (P - \tilde{P})Q} \tag{2.20}$$

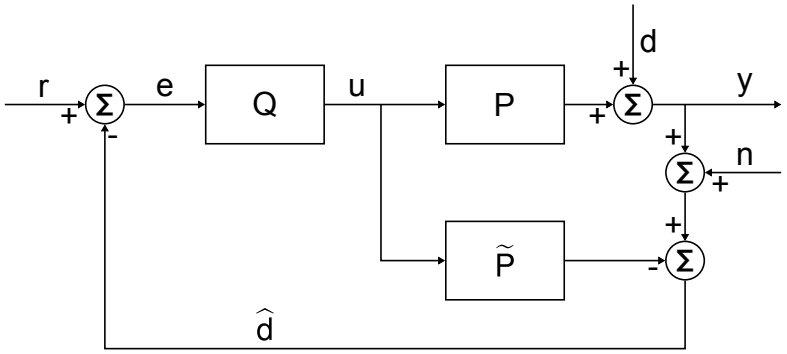


Figure 2.5.: *Block diagram of the IMC structure.* In addition to the feedforward structure it contains feedback via a sensor introducing noise n and a model $\tilde{P}$ that is used to estimate disturbance as $\widehat{d}$. The estimate is then used to obtain a trimmed setpoint e.

Via the controlled variable y we obtain the closed loop transfer function:

$$y = Pu + d = \frac{PQ(r-d)}{1+(P-\tilde{P})Q} + d = \frac{PQr + (1-\tilde{P}Q)d}{1+(P-\tilde{P})Q} \tag{2.21}$$

$$\Rightarrow T = \frac{PQ + (1-\tilde{P}Q)\frac{d}{r}}{1+(P-\tilde{P})Q} \tag{2.22}$$

In the steady state limit the non-invertible components of $\tilde{P}$ can usually be neglected. In other words $\lim_{s\to 0} \tilde{P}^{-1} = Q$. For the FOPDT model this means:

$$\lim_{s\to 0} \frac{\tau s+1}{k_p} e^{\theta s} = \frac{1}{k_p} = \lim_{s\to 0} \frac{\tau s+1}{k_p(\lambda s+1)} \tag{2.23}$$

When we use this with the expression obtained for the closed loop transfer function:

$$\lim_{s\to 0} T = \frac{PQ}{PQ} = 1 \tag{2.24}$$

So the steady state offset is zero and disturbances are fully rejected.

Considering the FOPDT model and assuming the ideal case of $P = \tilde{P}$ and $d = 0$ we find:

$$PQ = PP_-^{-1}F = P_+F \Rightarrow S = P_+F = \frac{e^{-\theta s}}{\lambda s+1} \tag{2.25}$$

Setpoint tracking is only affected by the filter F and the unavoidable dead time. This shows that, provided a good process model, the choice of F is the most influential step in our design procedure [19]. It was introduced to avoid excessive controller action and suppress model mismatch at high frequencies. But choosing it properly can also help to shape the setpoint tracking behaviour.

Let us consider a setpoint ramp input instead of a step. To obtain no steady state offset and no offset to a ramping input would require:

$$\lim_{s\to 0} S = 1 \tag{2.26}$$

$$\left.\frac{dS}{ds}\right|_{s=0} = 0 \tag{2.27}$$

Using our previous findings the second condition can be rewritten to:

$$\left.\frac{d(P_+F_n)}{ds}\right|_{s=0} = 0 \tag{2.28}$$

One possible filter that matches this condition is given as:

$$F_n = \frac{\left(2\lambda - P'_+|_{s=0}\right)s + 1}{(\lambda s + 1)^n} \tag{2.29}$$

Here n has to be chosen large enough to get a proper controller Q while also taking the derivative $P'_+|_{s=0}$ into consideration. The derivative may contain potencies of s which would increase the order of the numerator making it necessary to increase the order n of the denominator accordingly. The filter then also fulfils the condition for no steady state offset.

Considering the generality of the FOPDT model it seems reasonable to base tuning rules for a PID controller on the internal model controller developed using it. The next section will explore this subject.

2.5. PID tuning

As IMC relies heavily on a good model of the process, a measurement technique is needed to either find the model or to adjust its parameters. To obtain an equal excitation of all frequencies, an impulse, i.e. $\delta(t)$, as exciting input would be desirable. Yet such an input is impractical because of limited input amplitude. Therefore we measure the step response of the process. A step also theoretically excites all frequencies, albeit not equally, as can be seen from the Laplace transform:

$$\mathcal{L}\{\Theta(t)\} = \frac{1}{s} \tag{2.30}$$

Also for the FOPDT model all relevant parameters can be read off the time trace of the step response (see fig. 2.6). To obtain the precise values of τ and θ from the approximates we can use:

$$\tau = \frac{t_r}{-\ln(0.1) + \ln(0.9)} \approx 0.455\, t_r \tag{2.31}$$

$$\theta = t_{10} - \ln(0.9)\,\tau \approx t_{10} - 0.105\,\tau \tag{2.32}$$

This can be calculated from the fact that the step response has the form $k_p\left(1 - e^{-t-\theta/\tau}\right)$.

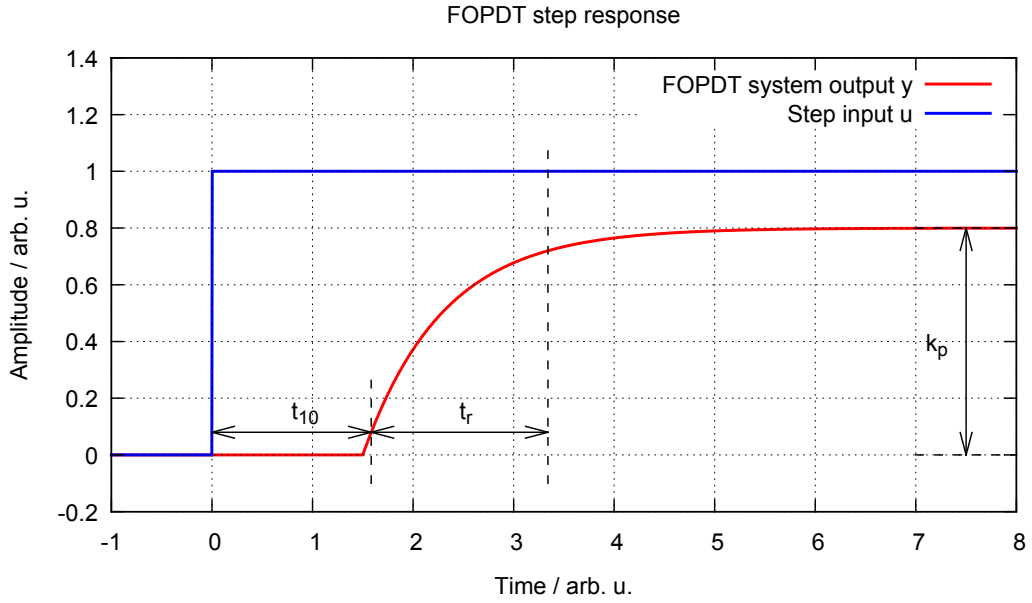


Figure 2.6.: *Step response of a first order plus dead time system.* The steady state gain can be calculated from the final values $k_p = {}^{y_f}/_{u_f}$ that are reached after sufficiently long time $\gg \theta + \tau$. The dead time θ equals the time from the input step to the onset of the response. In practice noise impedes finding the onset of the response. Instead usually the time t_{10} from the step to 10% of final output y_f is measured. From this point the time $t_r \approx 2\tau$ to 90% of y_f called rise time is measured.

We can also evaluate the performance of the control loop based on the closed loop step response. There is a plethora of criteria that may or may not apply to a certain application (e.g. overshoot, rise time, settling time etc.). A relatively general figure of merit is based on the idea that y should reach its desired value $y = r$ as soon as possible after the step, ideally without deviating after reaching it. To measure this, the square error $e^2 = (r - y)^2$ is integrated and yields the Integrated Square Error (ISE) [22]:

$$\text{ISE} = \int_0^\infty e^2(t)dt \tag{2.33}$$

In practice the upper limit of the integral is a time that is sufficiently larger than all process time constants. The ISE should be minimized.

In addition we can obtain the impulse response as the first time derivative of the ideal step response. Let the impulse response of a system, e.g. the process, be denoted by $P(t)$. The Laplace transform then gives us the transfer function of the system:

$$\mathcal{L}\{P(t)\} = P(s) \tag{2.34}$$

A real measurement will only give a good estimation for frequencies $f \ll f_c = 1/2\pi\tau$ assuming the real step behaves like an ideal step filtered by a first order low-pass with the time constant τ.

This can also be used to find the closed loop transfer function of a feedback control system. Employing Fourier transform to probe the values of the Laplace transform on the imaginary axis of the s-plane we can create a bode plot of the transfer function. The frequency at which gain falls below -3 dB for the first time is called the control bandwidth. The specific gain of -3 dB is chosen by convention (in analogy to the low-pass transition frequency). It indicates that the response is too slow to reasonably track inputs of frequencies higher than this frequency.

We might also be interested in the lag introduced by the closed loop system. We can use the frequency at which the phase falls below $-180°$ as indication. For a pure delay by θ we find:

$$T(t) = \delta(t-\theta) \xrightarrow{\mathcal{L}} T(s) = e^{-\theta s} \tag{2.35}$$

Then the phase of $T(i\omega)$ is $-180°$ exactly for:

$$-\theta\omega_{-180°} = -\pi \Rightarrow \omega_{-180°} = \frac{\pi}{\theta} \Rightarrow f_{-180°} = \frac{1}{2\theta} \tag{2.36}$$

Given the dead time of a process we can compare this frequency with the $-180°$ cross over in the closed loop response to find how close the controller comes to the absolute limit set by the process dead time.

Some additional figures that help determine if the system is stable cannot be found in the bode plot of the open loop transfer function. Rearranging the general relation of closed loop to open loop transfer function (eq. 2.10) we get:

$$PC = \frac{T}{1-T} \tag{2.37}$$

which is the open loop transfer function. This is helpful to measure the transfer function of a controller with integral action. Without feedback it integrates any small offset on the input until the output is saturated. Therefore direct measurement techniques are harder to apply.

The frequency f_1 at which the gain crosses 0 dB (i.e. a gain of 1, unity gain) and the frequency $f_{\pm 180°}$ at which the phase crosses $\pm 180°$ are of special interest. Should they be the same, the transfer function would be PC $= -1$ for this frequency and the closed loop transfer function would go to infinity (denominator 0). This has to be avoided. Simple systems (like an FOPDT system) that do not have multiple crossover frequencies are stable if $f_1 < f_{-180°}$. How close the system is to instability is quantified by the phase and gain margins. The phase margin is defined as:

$$\Delta\phi = \phi(f_1) - (-180°) \tag{2.38}$$

with the phase $\phi(f)$ of the transfer function as a function of frequency. And the gain margin is defined by:

$$\Delta g = -g(f_{-180°}) \tag{2.39}$$

with the gain $g(f)$ in dB as a function of frequency. If they are both < 0 and gain reduces with increasing frequency (which is typically the case for the simple systems considered here) the above criterion is fulfilled and the closed loop system is stable. In practice to achieve a smooth step response a phase margin of 45° - 60° and a gain margin between 5 dB and 10 dB are recommended [23].

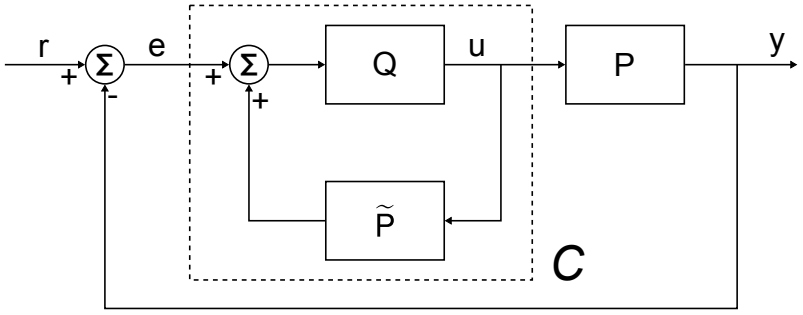


Figure 2.7.: *IMC scheme reordered to show equivalence to classic feedback control scheme.* The order of subtractions is switched with respect to fig. 2.5 and disturbance and noise are not shown. The equivalent to the classic controller denoted C has an internal feedback loop. Also note that e is the error signal in the traditional sense as in the case of PID.

