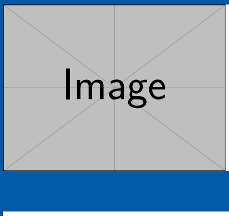
Development of Laser Systems and Spectroscopy of Highly Charged Ions

Zur Erlangung des Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.)
Vorgelegte Dissertation von Patrick Baus aus Mannheim
Tag der Einreichung: 4. November 2022, Tag der Prüfung: 4. November 2022

1. Gutachten: Prof. Dr. Gerhard Birkl
Foobar



Physics Department
Institut für Angewandte Physik
APQ

Development of Laser Systems and Spectroscopy of Highly Charged Ions

Submitted doctoral thesis by Patrick Baus

1. Review: Prof. Dr. Gerhard Birkl

Date of submission: November 4, 2022
Date of thesis defense: November 4, 2022

Foobar

1. Preparation

1.1. Grounding and Shielding

Add parts from ”references
Grounding and Shielding.pdf”

1.2. Laser Current Driver

1.2.1. Design

Simulation

Op Amp Stability

1.2.2. Noise Considerations

1.2.3. Voltage Reference

1.2.4. MOSFET Selection

1.3. LabKraken

1.3.1. Design Goals

LabKraken is a designed to be a asynchronous, resilient data aquisition suite, that scales to thousands of sensors and accross different networks.

1.3.2. Hardware

1.3.3. Software Architecture

LabKraken needs to scale to thousands of sensors, which need to be served concurrently. This problem is commonly referered to as the C10K problem as dubbed by Dan Kegel back in 1999 [16] and refers to serving 10 000 concurrent connections via network sockets. While today millions of concurrent connections can be handled by servers, handling 10 000 can still be challenging, especially, if the data sources are heterogeneous as is typical for sensor networks of different sensors from different manufacturers.

In order to meet the design goals, an asynchronous architecture was chosen and several different architectures were implemented over time. All in all four complete rewrites of the software were made to arrive at the architecture presented here. The reason for the rewrites is mostly historic and can be explained by the history of the programming language Python, which was used to write the code. The first first version was written for Python 2.6 and exclusively supported sensors made Tinkerforge. In 2015, Python 3.5 was released, which supported a new syntax for asynchronous coroutines. The software was rewritten from scratch to support this new syntax, because it made the code a lot more verbose and easier to follow. With the release of Python 3.7 in 2018 asynchronous generator expressions where mature enough to be used in productions and the programm was again rewritten to use the new syntax. In 2021 a new approach was taken and the programm was once more rewritten with a functional programming style. I will discuss each approach in the next sections to highlight the improvements, that were made over time. Each of these sections discusses the same programm, but written in different styles to show the differences.

Threaded Design

The first version of LabKraken used a threaded design approach, because the original libraries of the Tinkerforge sensors are built around threads. The following simplified example shows some code to connect to a temperature sensor over the network and read its data.

```
ipcon = IPConnection()
devices = dict()

# Callback function for temperature callback
def cb_temperature(temperature):
    print("Temperature: " + str(temperature/100.0) + " degC")

def cb_connected(connect_reason)::
    ipcon.enumerate()

def cb_disconnected(disconnect_reason)::
    log_reason(disconnect_reason)

def cb_enumerate(uid, *_args):
    if uid == OUR_KNOWN_DEVICE:
        dev = BrickletTemperatureV2(uid, ipcon)
        # Register temperature callback to function cb_temperature
        dev.register_callback(dev.CALLBACK_TEMPERATURE, cb_temperature)
        dev.set_temperature_callback_configuration(1000, False, "x", 0, 0)
        devives[uid] = dev

if __name__ == "__main__":
    ipcon.connect(HOST, PORT)  # blocking call
    # Register Enumerate Callback
    ipcon.register_callback(IPConnection.CALLBACK_ENUMERATE, cb_enumerate)
    ipcon.register_callback(IPConnection.CALLBACK_CALLBACK_CONNECTED,
                                             cb_connected)
```

Device Identifiers

Every sensor network needs device identifiers. Preferably those identifiers should be unique. Typically a device has some kind of internal indetifier. Here are a few examples of the sensors used in our network:

As it can be seen above, these identifiers do not guarantee to uniquely identify a device within a network. The Tinkerforge id is the weakest, as it is a 32 bit integer (4.294.967.295 options), which might easily collide with another id from a different manufacturer. The tinkerforge id

Device Type	Identifiers	Example
GPIB (SCPI)	**IDN?* returns $manufacturer,$name,$serial,$revision	
Tinkerforge	Each sensor has a base58 encoded integer device id	QE9 (163684)
Labnode	Universal Unique Identifier (UUID)	cc2f2159-e2fb-4ed9-8021-7771890b37ad

is presented as a base58 encoded string. An encoder/decoder example can be found in the TinkerforgeAsync library [5].

The id string returned by a SCPI device is slightly better, but again does not guarantee uniqueness. As it is shown in the example the same device might return a different id defpending on its settings. This typically done by manufacturers for compatibility reasons.

The only reasonably unique id is the universal unique identifier (UUID) or globally unique identifier (GUID), as dubbed by Microsoft, used in the Labnodes. Their id can be used for networks with participant numbers going into the millions.