As can be observed in fig. 2.7 the IMC structure can be rearranged to yield the classic feedback structure [19]. Classic here means an error input $e = y - r$ that is driven to 0, as in the PID scheme. The classical controller is then:

$$C = \frac{Q}{1 - \tilde{P}Q} \tag{2.40}$$

For a low-pass without time delay this equation yields:

$$C = \frac{\tilde{P}^{-1}F}{1 - \tilde{P}\tilde{P}^{-1}F} = \frac{\tau}{k_p \lambda}\left(1 + \frac{1}{\tau s}\right) \tag{2.41}$$

which is the s-space representation of a PI-controller. The tuning parameters can simply be read off (compare eq. 2.41 and eq. 2.11):

$$k_c = \frac{\tau}{K\lambda} \tag{2.42}$$

$$\tau_i = \tau \tag{2.43}$$

Unfortunately this is not as easy when dead time is taken into account. To be able to relate the corresponding IMC design to a PID we first need to linearise the dead time term. To do this the first order Padé-approximation is commonly used [19]:

$$e^{-\theta s} = \frac{1 - \theta s/2}{1 + \theta s/2} \tag{2.44}$$

This is then added into P and $\tilde{P}$ in eq. 2.41 and we get:

$$C = \frac{\tilde{P}_-^{-1} F}{1 - \tilde{P}\tilde{P}_-^{-1} F} = \frac{\tilde{P}_-^{-1}}{F^{-1} + \tilde{P}_+} = \frac{1/k_p(\tau s + 1)}{\lambda s + 1 - \frac{1-\theta s/2}{1+\theta s/2}} \tag{2.45}$$

By factorising out a low pass filter and decomposing the other factor in terms of potencies of s we finally find an equation of the form:

$$C = \frac{k_c}{\tau_f s + 1}\left(\tau_d s + 1 + \frac{1}{\tau_i s}\right) \tag{2.46}$$

which is a PID controller with a low-pass filter and coefficients:

$$k_c = \frac{\tau + \theta/2}{k_p(\lambda + \theta)} \tag{2.47}$$

$$\tau_i = \tau + \theta/2 \tag{2.48}$$

$$\tau_d = \frac{\tau\,\theta/2}{\tau + \theta/2} \tag{2.49}$$

$$\tau_f = \frac{\lambda\,\theta/2}{\lambda + \theta} \tag{2.50}$$

Note that the choice of λ only affects k_c and τ_f. As explained in sec. 2.3 most PID controllers use a filter on the D-part but in general not on the other (PI) parts. For $\tau \approx 2\lambda \gg \theta$ this is reasonable. In addition the opposite case $\tau \approx 2\lambda \ll \theta$ leads to a tuning that is effectively a PI controller. This might be one of the main reasons why differential action often does not seem to improve the behaviour of a control loop in practice. Based on our considerations we can find out if this is the case just based on a reasonably good model of the process to be controlled.

Analogous to the derivation of the PID tuning above we can find a PI tuning for the FOPDT model:

$$k_c = \frac{\tau + \theta/2}{k_p\lambda} \tag{2.51}$$

$$\tau_i = \tau + \theta/2 \tag{2.52}$$

Further research has shown how to best choose the tuning parameter λ in terms of the dead time θ. For the PID tuning $\lambda/\theta = 0.8$ is recommended and for the PI tuning $\lambda/\theta = 1.7$ [19].

Since our considerations up to this point required the systems to be LTI systems we did not yet consider the advantages of digital control in case of a controlled system that exhibits nonlinearity. This will be subject of the next section.

2.6. How to deal with nonlinearity

All systems used in the previous descriptions were LTI systems. In reality neither linearity nor time invariance is a given. For many real world systems time invariance is reasonably

well met. Slight deviations may be treated as disturbances.

Linearity is more rarely well met. In nonlinear systems the output amplitude for a given signal shape is not linearly related to the input amplitude. One possible way to deal with this problem is gain scheduling. We simply identify different sets of parameters for which the system behaves approximately linear and find a linear controller that works well for this set of parameters [24].

Using a digital controller and if the Process simply exhibits a non-linear steady state gain this may be simplified even further. Based on the philosophy of IMC we just write the steady state output of the process as a function of the input:

$$y = \kappa_p(u) \tag{2.53}$$

Factorising the process in an s dependent and a static part lets us replace the reciprocal of the process gain in the controller Q by the inverse of $\kappa_p(u)$:

$$Q = \kappa_p^{-1}(u)\ \tilde{P}_-^{-1}\Big|_{k_p=1} F \tag{2.54}$$

To implement such a measurement routine might not be necessary when using a PID controller but is vitally important for the direct implementation of an IMC controller as was explained earlier (see 2.4).

3. Discrete time control theory

3.1. Conversion between continuous and discrete time signals

To be able to work with a digital system that is deterministic (predictability) and internally free from potentially stacking noise sources (scalability) we first have to pay a price [8]. As long as the process we control is not digital itself, we have to convert the analogue process variable y to a digital signal and then convert the digital control action into an analogue signal u. The price is that the digitised signals are always only an approximate of the analogue ones interacting with the process. Devices making such conversions are called Analogue to Digital Converter (ADC) and Digital to Analogue Converter (DAC). Here we will not go into the details of their operating principles but rather treat them as black boxes.

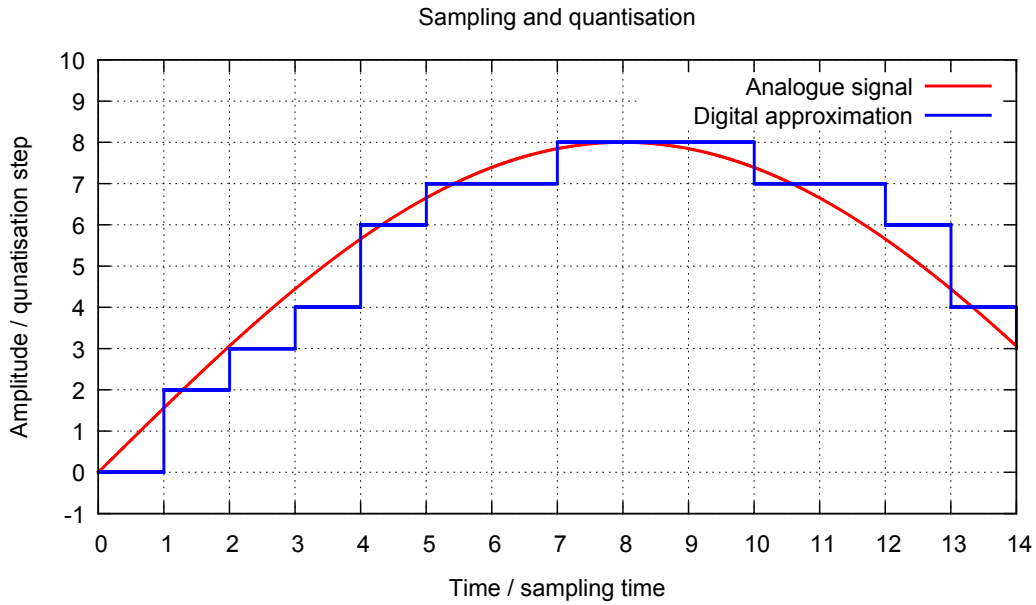


Figure 3.1.: *Effects of sampling and quantisation.* This example shows part of a continuous time and amplitude sine function and its digital approximate. To make the effects visible a very coarse discretisation was chosen.

We consider a signal to be digital if its amplitude takes discrete values at discrete points in time (see fig. 3.1). Therefore the two most important characteristics of the aforementioned converters are sampling rate (discrete time-points per time unit) and resolution (discrete

amplitude values per amplitude unit). Let us first explore the latter.

The nominal resolution of the converters is given by the number of bits of the binary number they may give (ADC) or take (DAC) as input or output respectively. This is a resolution relative to the maximum values the analogue signal may take. Therefore, as an example, a resolution of 14 bits with a voltage span from -1 to +1 V corresponds to:

$$14\,\text{bit} = \frac{1}{2^{14}} \approx 0.061\,‰ \rightarrow 0.061\,‰ \cdot 2\,\text{V} = 0.122\,\text{mV} = 1\,\text{LSB} \tag{3.1}$$

This step in amplitude is the smallest possible one for a converter with a given resolution. It corresponds to a change of the rightmost cipher in the binary number describing the signal amplitude. This so called Least Significant Bit (LSB) is often used as unit of amplitude in this context (as in [25] [26]).

With the assumption that the analogue signal changes slowly compared to the sampling rate we may for now ignore the time discretisation. We find that our discrete amplitude signal will deviate by up to $\pm 1/2$ LSB from the analogue signal. From this fact one can derive a white noise power generated when an analogue sine signal with a given amplitude is converted which is called quantization noise [26]. Then a ratio between sine signal power and noise power can be found called Signal to Noise Ratio (SNR). Therefore a certain SNR can be associated with the resolution measured in number of bits N:

$$\text{SNR} = 6.02\,N\,\text{dB} + 1.76\,\text{dB} \tag{3.2}$$

When using this ideal relation to characterise the resolution that is available from a real world converter based on a measurement of the SNR, N is called Effective Number Of Bits (ENOB) [25]. This number is often lower than the nominal resolution because of other noise sources besides the quantization noise.

Since oscillations are usually undesired in control applications a characterisation of the static (i.e. no change in the analogue signal) behaviour is also important. We define an effective resolution [25] based on the assumption that we deal with white noise. We can determine the root mean square (rms) amplitude of this noise either directly or using its total power (within a given spectral range). Then the effective resolution is:

$$N_{\text{eff}} = \log_2\left(U_{\text{rms}}/U_{\text{LSB}}\right), \quad U_{\text{rms}} = \sqrt{P_{\text{noise}} \cdot |Z|} \tag{3.3}$$

where $|Z|$ is the magnitude of the impedance over which the noise measurement is taken (usually 50 Ω).

For our considerations of time discretisation let us assume a continuous amplitude. Given an infinite discrete time sampling x_n of an infinitely long signal $x(t)$ we want to estimate the Fourier transform $X(f)$ of this signal. We use the the Discrete Time Fourier Transformation (DTFT) [27] with the sampling time T and find:

$$\widetilde{X}(f) = \sum_{n=-\infty}^{+\infty} x_n \cdot e^{-2\pi i n f T} = \sum_{k=-\infty}^{+\infty} X\left(f - k/T\right) \tag{3.4}$$

If $X(f)$ is sufficiently small for $f > f_N = {}^{f_S}/_2$ then $\widetilde{X}(f)$ is a good estimate for $X(f)$ up to f_N.

This fact is closely related to the Nyquist-Shannon theorem. It states that we can only measure frequencies smaller than half the sampling rate:

$$f_N = \frac{1}{2} f_S \tag{3.5}$$

Therefore f_N is often called Nyquist frequency.

Since we always deal with finite time signals in practice we use the Discrete Fourier Transformation (DFT) to sample the DTFT based on a finite discrete time measurement $x_0, x_1, \ldots, x_{N-1}$:

$$X_g = \sum_{n=0}^{N-1} x_n \cdot e^{-2\pi i g n / N} \tag{3.6}$$

where $g \in \mathbb{Z}$ are discrete frequencies and $X_g \in \mathbb{C}$ are spectral amplitudes.

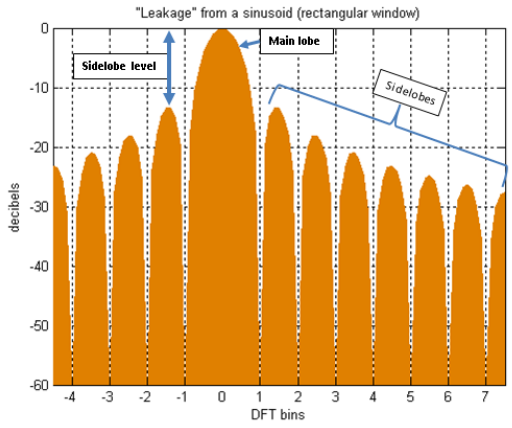


Figure 3.2.: *Spectral leakage from a sinusoid using a rectangular window [28].* The graph shows the continuous frequency DTFT (orange) and the points at which a DFT would sample it (DFT bins).

Simply transforming a finite time measurement is equivalent to multiplying the original infinite signal with a rectangular function and then sampling the DTFT using the DFT. The rectangular window function may be given in terms of two step functions:

$$w_r(t) = \Theta(t) - \Theta(t - (N-1)T) \tag{3.7}$$

This estimation will show two distinct deviations from the true spectrum. On one hand for a spectrally sharp signal as a sinusoid spectral leakage will occur. Which means that frequencies other than the frequency of the sinusoid also show a value $\neq 0$ although they are $= 0$ for the true spectrum. The magnitude of these amplitudes depends on the window

function used [29].

Another deterioration occurs when the DFT sampling does not coincide with the maximum of the main lobe, in the worst case resulting in two equally high amplitudes in the DFT spectrum. This effect is called scalloping loss. The width of the main lobe and how flat the top is can also be influenced by window functions [29].

If there is white noise present in the measured signal its amplitude in the DFT spectrum will also be affected by the chosen window function. The so called Hann window is such a function. It is given in terms of the samples in the measurement:

$$w(t) = \cos^{\alpha}\left(\frac{2\pi n}{N-1}\right) \tag{3.8}$$

For example using the Hann window with $\alpha = 2$ will yield a Noise Spectral Density (NSD) that is 1.5 times higher than the true NSD [29]. For the rectangular window this factor is 1. Using the Hann window function could be compared to having the noise recorded with a rectangular window with wider bins. Therefore this factor is also called the Equivalent Noise Bandwidth. Also a maximum scalloping loss of 1.42 dB can be determined for the Hann window while the maximum scalloping loss for a rectangular window is 3.92 dB [29].

3.2. PID in discrete time

To take advantage of the unique properties of digital signals, we have to translate the control schemes presented in 2 to discrete time. A reasonable first approximate for the ideal PID (see also eq. 2.2) may be written as [30]:

$$C_n = k_c\left(e_n + \frac{T}{\tau_i}\sum_{i=0}^{n} e_i + \tau_d\frac{e_n - e_{n-1}}{T}\right) \tag{3.9}$$

The proportional part remains unchanged besides discretisation. The indefinite integral is replaced with a sum starting at $t = nT = 0$ where the controller is switched on. Also the time differential that is part of the integration is translated to a multiplication of the sum with the sampling time step T. The time differential is replaced with a backwards difference.

To evaluate how good this approximation is and to add the low pass filter on the differential part that is missing we have to use frequency space as before. Similar to the way the DTFT acts as the complement of the Fourier transform in discrete time, the so called Z-transform is the discrete time complement of the Laplace transform. It is defined as:

$$X(z) = \sum_{n=-\infty}^{+\infty} x_n z^{-n} \tag{3.10}$$

It maps the s-plane such that the imaginary axis from $-2\pi f_N$ to $-2\pi f_N$ is mapped to the unit circle. The part of the right half part of the band delimited by $Im(s) = \pm 2\pi f_N$ is mapped to the inside of the unit circle while the left part is mapped to the outside [31].

3.3. Digital filters for control applications

In general digital filters are classified as either Finite Impulse Response (FIR) or Infinite Impulse Response (IIR) filters. The difference is that a FIR filter hast an impulse response that is non-zero only for a finite number of samples. It only takes a limited number of input samples to determine its output e.g. the past 4 samples. An IIR filter on the other hand in general has an infinite number of non-zero impulse response output samples. Such a filter takes both past input and output samples into account i.e. it contains feedback.

In general FIR filters need a lot more input samples to accomplish a similarly good approximation to the target continuous time spectral shape as an IIR filter. This means IIR filters will have a lower internal delay time which makes them invaluable for control applications. However they require more considerate design, since they can become unstable and require higher internal precision [32].

Considering the way s is mapped to z we use a procedure called zero-pole-matching [33] to construct our simple IIR first order low-pass filter:

1. Determine s-space poles and zeros.
2. Map poles and zeros to z-space using $z = e^{sT}$.
3. Based on these and additional information on the gain (low frequency limit gain for a low pass) form the z-space transfer function.
4. Translate this function into a difference equation.

From previous considerations (see eq. 2.9) we know a first order low pass with time constant τ has a pole at $s = -1/\tau$. Therefore the pole of the digital filter should be at $z = e^{-T/\tau}$ giving the transfer function:

$$F(z) = \frac{\widetilde{K}}{z - e^{-T/\tau}} \tag{3.11}$$

From here we can compare the low frequency limit to find the gain constant $\widetilde{K}$. As $lim_{s\to 0}\, z = lim_{s\to 0}\, e^{sT} = 1$ the low frequency limit is:

$$lim_{z\to 1} F(z) = \frac{\widetilde{K}}{1 - e^{-T/\tau}} \tag{3.12}$$

compared to 1 for the s-space transfer function. This gives $\widetilde{K} = 1 - e^{-T/\tau}$. To find the difference equation we have to express eq. 3.11 in terms of z^{-n} (which corresponds to a time delay by nT):

$$F(z) = \frac{\left(1 - e^{-T/\tau}\right) z^{-1}}{1 - e^{-T/\tau} z^{-1}} = \frac{x(z)}{y(z)} \tag{3.13}$$

The nth output y_n can then be expressed in terms of inputs x_m, $m \leqslant n$ and outputs y_l, $l < n$:

$$y_n = e^{-T/\tau} y_{n-1} + \left(1 - e^{-T/\tau}\right) x_{n-1} \tag{3.14}$$

Device	Advantage	Disadvantage
PC (Central Processing Unit)	High clock speed, good availability	Few data per cycle, large overhead
Digital Signal Processor	Massively parallel, low power consumption	Limited utility
Field Programmable Gate Array	Reprogrammable logic	High power consumption
Application-Specific Integrated Circuit	Custom logic, low power consumption	High development cost, long development time

Table 3.1.: *Comparison of typical devices used in digital control.* The properties collected in this table are only very rough descriptions or tendencies. They represent the state of technology of recent years [34], [35].

This can be used to filter the differential part of the PID and to construct the IMC controller based on the FOPDT model.

Designing digital filters can be very involved and one should be aware of the possible pitfalls. Nevertheless software is available to help with and speed up all steps of this process (e.g. MATLAB). To realize the digital controller, a device that is suitable for the algorithm described has to be selected and programmed which will be the subject of the next section.

3.4. Digital control hardware and programming

The hardware used to realize the digital controller contains a Field Programmable Gate Array (FPGA) which is an integrated circuit that can be reprogrammed to represent complex logic. This is usually done by describing the required logic using a Hardware Description Language (HDL) and then having software convert this description into a combination of connected logic gates and memory units.

A comparison of other devices used in digital control is shown in tab. 3.1. FPGAs avoid high upfront costs application-specific integrated circuits (ASIC) [34] and do not have the overhead problems often encountered on PCs (limiting control bandwidth due to delay amongst others). However, the line between FPGA and Digital Signal Processor (DSP) is blurred by the fact that current high-end FPGAs implement pre-manufactured DSP hardware-blocks to be connected to the remaining design. Since the edge in power efficiency that DSPs have is rarely a concern FPGAs tend to be the best platform for digital control applications in experimental physics [35].

To implement algorithms developed above the HDL Verilog [36] was used. It is in some regards similar to the commonly known programming language C. Much like C it tends to keep quite close to the hardware and provides fine control over the way the available resources are used. This enables highest performance but expert knowledge and considerable time investments are necessary to achieve this [37].

As for digital filter design there is software available (e.g. MATLAB) to generate HDL code from high level languages. This takes some fine control and possibly performance away but requires less detailed knowledge and time investment making this an interesting alternative for many experimental physicists.

This short description of digital control hardware in general directly leads into the following part about the specific setup used here to explore the practical benefits and drawbacks of digital control for quantum optics experiments.

4. Building a digital control test set-up

4.1. Red Pitaya as a platform for digital control

As the platform for the implementation the Red Pitaya board was selected [38]. It is very versatile featuring a Xilinx Zynq 7010 System on Chip as its main computing unit [39]. The Zynq joins two ARM microprocessor cores and a Field Programmable Gate Array on one die. It runs a Linux OS using the ARM cores, providing an environment that is familiar to most programmers. There is also an Ethernet connection allowing most data transfers and commands to be run using ssh. In addition the ARM cores are well suited for tasks that require sequential programs (most user interface applications, some algorithms). However for parallel computing and time-critical applications like high bandwidth control (or signal processing more generally) the FPGA is the more appropriate choice.

An FPGA can be electronically programmed to contain any kind of digital logic based on a connection of logic gates making it most flexible. Modern FPGAs as the one used here usually also have some more specialized hardware for digital signal processing (e.g. pre-made integer multiplier units and RAM blocks) to help implementing typical applications more efficiently both in terms of electrical power and in terms of number of FPGA gates used (see also sec. 3.4).

Most FPGA vendors today offer software tools to implement a design using block diagrams. This offers an easy start for anyone more familiar with analogue and small scale (in terms of gates) digital electronics. However for large projects and a precise control over the details of the implementation of the design (e.g. whether general purpose flip-flops or block RAM is used as memory) the use of a Hardware Description Language (HDL), in this case Verilog, is more convenient. In addition in the case of the Red Pitaya there is already a large amount of code written in Verilog implementing some basic functionalities.

The Red Pitaya also features a 125 Msps two channel Analogue to Digital Converter (ADC) with 14 bit nominal resolution and a Digital to Analogue Converter (DAC) with the same nominal specifications [40], [41]. Thanks to all these features united on one board the Red Pitaya can be used for various applications as demonstrated by software available from the manufacturer including oscilloscope, arbitrary waveform generator, spectrum or network analyser. However it cannot compete with high end laboratory equipment specialized in these areas of course.

It is important to measure the characteristics of the converters since they are crucial components of the digital control loop (see sec. 3.1). In the following section I focus especially

on the noise characteristics, since harmonic distortions are not as significant for most control applications.

4.2. Characterization of the Red Pitaya AD and DA converters

To characterise the performance of the converters on the Red Pitaya board various power spectra are measured. Where applicable we use a network analyser of the type HP 3589A [42]. It is also used to measure the gain of pre-amplifiers. For reasons explained in more detail in [43] the operator is required to turn off functions that enable detection of monochromatic signals in wideband spectra to measure rms noise spectral density. This was done unless otherwise noted.

Analogue to Digital Converter

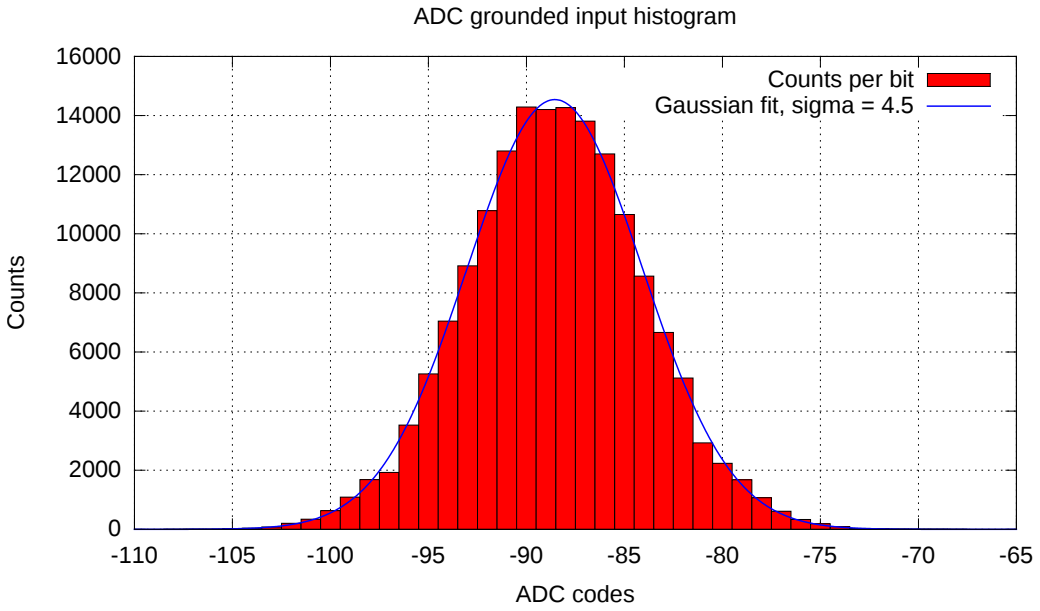


Figure 4.1.: *Distribution of codes measured by the ADC for grounded input.* It depicts the results of ten measurements with 16k samples each. The standard deviation is determined from the Gaussian fit to the histogram.

To measure the intrinsic noise on the ADC input for zero input voltage we simply apply a short circuit termination to the input [25]. Then the FPGA reads the digital values (so called ADC codes) that the ADC puts out at the full sample rate of 125 Msps (125 Mega-samples per second). This is done using the acquire utility supplied with the board. 2^{14} samples were recorded which corresponds to a time interval of 131 μs. The result is a histogram (see fig. 4.1) with an approximately Gaussian distribution of the number of times the respective codes were recorded. Fitting a Gaussian to the histogram we find the standard deviation of

the noise distribution in units of ADC codes:

$$\sigma_{\mathrm{ADC}}|_{\omega=0} = 4.5\,\mathrm{codes} \tag{4.1}$$

To get a more intuitive measure we can convert the standard deviation to an effective number of bits N_{eff} [25]:

$$4.5\,\mathrm{LSB} \rightarrow N_{\mathrm{eff}} = 12\,\mathrm{bit} \tag{4.2}$$

Comparing this to the specifications of the ADC manufacturer we find by reading the histogram given in the data sheet [40] and applying the same technique as above:

$$\sigma_{\mathrm{ADC}}|_{\omega=0} = 1.2\,\mathrm{LSB} \rightarrow N_{\mathrm{eff}} = 14\,\mathrm{bit} \tag{4.3}$$

This should be understood as the upper limit that can be achieved by the ADC in question under optimal conditions that may not be achieved on a circuit board under space and economic constraints. This also applies to all following measurements that all show that the converters perform worse than specified by the manufacturer.

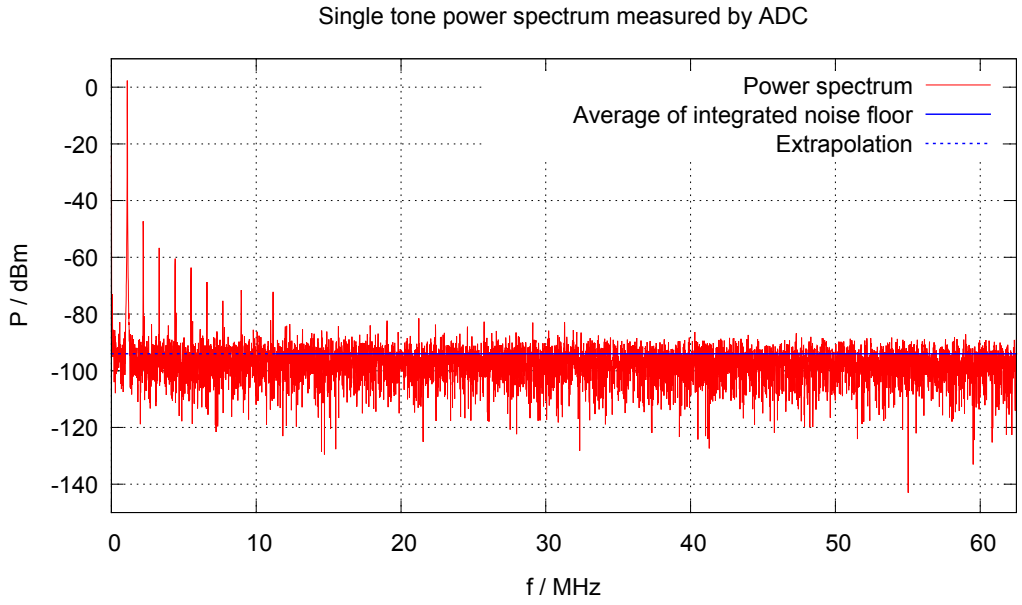


Figure 4.2.: *Spectrum of a single tone 1.1 MHz input measured by the ADC.* Many higher harmonics of the carrier are visible. For this reason the noise floor has been determined from frequencies above 11.2 MHz. Note that the resulting mean lies at the upper end of the noise range since it is shown on a logarithmic scale.

To find the maximum Signal to Noise Ratio (SNR) of the ADC we apply a sine signal from a specialized low noise sine generator (marconi 2032 [44]) to the ADC input. The amplitude is set to -1 dB Full Scale = 9 dBm, i.e. -1 dB with respect to the maximum applicable input (±1 V corresponding to 10 dBm over 50 Ω). A measurement of 2^{14} samples is taken at

full sampling rate. An FFT with Hann windowing is applied to the data and the result is presented in fig. 4.2. As described in sec. 3.1 using the Hann window can lead to a decrease in apparent carrier power of up to 1.42 dB. The observed loss is greater than expected. We also do not expect the broadening of the peak from data processing or the input signal. This is most probably due to timing jitter of the ADC. By integrating over 10 bins around the maximum a carrier power of +5 dBm can be recovered.

The measurement of the total noise power also proved difficult. Here the many higher harmonics visible in the spectrum played a role. Although the input signal contained a second harmonic at $\approx -60\,\mathrm{dBc}$ (dBc: dB relative to carrier), visible on the spectrum analyser, the higher harmonics are distortions produced by the ADC. It is very hard to separate the spectral leakage from the true noise up to approx. 11.2 MHz. For this reason spectral noise density is integrated over the frequency range from 11.2 to 62.5 MHz (Nyquist frequency for a sampling rate of 125 Msps) giving P_{partial}. To extrapolate to the full frequency span noise power we calculate:

$$P_{\mathrm{full}} \approx \frac{62.5\,\mathrm{MHz}}{62.5\,\mathrm{MHz} - 11.2\,\mathrm{MHz}} P_{\mathrm{partial}} \tag{4.4}$$

Setup	Measured resolution		Manufacturer resolution	
	relative	bit	relative	bit
ADC static	$3 \cdot 10^{-4}$	12 bit	$7 \cdot 10^{-4}$	14 bit
ADC dynamic	$8 \cdot 10^{-4}$	10 bit	$2 \cdot 10^{-4}$	12 bit
DAC static	$3 \cdot 10^{-4}$	12 bit	$6 \cdot 10^{-4}$	14 bit
DAC dynamic	$4 \cdot 10^{-4}$	11 bit	$9 \cdot 10^{-5}$	13 bit

Table 4.1.: *Summary of DAC and ADC resolution properties.* The manufacturer specifications are given int the data-sheets of the ADC and DAC devices used on the Red Pitaya board [40], [41].

From the two Powers we can now find the Signal to Noise Ratio, correcting for the Equivalent Noise Bandwidth 3.1 of the Hann window:

$$\mathrm{SNR} = \frac{P_{\mathrm{carrier}}}{P_{\mathrm{noise}}} \cdot \mathrm{ENBW}_{\mathrm{Hann}} = 62\mathrm{dB} \tag{4.5}$$

From this we can infer an Equivalent Number of Bits of the ADC (eq. 3.2):

$$\mathrm{ENOB} = 10\,\mathrm{bit} \tag{4.6}$$

For the corresponding values specified by the ADC manufacturer see tab. 4.1.

Digital to Analogue Converter

First DAC output noise for a fixed voltage is measured. With grounded input the spectrum analyser shows a Noise Spectral Density (NSD) of about -138 dB/Hz for most of its range of operation (1 kHz to 150 MHz). To detect the DAC noise floor an additional pre-amplifier

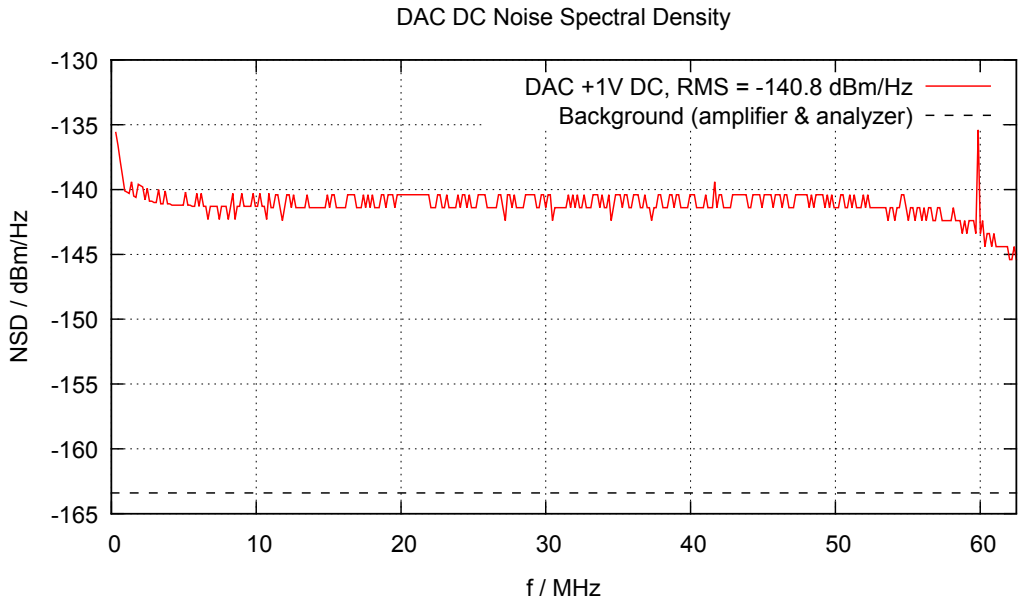


Figure 4.3.: *Noise Spectral Density (NSD) at DAC output from 300 kHz to 62.5 MHz.* This measurement refers to the NSD at the input of the pre-amplifier. The DAC is low-pass filtered on the Red Pitaya board with a transition frequency of 50 MHz causing the NSD to decrease above that frequency. The average background NSD shown is measured with the input of the amplifier connected to ground.

of type ZFL-500HLN+ [45] was used.

The pre-amplifiers nominal frequency range of 10-500 MHz does not fully cover the lower frequencies. A transmission measurement of the pre-amplifier is taken using a network analyser. The gain is 23.4 dB within the nominal frequency range and decreases to 21.6 dB at 300 kHz.

To measure the background noise caused by amplifier and analyser the input of the amplifier is connected to ground and a background noise spectrum is measured. Then the output of the Red Pitaya DAC is connected while set to a DC voltage of +1 V and a noise spectrum is measured. Both noise measurements are divided by the gain measured for the pre-amplifier.

The result is shown in fig. 4.3. From this spectrum an average noise level at the DAC output can be determined giving:

$$\mathrm{NSD}_{\mathrm{DAC,full}} = -140.8\,^{\mathrm{dBm}/\mathrm{Hz}} \rightarrow N_{\mathrm{eff}} = 13\,\mathrm{bit} \tag{4.7}$$

This assumes a flat noise spectrum and only applies to the measured frequency range (see also tab. 4.1 for comparison).

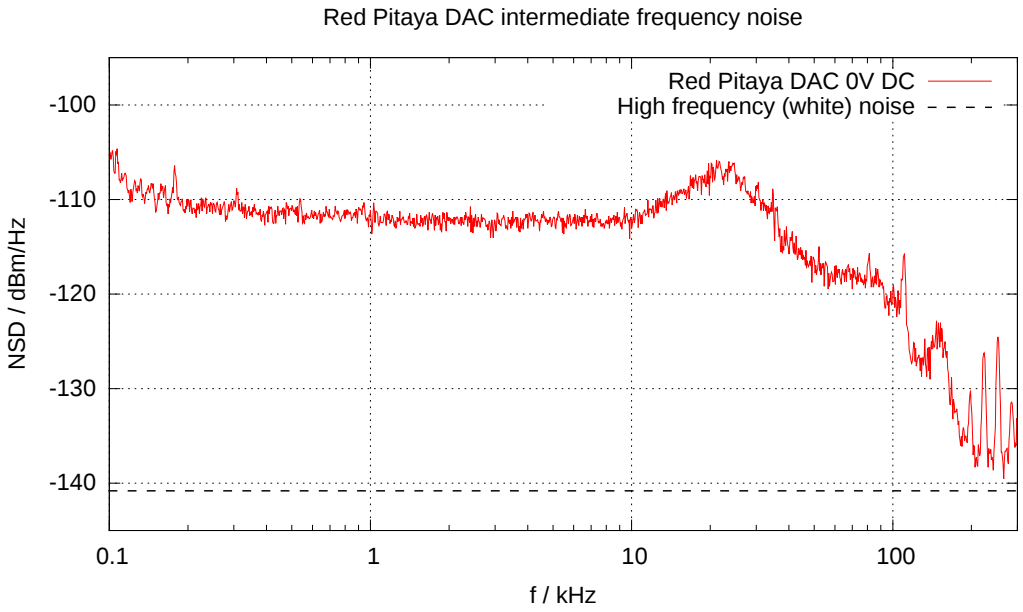


Figure 4.4.: *Output noise produced by the DAC at frequencies between 100 Hz and 300 kHz.* Note that this graph uses a logarithmic frequency scale. Above 300 kHz the amplifier used for this measurement distorts the results. Also note that this is not $1/f$ noise, which would resemble a linear function with a slope of 20 dB per decade.

The lower frequency range (0.1-300 kHz) was measured in a similar way using a Stanford Research SR560 [46] as pre-amplifier. The result presented in fig. 4.4 shows a huge excess noise below 100 kHz. For the frequency range below 40 kHz the NSD is above:

$$\mathrm{NSD}_{\mathrm{DAC,intermediate}} \gtrsim -112\,\mathrm{dBm/Hz} \tag{4.8}$$

The cause of this excessive low-frequency noise seems to lie on the Red Pitaya board. Due to the smaller frequency range the total noise power is only approximately double, yielding an effective number of bits of:

$$N_{\mathrm{eff}} = 12\,\mathrm{bit} \tag{4.9}$$

Finally a measurement of the DAC spectrum with a single tone at 110 kHz and 0 dBFS = 10 dBm output power is taken (see fig. 4.5). The noise floor was measured at 4 frequencies with an average of -136 dBm/Hz. The measurements indicate that the noise spectrum is flat. Analogous to the dynamic measurement for the ADC we integrate the NSD and find the SNR:

$$\mathrm{SNR} = \frac{P_{\mathrm{carrier}}}{\mathrm{NSD}_{\mathrm{DAC,dynamic}} \cdot 62.5\,\mathrm{MHz}/1\,\mathrm{Hz}} = 68\,\mathrm{dB} \rightarrow \mathrm{ENOB} = 11\,\mathrm{bit} \tag{4.10}$$

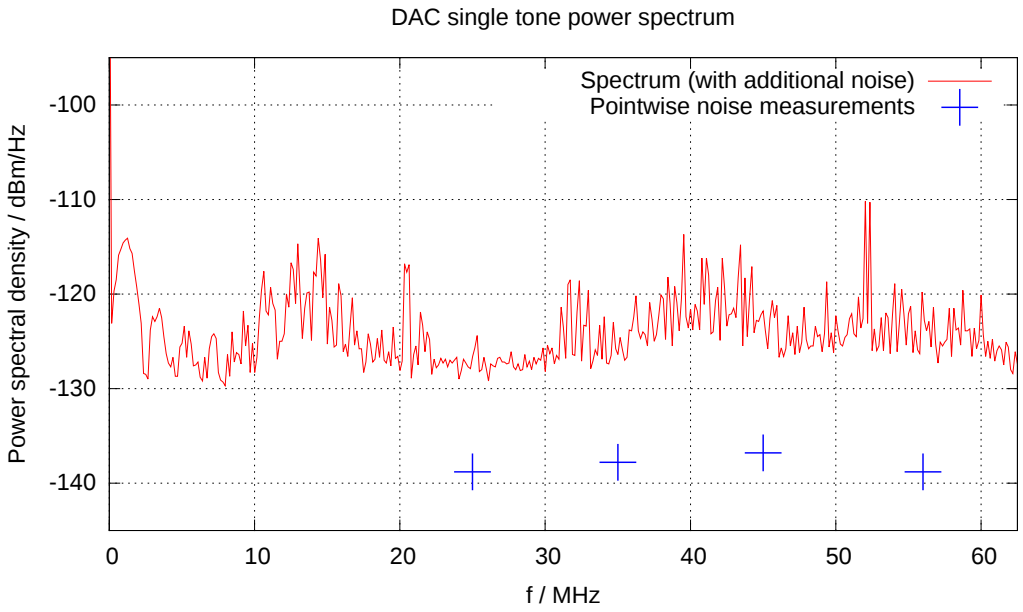


Figure 4.5.: *Spectrum with suppressed low frequency carrier and additional point-wise noise measurements.* In order to preserve the full sensitivity of the spectrum analyser, the carrier is suppressed with a simple RC high-pass filter with a transition frequency of ~25 MHz. The spectrum analyser was set to detect monochromatic signals. The point-wise measurements show the true noise level (start of sec. 4.2).

This is only a rough estimate since the noise floor may be lower for high frequencies and should also exhibit the intermediate frequency properties measured for the static case. In addition this calculation disregards harmonic distortions.

Input to output delay

The delay introduced by the entire signal chain (AD, digital signal processing, DA) is significant compared to the sampling rate of 125 Msps which corresponds to a sampling time of 8 ns. The delay of a signal from entering the ADC input over the FPGA with minimal processing to the output of the DAC is:

$$\Delta t \approx 130\,\text{ns} \tag{4.11}$$

which corresponds to ~16 sampling time steps. It limits the bandwidth that may be achievable by reducing the phase margin (see also sec. 2.5).

Although it is part of the physical device, that would usually be called the controller, it may be added to the process dead time absorbing ADC and DAC into the process. This is

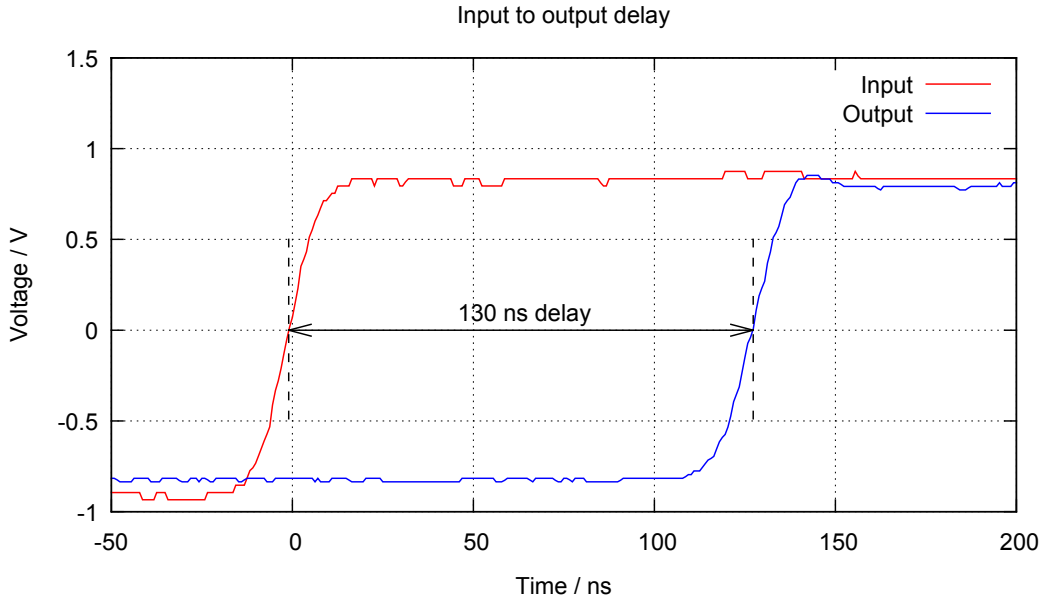


Figure 4.6.: *Delay of a signal from input to output of the Red Pitaya.* A rectangular input is sent to the Red Pitaya and also used as trigger for the oscilloscope. The delay is measured from the midpoint of the input signal to the midpoint of the output signal.

justifiable since the delay arising due to the calculations for the control algorithm (3 clock cycles at 125 MHz clock frequency i.e. 24 ns) is small compared to the delay introduced by the converters and possible processing that is done before or after the controller calculations.

The control bandwidth limitation due to the delay may be estimated by calculating the frequency at which it causes a phase of $-180°$. This is the case for:

$$f_{-180°} = \frac{1}{2\Delta t} \tag{4.12}$$

For 130 ns delay this yields a maximum achievable control bandwidth of ≈ 4 MHz (for a more details see eq. 2.36).

We may compare the results for resolution and bandwidth to the values necessary for some control applications (see tab. 4.2). We find that most applications require a better output resolution than can be achieved by the Red Pitaya. However bandwidth should not be a problem and for standard dipole traps Red Pitaya could be used without limitations.

4.3. The intensity stabilization test setup

The setup shown here (see fig. 4.7) can be used to control light power fast (≈ 1 μs) and over a wide range (> 1 decade). Usually this cannot be achieved by changing typical laser

Stabilized parameter	Input resolution	Output resolution	Bandwidth
Laser frequency via spectroscopy	10^{-2}	10^{-5}	5 kHz
Laser phase in an optical phase-locked loop	10^{-2}	10^{-5}	3 MHz
High finesse cavity length	10^{-2}	10^{-5} to 10^{-6}	50 kHz
Temperature of a laser	10^{-5}	10^{-4}	1 Hz
Intensity, high end dipole trap	10^{-4}	10^{-4}	0.5 MHz
Intensity, planned high end dipole trap	10^{-5}	10^{-5}	0.5 MHz
Intensity, standard dipole trap	10^{-3}	10^{-3}	0.5 MHz

Table 4.2.: *A table of performance requirements for different control applications in quantum optics experiments.* The basis of the shown estimates is experience in the laboratory [47]. Some simple estimates are shown in A.1. The effective resolution of the Red Pitaya DAC is not suitable for most applications while the delay is not a problem. State of the art converters can solve most of these issues (see 6).

operating parameters, e.g current, which may be used for frequency stabilisation. This is why we rely on an Acusto Optical Modulator (AOM) [49] to achieve this.

In an AOM a running acoustic wave is sent through a transparent, refractive material (usually a crystal or glass). Since density changes lead to changes in refractive index this running wave can be considered a moving refractive transmission grating. In the configuration used here the frequency of the sound wave and with it the direction of the first diffraction maximum is held constant. An aperture is then used to only transmit the first maximum. Since the fraction of light diffracted into the first maximum depends on the amount of change in refractive index induced by the density changes as a result of the sound wave we can control the amount of power that is transmitted via the intensity of the sound wave [51].

There are two characteristic time intervals related to this working principle. On the one hand the time that a change of amplitude needs to travel from its source to the beam. On the other hand the time it takes for the change to propagate through the beam diameter. Typically the first is much longer than the latter. They of course depend on how strongly the beam is focused and how far away from the sound source it passes the refractor. Based on this knowledge we can predict that the time travelling to the beam will produce a dead time. The time travelling through the beam will result in a rise of optical power in the shape of a Gaussian error function provided the beam has a Gaussian transversal intensity profile and the rise in amplitude is instantaneous.

Indeed when taking the step response of the system (see fig. 4.8) we observe a rise in optical power after some delay. It has approximately the shape of an error function. However since we use the FOPDT model we apply the rules stated for the pure exponential (see fig. 2.6) as

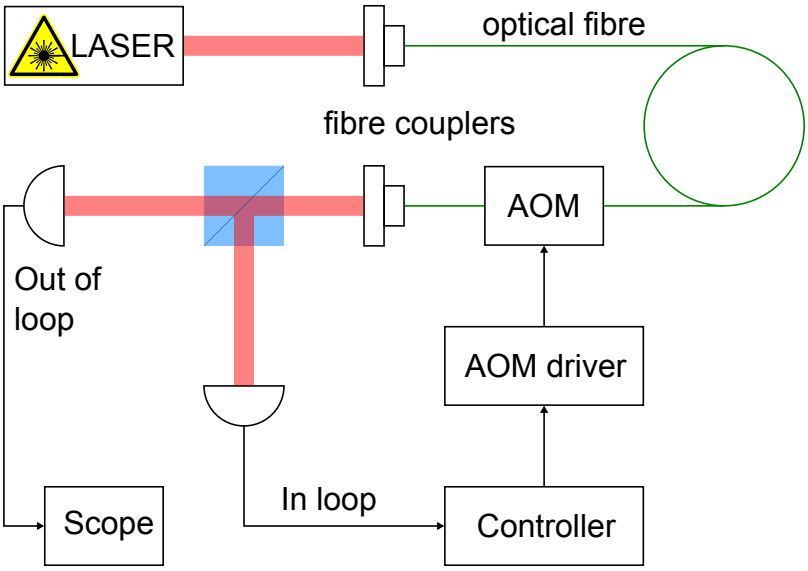


Figure 4.7.: *Setup for intensity control with fibre-coupled AOM.* The diode laser [48] with external cavity is running at 852 nm. The AOM [49] transmits a fraction of the incoming laser power depending on the electrical power it receives from the driver which is steered by the controller. Behind the AOM the beam is split into two each going on a photodiode [50]. One photodiode is used as feedback for the controller, the other to take out of loop measurements.

approximation to obtain dead time θ and characteristic time τ:

$$\theta = 693\,\text{ns} \tag{4.13}$$

$$\tau = 23\,\text{ns} \tag{4.14}$$

The fact that the error function is clearly visible suggests that the amplitude changes fast enough. Nevertheless this was also tested directly. The AOM driver has a rise time of:

$$t_{\text{r,driver}} \approx 20\,\text{ns} \tag{4.15}$$

which of course does not take the behaviour of the sound source inside the AOM into account.

The measured times can be used to tune a PI, PID or IMC controller based on the FOPDT model. As discussed (in sec. 2) this controller requires feedback which is provided by the in loop photodiode. The out of loop diode provides an independent measurement of the beam power which is needed for the evaluation of the closed loop noise characteristics. Both photodiodes are of the same type (Thorlabs PDA10A-EC [50]).

Besides the timing characteristics of the AOM the change in light power when changing the input electrical power depends on the power i.e. the steady state gain of the AOM is not linear. This leads to a nonlinear behaviour of the process.

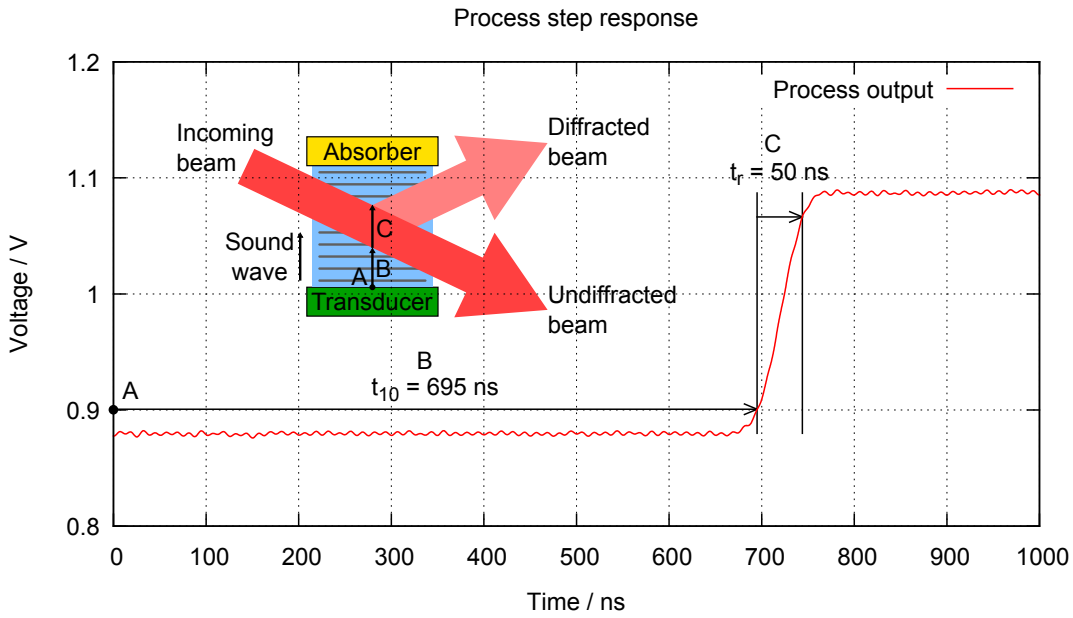


Figure 4.8.: *Process step response with AOM schematic.* The Amplitude change from the driver arrives at A. It travels (B) for the time t_{10} until a small part of the beam is affected so that power has risen to 10% of its final rise. From this time on it rises (C) within t_r until most of the beam is affected (90% of final rise).

5. Comparison of a digital and an analogue controller

5.1. Analogue reference controller

To benchmark the performance of the Red Pitaya we implement a digital PID controller compare it to a state-of-the-art analogue laser servo. For this purpose we choose the D125-2 laser servo by Vescent Photonics, as reference [52]. It is a PI^{2}D controller with a bandwidth of >10 MHz, which is sufficient for this comparison. The I^2 means that it has two integrators connected in series. The integrators and the differentiator can be set to discrete transition frequencies or deactivated individually.

Furthermore there is a voltage offset knob which corresponds to a static setpoint. For a reasonable step response measurement however an electronic input is needed. Since the available input has an extremely limited dynamic range the setpoint was instead subtracted from the photodiode signal.

In difference to the digital controller, for the Vescent laser servo the setpoint voltage has to be subtracted from the analogue voltage provided by the photodiode.

5.2. Step responses

Since our system is dead time limited we use the improved PI tuning (see eq. 2.51 in sec. 2.5). The transition frequency for the Red Pitaya can be calculated from the dead time θ including the delay introduced by the Red Pitaya (total ~820 ns) and the low pass time constant τ (~23 ns):

$$f_i = \frac{2\pi}{\tau_i} = \frac{2\pi}{\tau + \theta/2} = 373\,\text{kHz} \tag{5.1}$$

The digital controller can be set to exactly this frequency while for the analogue controller the nearest setting of $f_i = 500\,\text{kHz}$ is selected.

One can calculate an estimate of the controller gain based on the approximately linear part of the process gain k_p via:

$$k_c = \frac{\tau + \theta/2}{1.7 \cdot \theta k_p} = 0.24 \tag{5.2}$$

The digital controller can be set to this value conveniently and it only deviates slightly from the value of $k_c = 0.23$ chosen for the comparison of the controllers.

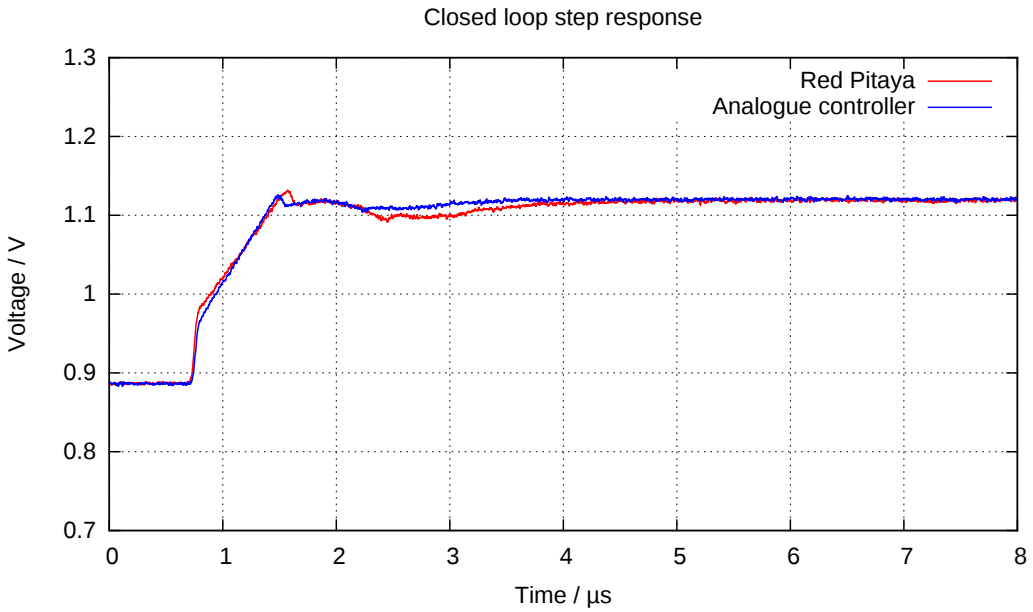


Figure 5.1.: *Comparison of closed loop step response for the two controllers in PI mode.* Both controllers are set to similar transition frequencies (Red Pitaya: 373 kHz, analogue controller: 500 kHz) while the controller gain is adjusted to get a slight overshoot of approx. 10 %. In order to suppress 80 MHz modulation caused by the AOM during analysis a digital lowpass with a transition frequency of 62.5 MHz is applied to both traces.

The step response of each lock after the described tuning are presented in fig. 5.1 and show very similar behaviour. After the process dead time (caused by AOM) has passed both responses rise rapidly because of the proportional action. From looking at the step response of the two locks it seems that both systems have the same dead time. However the trigger signal was generated using the second DAC output of the Red Pitaya. Therefore any delay between the command to change the setpoint from the PC connected to the Red Pitaya and the DAC output is not included here. The rapid rise then stops and continues linear. As mentioned in sec. 2.1 proportional action can only raise the controlled variable, in this case intensity, to a fraction of its desired value. During the dead time the error signal has not changed hence it has been constant. Therefore the integrator has applied linearly rising control action which is the reason for the linear rise of the response. When the rapid rise of the intensity is fed back to the controller after one dead time the error signal is decreased as rapidly albeit not by the same amount as in the first rise. This of course is visible in the response one dead time after the initial rise of the response as a kink downwards that ends the linear part. Which is then followed by a quadratic part, another smaller kink and so forth converging to the setpoint.

The most significant difference can be found comparing the position of the kink at the end of the linear rise. For the Red Pitaya it is slightly delayed compared to the analogue reference

by ~0.1 μs. Considering that the analogue controller may also have some delay time this is in good agreement with the added delay of the Red Pitaya measured in sec. 4.2.

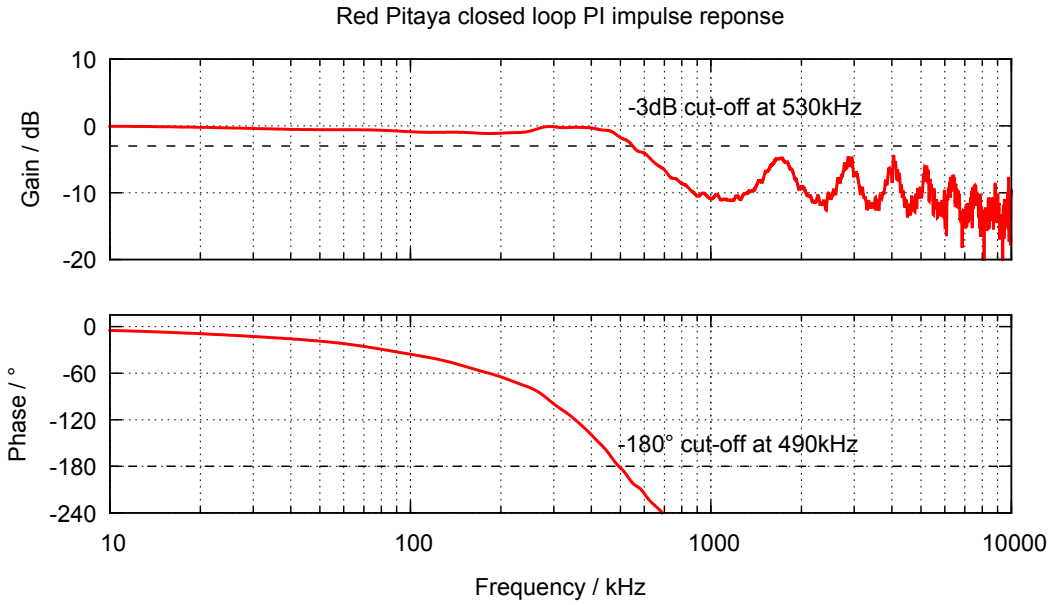


Figure 5.2.: *Bode plot of the closed loop impulse response from the step response of the Red Pitaya (see fig. 5.1).* Gain stays close to unity (0 dB) up to the lowpass-like cut-off at 530 kHz. The phase decreases approximately linear (exponential in semi-logarithmic representation). This tells us that the behaviour is dominated by the combined delay of Red Pitaya and Process (830 ns).

To evaluate control bandwidth using the closed loop step response data, we numerically compute the first time derivative of the step response and then use a DFT that gives the closed loop transfer function $T(f)$ shown in fig. 5.2 for the Red Pitaya and in fig. 5.3 for the analogue reference. As described in sec. 2.5 the frequency at which the gain falls below -3 dB indicates the control bandwidth. The bandwidth of the analogue reference is higher at 610 kHz than the one of the Red Pitaya PI which reaches 530 kHz. The difference is ~15% (of the higher bandwidth) which is also the approximate amount of dead time added by the Red Pitaya. We therefore assume this to be the origin of this difference.

Next we examine the frequency at which the phase falls below $-180°$. For the analogue reference this is the case at 500 kHz and for the Red Pitaya PI at 490 kHz. The reason this is more similar than the control bandwidth is that the delay of the initial response is not increased by the Red Pitaya. However the upper limit from the process dead time of 695 ns is 720 kHz (eq. 2.36). Neither PI controller can reach this limit because of the relatively slow linear rise that makes up the second part of their response.

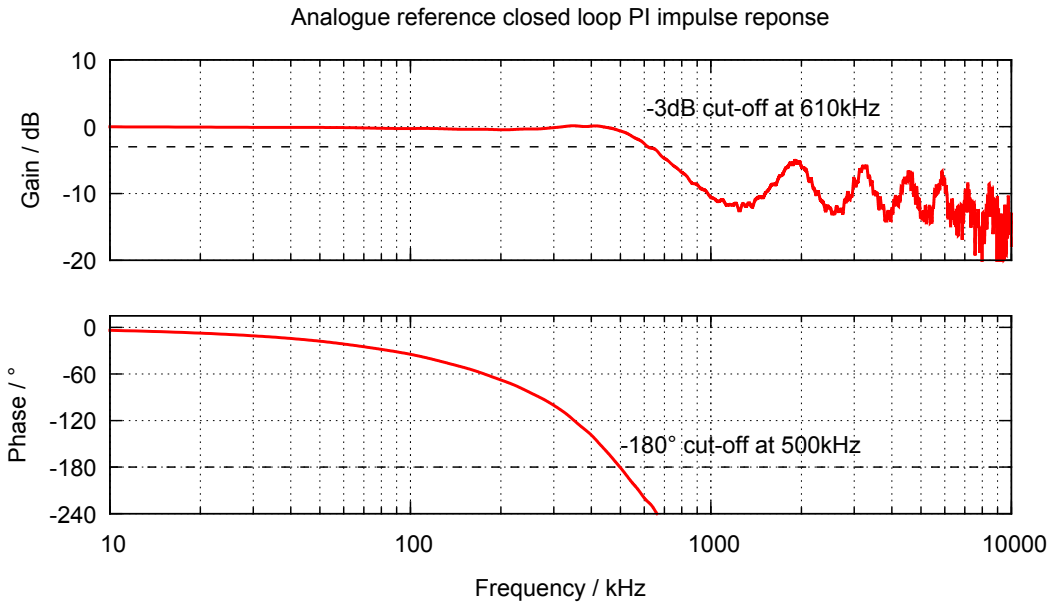


Figure 5.3.: *Bode plot of the open loop impulse response calculated from the closed loop PI step response of the analogue controller (see fig. 5.1).*

After comparing the step responses we also want to compare the noise characteristics of the two controllers in the following section.

5.3. Closed loop noise

A typical goal when building an intensity control loop is to not only stabilize the intensity at steady state but to also achieve a low root mean square (rms) noise amplitude. The ratio of this noise amplitude to the average intensity is called Relative Intensity Noise (RIN). The average intensity is considered a carrier signal at frequency zero. Therefore the spectral density of the RIN is denoted as dBc/Hz (dBc: dB relative to carrier).

There are two monitoring ports on the analogue controller where the error input signal is available amplified by 20 dB and filtered. The AC monitor covers a frequency range of 10 Hz-20 MHz and was used as pre-amplified signal for the RIN spectral density measurements with a spectrum analyser.

The closed loop RIN measurement of the two controllers in comparison shows some similarities and some differences between Red Pitaya and analogue controller (see fig. 5.4). It is important to keep in mind that both controllers do not operate at their full dynamic range and therefore not under optimal conditions. The input voltage for the controller of 0.2 V is only 10% of the input range of the Red Pitaya (±1 V) and 20% of the input range of the

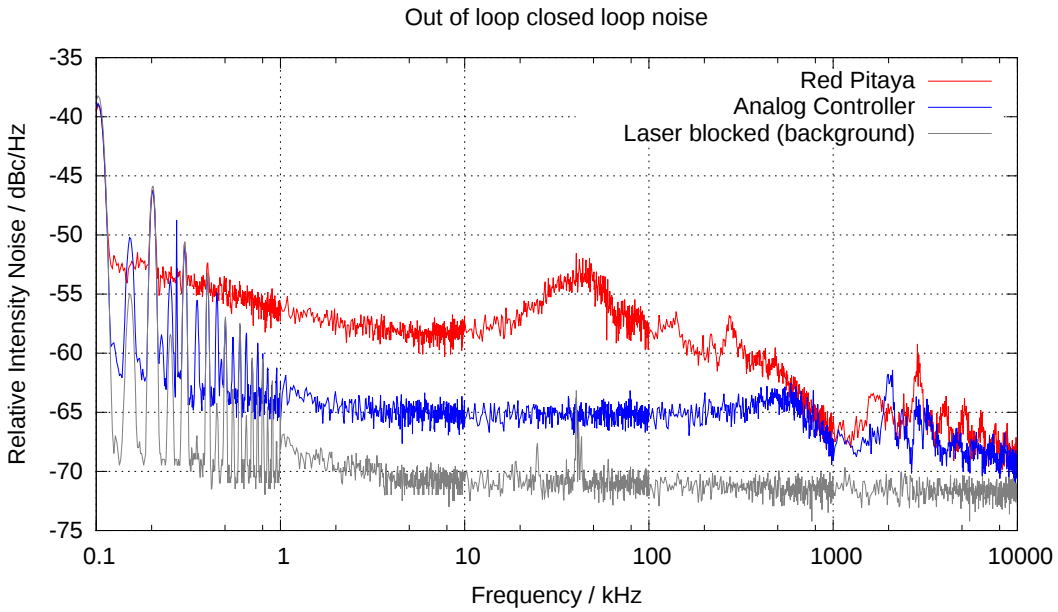


Figure 5.4.: *Relative Intensity Noise (RIN) spectral density measured via out of loop photodiode using a spectrum analyser.* The RMS Voltage noise from the bare photodiode is treated as if it was intensity noise to provide a comparable estimate of the electronic background noise. The shot noise limit (being the lowest physically achievable RIN spectral density) is not shown here since it is far lower at -147 dBc/Hz with 0.23 mW optical power at 852 nm [53].

analogue controller (±0.5 V). The output voltage is similar to the input voltage and therefore at 10% of the output range of the Red Pitaya (±1 V) and only 1% of the output range of the analogue controller (±10 V). This arguably favours the Red Pitaya since it uses a larger part of its total dynamic range.

At frequencies $< 1kHz$ residual harmonics of the mains frequency (50 Hz) are visible in all measurements including the background. This indicates that they are caused by the photodiode and subsequently imprinted on the laser power by the controller. From 1 kHz up to about 500 kHz the analogue controller shows a flat RIN spectral density approx. 5 dBc/Hz higher than the electronic background. At 500 kHz the control bandwidth limit is reached and the RIN spectral density decreases. At frequencies >1000 kHz both controllers exhibit a series of peaks of unknown origin in the RIN spectral density.

The Red Pitaya does not reach the same noise levels as the analogue reference. Its RIN spectral density is generally higher by approx. 7-12 dBc/Hz and shows a broad peak between 10 kHz and 100 kHz probably caused by the DAC noise which shows a similar spectral feature (see fig. 4.4). From 300 kHz on it decreases in a similar way as for the analogue controller.

We integrate the RIN spectrum from 1 kHz to 1000 kHz and obtain the total relative rms noise amplitude within the control bandwidth. The lower limit was chosen to exclude residual harmonics of the mains frequency. The upper limit corresponds roughly to the control bandwidth. The bare photodiode produces 0.5% relative noise. The closed loop with analogue controller is measured at 0.8% and the Red Pitaya PI at 1.5%. This measurement only provides a rough comparison and could be improved by adapting the input and output ranges and another photodiode. Nevertheless it shows that the Red Pitaya is as expected from the measurements of the converter properties not able to provide the same noise performance as the analogue reference.

The comparisons between analogue reference and Red Pitaya are made on the basis of noise and bandwidth of a simple PI controller on the Red Pitaya board. But what is possible with a more advanced algorithm? The next section will present some results regarding an internal model controller.

5.4. Application of internal model control

After comparing the capabilities of a PI controller based on the Red Pitaya with a state-of-the-art analogue controller, here we present an application in which the digital aspect of the controller plays a crucial role. A proof of concept implementation of an Internal Model Controller (IMC) based on the First Order Plus Dead Time (FOPDT) model is tested (see sec. 2.4). I measured steady state gain of the process and selected an approximately linear amplitude range to perform a step response test of the IMC.

The result of this test is shown in fig. 5.5. Owing to feedforward, the response of the intensity to a stepwise change of the set point displays approximately the same slope over most of the step amplitude. In contrast, the response of the PI controller exhibits the same steep slope only in the initial part and becomes about 15 times slower during the linear part of the response driven by the integrator (see sec. 5.2). This leads to a smaller ISE:

$$\mathrm{ISE}_{\mathrm{IMC}}(12\,\mu\mathrm{s}) = 8.0\,\mu\mathrm{s}^2, \quad \mathrm{ISE}_{\mathrm{PI}}(12\,\mu\mathrm{s}) = 8.7\,\mu\mathrm{s}^2 \tag{5.3}$$

The long term behaviour shows several kinks similar to the ones visible in the PI response. They are a clear sign of model mismatch. The internal model prediction does not fit the measured value of the controlled variable (i.e. the beam power) and the setpoint is changed to correct this (sec. 2.4).

From our theoretical understanding of the IMC scheme we can argue that the dips visible in fig. 5.5 after the initial rise of the response are indications that two parts of the model do not match the process. One is the dead time. In the shown case it is underestimated so the output of the model rises earlier and faster than the output of the process. This leads to a decrease in control action and therefore a decrease in the controlled variable (intensity) one dead time later. When the controlled variable catches up to the predicted rise shortly after (~10 ns) the control action returns to its intended value. The dead time can be changed in multiples of two clock cycles ($2 \cdot 8$ ns). Comparing the discrete resolution to the overall

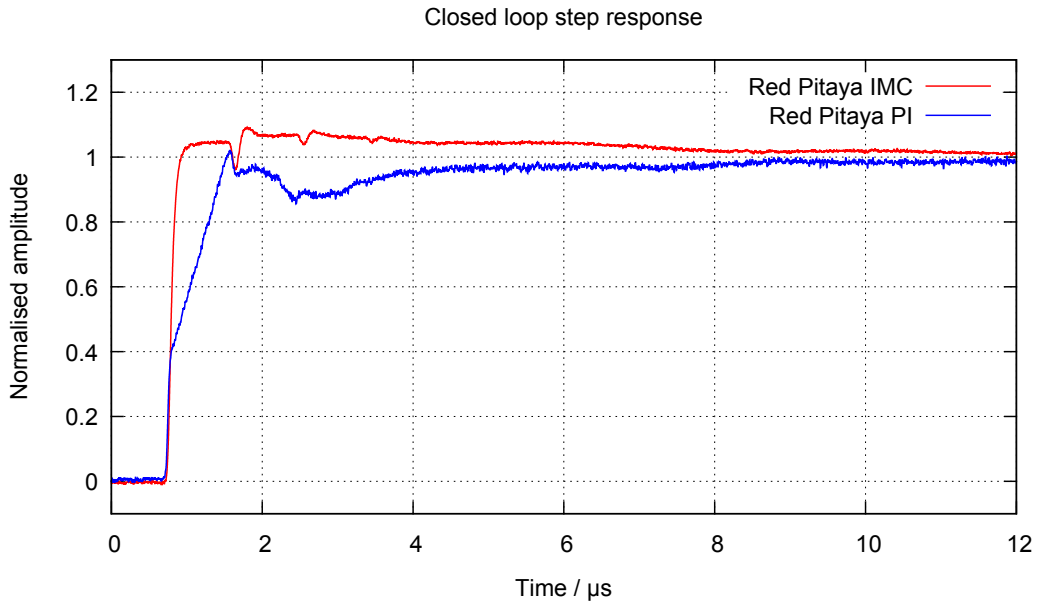


Figure 5.5.: *Normalised IMC and PI step responses using Red Pitaya.* The responses were recorded over different amplitudes and normalised to make them easier to compare. This explains why the noise associated with the PI signal appears stronger. The responses are low pass filtered as described above (see fig. 5.1).

dead time of 820 ns, one could argue that the slight mismatch is not a limitation. However, the prediction must be correct on a much shorter time scale corresponding to a fraction of the rise time of about 50 ns.

In addition the modelled response is of the from $1 - \exp(-\omega t)$ while the actual response is of the form of $\mathrm{erf}(t)$ (see fig. 4.8 and sec. 4.3). The predicted response should then rise faster than the process response in the beginning and slower in the end. This exacerbates the dip in the controlled variable introduced by the dead time mismatch and adds the slight overshoot at the end of the dip.

These considerations cannot explain, though, why the output stays above its final value for a relatively long time. This might be caused by the nonlinearity of the process gain. Since the IMC can be viewed as a classic feedback controller (sec. 2.5) we may also evaluate the closed loop impulse response of this equivalent controller in the same way as for the PI controller. The result presented in fig. 5.6 shows that control bandwidth is more than four times higher for IMC (2400 kHz) compared to PI (530 kHz).

The phase cut-off is also slightly improved from 490 kHz (PI) to 620 kHz (IMC). The limit of 720 kHz is not reached in part due to computational delay caused by the IMC algorithm (~24 ns) and various other effects that require further investigation.

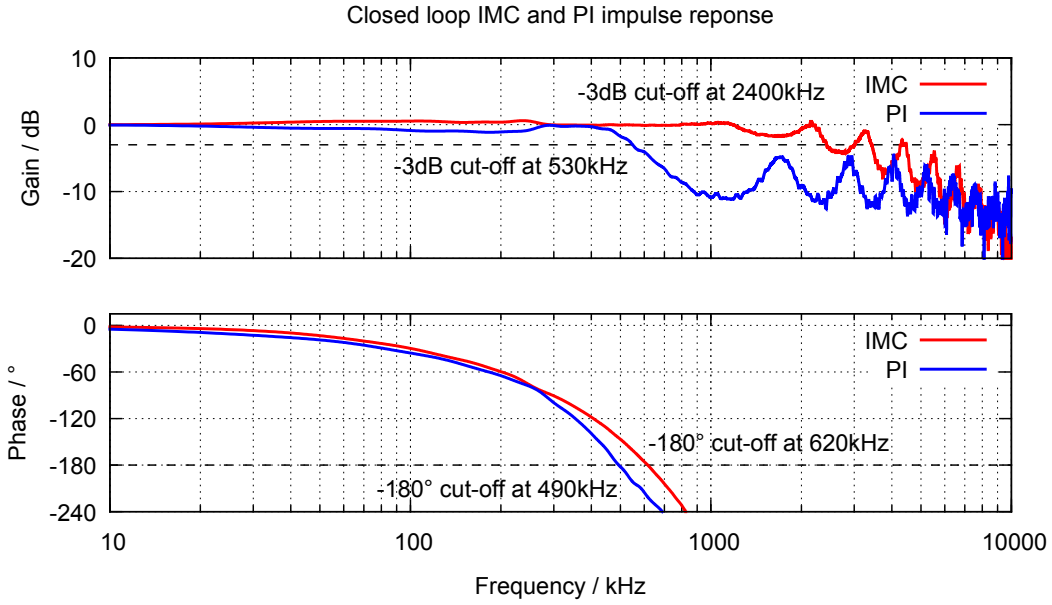


Figure 5.6.: *Bode plot of the closed loop impulse response from the step responses shown in fig. 5.5.* Gain stays close to unity (0 dB) up to the lowpass-like cut-off at 530 kHz for PI and at 2400 kHz for IMC. The phase decreases similarly for both control schemes while reaching $-180°$ at 620 kHz for IMC as opposed to 490 kHz for PI.

Possible ways to improve over this proof of concept test will be outlined in the next section among other ideas.

6. Conclusion and outlook

In this work I have presented
The digital controller based on the Red Pitaya compared to the Vescent laser servo can be used as an example and help to guide the selection of a digital device more suitable to reach or even surpass the performance of the Vescent laser servo. Special attention has to be paid to the converters and the board they are placed on to achieve low noise and good resolution. High resolution ADCs and DACs with 20 bit nominal are available for sampling rates of up to 1 Msps [54], [55] . A resolution of 20 bit corresponds to a relative resolution of 10^{-6}. This is sufficient for all intended applications (tab. 4.2). If the sampling time is 1 μs for both devices the delay amounts to at least 2 μs limiting the achievable control bandwidth to approx. 250 kHz (eq. 4.12). This is fine for most relevant applications.

For high bandwidth applications converters with 16 bit corresponding to $1.5 \cdot 10^{-5}$ relative resolution with lower delay times of 100 ns (ADC) and 20 ns (DAC) can be used [56], [57] . Provided computation time delay added by the FPGA can be kept under 20 ns such a configuration would provide a control bandwidth of >3 MHz (closer to the one of the analogue reference). Adding an analogue low pass filter after the DAC, excess bandwidth can be sacrificed to reliably reach the nominal resolution by oversampling and noise shaping [25]. This master thesis can serve as guideline how to characterise such ADCs and DACs, which is a crucial part of evaluating if a digital controller is fit for a given task.

To then harness the possibilities of an FPGA based controller a pure FPGA system should be considered removing the overhead of programming for a heterogeneous system which in return does not lead to any advantage for our intended applications. It is also possible to use software tools for designing digital filters and converting the design to HDL code (e.g. MATLAB). Preprogrammed and optimised modules provided by the FPGA manufacturer (that are usually well documented) can implement utility functions (such as user interface etc.) and may help avoiding some technical difficulties encountered during this project (see 5.4).

Using these tools one the presented IMC scheme could be improved by implementing a better frequency space model of the AOM and applying a digital band-stop filter eliminating unwanted resonances. Since such resonances often limit the bandwidth when using piezo elements (e.g. cavity length stabilisation) they too might benefit from IMC. In addition automated linearisation or gain scheduling can help with nonlinear systems. Further automation like automated search of and lock to a dispersive signal as used in many control loops is possible.

In the long term artificial intelligence could be used to supervise and adapt control loops as conditions change as well as provide data logging and remote analysis of the controlled system. This would ensure optimal conditions for experiment and researchers.

A. Appendix

A.1. Performance necessary for typical quantum optics control applications

This section presents some rough estimates leading to numbers shown in tab. 4.2 [47]:

- Laser frequency via spectroscopy:
  - Output resolution, typical linewidth of <100 kHz divided by typical scan range of 10 GHz
  - Bandwidth, mechanical resonances in a piezo actuator system with high mass load
- Laser phase in an optical phase-locked loop [58]:
  - Bandwidth, modulation bandwidth for laser diode current
- High finesse cavity length [59]:
  - Output resolution, inverse finesse, since Fabry-Perot resonance frequency $propto$ length, finesse = free spectral range divided by width of resonance
  - Bandwidth, piezo actuator system with low mass load
- Temperature of a laser [60]:
  - Input resolution, resolution of a high precision temperature sensor divided by typical temperature range of 10 K
  - Bandwidth, time scale at which typical devices change temperature (1 s)
- Intensity, dipole trap:
  - Bandwidth, limited by the dead time of the AOM (typ. <1 μs)

Bibliography

[1] J. Bechhoefer, "Feedback for physicists: A tutorial essay on control," *Rev. Mod. Phys.*, vol. 77, no. 3, pp. 783–833, 2005.

[2] T. Purdy, D. Brooks, T. Botter, N. Brahms, Z. Ma, and D. Stamper-Kurn, "Tunable cavity optomechanics with ultracold atoms," *Phys. Rev. Lett.*, vol. 105, p. 133602, 2010.

[3] N. Brahms, T. Botter, S. Schreppler, D. Brooks, and D. Stamper-Kurn, "Optical detection of the quantization of collective atomic motion," *Phys. Rev. Lett.*, vol. 108, p. 133601, 2012.

[4] A. Derevianko and H. Katori, "Physics of optical lattice clocks," *Rev. Mod. Phys.*, vol. 83, pp. 331–347, 2011.

[5] H. J. Kimble, "The quantum internet," *Nature*, vol. 453, pp. 1023–1030, 2011.

[6] S. Ritter, C. Nölleke, C. Hahn, A. Reiserer, A. Neuzner, M. Uphoff, M. Mücke, E. Figueroa, J. Bochmann, and G. Rempe, "An elementary quantum network of single atoms in optical cavities," *Nature*, vol. 484, pp. 195–200, 2012.

[7] I. Bloch, J. Dalibard, and S. Nascimbene, "Quantum simulations with ultracold quantum gases," *Nat. Phys.*, vol. 8, pp. 267–276, 2012.

[8] N. Rafee, T. Chen, and O. P. Malik, "A technique for optimal digital redesign of analog controllers," *IEEE transactions on control systems technology*, vol. 5, no. 1, 1997.

[9] K. J. Åström and B. Wittenmark, *Adaptive Control*. Courier Corporation, 2 ed., 2013.

[10] P. Horowitz and W. Hill, *The Art of Electronics*, p. 471. Cambridge University Press, 2 ed., 1989.

[11] A. O'Dwyer, "Tuning rules for pi and pid control of time delayed processes: some recent developments," *Proceedings of the Irish Signals and Systems Conference*, pp. 463–468, 1997.

[12] G. James, *Advanced Modern Engineering Mathematics*, p. 364. Pearson Education, 2004.

[13] S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, ch. 32. California Technical Publishing, 1997.

[14] U. Bakshi and V. Bakshi, *Control System Engineering*, ch. 11. Technical Publications, 2008.

[15] K. R. Rajeswari and B. V. Rao, *Signals and Systems*, p. 156. PHI Learning Private, 2009.

[16] W. S. Levine, *The Control Handbook*, p. 88. CRC Press, 1996.

[17] C. W. de Silva, *Modeling and Control of Engineering Systems*, p. 207. CRC Press, 2009.

[18] A. P. Prudnikov and O. I. Marichev, *Integrals and Series: Direct Laplace transforms*, p. 1. CRC Press, 1992.

[19] D. E. Rivera, "Internal model control: A comprehensive view," tech. rep., Arizona State University, 1999.

[20] M. T. Tham, "Internal model control." Lecture notes on Introduction to Robust Control, 2002.

[21] B. G. Liptak, *Instrument Engineers' Handbook*, vol. 2, p. 300. CRC Press, 4 ed., 2005.

[22] W. S. Levine, *Control System Fundamentals*, p. 170. CRC Press, 1999.

[23] W. S. Levine, *The Control Handbook*, p. 184. CRC Press, 1996.

[24] W. S. Levine, *The Control Handbook*, p. 393. CRC Press, 1996.

[25] W. Kester, "Adc input noise: The good, the bad, and the ugly. is no noise good noise?," *Analog Dialogue*, vol. 40, no. 1, 2006.

[26] W. Kester, "Mt-001," tech. rep., Analog Devices Inc., 2009.

[27] "Discrete-time fourier transform." `http://en.wikipedia.org/wiki/Discrete-time_Fourier_transform`. Wikipedia article accessed 13.5.2015.

[28] B. K, "Spectral leakage from a sinusoid and rectangular window." `http://commons.wikimedia.org/wiki/File%3ASpectral_leakage_from_a_sinusoid_and_rectangular_window.png`, 2005.

[29] F. J. Harris, "On the use of windows for harmonic analysis with the discrete fourier transform," *Proc. IEEE*, vol. 66, pp. 51–83, 1978.

[30] "Avr221: Discrete pid controller." Atmel Corp., 2006.

[31] D. Rowell, "2.161 signal processing: Continuous and discrete." MIT OpenCourseWare, 2008. Lecture 13.

[32] E. C. Ifeachor and B. W. Jervis, *Digital Signal Processing: A Practical Approach*, p. 321. Prentice Hall, 2002.

[33] D. Rowell, "2.161 signal processing: Continuous and discrete." MIT OpenCourseWare, 2008. Lecture 19.

[34] I. Kuon and J. Rose, "Measuring the gap between fpgas and asics," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 26, no. 2, pp. 203–215, 2007.

[35] A. Fratta, G. Griffero, and S. Nieddu, "Comparative analysis among dsp and fpga-based control capabilities in pwm power converters," in *Industrial Electronics Society, 2004. IECON 2004. 30th Annual Conference of IEEE*, vol. 1, pp. 257–262, 2004.

[36] "Ieee standard for verilog hardware description language," *IEEE Std 1364-2005 (Revision of IEEE Std 1364-2001)*, 2006.

[37] G. Bonanome, "Hardware description languages compared: Verilog and systemc," tech. rep., Columbia University, 2001.

[38] "Red pitaya hardware specifications v1.1.1." Red Pitaya, 2014.

[39] "Zynq-7000 all programmable soc overview." Xilinx, Inc., 2014.

[40] "Ltc2145cup-14 adc data-sheet." Linear Technology Corp., 2011.

[41] "Dac1401d125 rev. 03 dac data-sheet." Integrated Device Technology, Inc., 2012.

[42] "Hp 3589a operator's guide." Hewlett Packard, 1991.

[43] "Better noise measurements with the 588a and 3589a." Hewlett Packard, 1991. Application Note 1213.

[44] "Signal sources 2030 series manual." Aeroflex, Inc., 2004.

[45] "Zfl-500hln+ rev. b data-sheet." Mini-Circuits, 2002.

[46] Stanford Research Systems, 1290-D Reamwood Avenue, Sunnyvale, CA 94089 U.S.A., *Model SR560 low-noise preamplifier*, 2013.

[47] W. Alt. personal communication.

[48] "Product specifications." Axcel Photonics, Inc. Model M9-852-0150-S3P.

[49] "Mt80-a1.5-ir data sheet." AA opto-electronic, 2006.

[50] "Pda10a(-ec) si amplified fixed detector user guide." Thorlabs GmbH, 2015.

[51] R. Paschotta, *Encyclopedia of Laser Physics and Technology*, ch. article on 'acousto-optic modulators'. Wiley-VCH, 1 ed., 2008.

[52] Vescent Photonics, Inc., 4865 E. 41st Ave, Denver, CO 80216, *Reconfigurable Laser Servo Model No. D2-125*, 2011.

[53] R. Paschotta, *Encyclopedia of Laser Physics and Technology*, ch. article on 'shot noise'. Wiley-VCH, 1 ed., 2008.

[54] "Ltc2378-20 adc data-sheet." Linear Technology Corp., 2013.

[55] “Ad5791 dac data-sheet.” Analog Devices, Inc., 2013.

[56] “Ad7626 adc data-sheet.” Analog Devices, Inc., 2012.

[57] “Ad9142a dac data-sheet.” Analog Devices, Inc., 2014.

[58] G. Santarelli, A. Clairon, S. Lea, and G. Tino, “Heterodyne optical phase-locking of extended-cavity semiconductor lasers at 9 {GHz},” *Optics Communications*, vol. 104, no. 4–6, pp. 339 – 344, 1994.

[59] R. Drever, J. Hall, F. Kowalski, J. Hough, G. Ford, A. Munley, and H. Ward, “Laser phase and frequency stabilization using an optical resonator,” *Applied Physics B*, vol. 31, no. 2, pp. 97–105, 1983.

[60] L. A. J. Vikram Savani, “Achieving millikelvin temperature stability,” tech. rep., ILX Lightwave Corp., 2013.

Acknowledgements

After an fascinating and exciting year full of study and experimentation I would like to thank following persons:

- Prof. Dr. Dieter Meschede for the opportunity to work on a subject so deeply technical.
- Prof. Dr. Martin Weitz for acting as second referee.
- Dr. Andrea Alberti and Dr. Wolfgang Alt for their guidance and advice during my experiments and feedback on this thesis.
- Dipl.-Phys. Carsten Robens and Dipl.-Phys. Stefan Brakhane for their feedback on my Colloquium talk and this thesis and practical advice for my work at the lab.
- B.Sc. Gautam Ramola for his feedback on the theoretical part of this thesis.
- Annelise von Rudloff-Miglo, Fien Latumahina and Dr. Dietmar Haubrich for their help with organisatorial matters.
- The whole research group Meschede for the pleasant working atmosphere and mutual support.

Declaration

I hereby declare that this thesis was formulated by myself and that no sources or tools other than those cited were used.

Bonn, 13.05.2015 _____