Calculating the probability of a collision between two random UUIDs is called the birthday problem [30] in probability theory. A randomly generated version 4 UUID of variant 1 as defined in RFC 4122 [17] has 122 bit of entropy, that is out of 128 bit, 4 bit are reserved for the UUID version and 2 bit for the variant. This gives the probability of at least one collision in n devices out of $M = 2^{122}$ possibilities:

$$\begin{aligned} p(n) &= 1 - 1 \cdot \left(1 - \frac{1}{M}\right) \cdot \left(1 - \frac{2}{M}\right) \dots \left(1 - \frac{n-1}{M}\right) \\ &= 1 - \prod_{k=1}^{n-1} \left(1 - \frac{k}{M}\right) \end{aligned} \tag{1.1}$$

Using the Taylor series $e^x = 1 + x \dots$, assuming $n \ll M$ and approximating we can simplify this to:

$$\begin{aligned} p(n) &\approx 1 - \left(e^{\frac{-1}{M}} \cdot e^{\frac{-2}{M}} \dots e^{\frac{-(n-1)}{M}}\right) \\ &\approx 1 - \left(e^{\frac{-n(n-1)/2}{M}}\right) \\ &\approx 1 - \left(1 - \frac{n^2}{2M}\right) = \frac{n^2}{2M} \end{aligned} \tag{1.2}$$

For one million devices, this gives a probability of about 2×10^{-25}, which is negligible.

In the Kraken implementation, all devices, except for the Labnodes, will be mapped to UUIDs using the underlying configuration database. It is up to the user to ensure the uniqueness of the non-UUID ids reported by the devices to ensure proper mapping.

Limitations

There is one inherent limitation to the ethernet bus for instrumentation. The ethernet bus is inherently asynchronous and multiple controllers can talk to the device at the same time. Not only that, but different processes within the same controller can talk to the same device. This makes deterministic statements about the device state challenging.

While it is impossible to rule out the possibility of multiple controllers on a network, care was taken to synchronize the workers within Kraken.

1.3.4. Databases

Cardinality

- TimescaleDB vs Influx
- Example Sensors vs. Experiment

1.4. Short Introduction to Control Theory

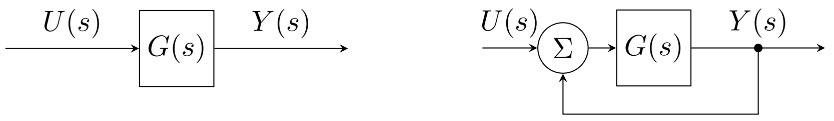
This section will give a very brief introduction into some basic concepts of control theory. Many systems require control over one or more process variables. For example, temperature control of a room or a device, or creating a current from a voltage. All of this requires control over a process and is established trough feedback, which allows a controller to sense the state of the system.

The focus of this section lies on the principels feedback and control and will be detailed in the following sections.

1.4.1. Transfer Functions

1.4.2. Open and Closed Loop Systems

To understand feedback, one needs to take a look at dynamical systems. There are two types of systems: open and closed loop systems. A system is called open loop, if the output of a system does not influece its input as in figure 1.1a. On the other hand, if the output is connected to the input of the system it is called closed loop system, an example is shown in figure 1.1b. $G(s)$ is called the transfer function of the system, while $R(s)$ is the input, $Y(s)$ is the output and s the Laplace variable.



(a) Open loop system.

(b) Closed loop system.

Figure 1.1.: Block diagram of closed and open loop systems.

It is convenient to express the transfer function as its Laplace transform. The unilateral Laplace transform is definded as:

$$\mathscr{L}\left(f(t)\right) = F(s) = \int_0^\infty f(t)e^{-st}\,dt. \tag{1.3}$$

with $f : \mathbb{R}^+ \to \mathbb{R}$, that is integrable and grows no faster than $e^{s_0 t}$ for $s_0 \in \mathbb{R}$. The latter property is important for deriving the rules of differentiation and integration.

To understand the benefits of using the Laplace representation for transfer function a few useful properties must be discussed. First of all the Laplace transform is linear:

$$\begin{aligned}\mathscr{L}\left(a \cdot f(t) + b \cdot g(t)\right) &= \int_0^\infty (a \cdot f(t) + b \cdot g(t))e^{-st}\,dt \\ &= a\int_0^\infty f(t)e^{-st}\,dt + b\int_0^\infty g(t)e^{-st}\,dt \\ &= a\mathscr{L}\left(f(t)\right) + b\mathscr{L}\left(g(t)\right)\end{aligned} \tag{1.4}$$

Another interesting property is the derivative and integral of a function f:

$$\begin{aligned}\mathscr{L}\left(\frac{df}{dt}\right) &= \int_0^\infty \underbrace{f'(t)}_{v'(t)}\,\underbrace{e^{-st}}_{u(t)}\,dt \\ &= \left[e^{-st}f(t)\right]_0^\infty - \int_0^\infty (-s)f'(t)\,dt \\ &= -f(0) + s\int_0^\infty f'(t)\,dt \\ &= sF(s) - f(0) \end{aligned} \tag{1.5}$$

$$\begin{aligned}\mathscr{L}\left(\int_0^t f(\tau)\,d\tau\right) &= \int_0^\infty \left(\int_0^t f(\tau)\,d\tau e^{-st}\right)\,dt \\ &= \int_0^\infty \underbrace{e^{-st}}_{v'(t)}\,\underbrace{\int_0^t f(t)\,d\tau}_{u(t)}\;dt \\ &= \left[\frac{-1}{s}e^{-st}\int_0^t f(t)\,d\tau\right]_0^\infty - \int_0^\infty \frac{-1}{s}e^{-s\tau}f(\tau)\,d\tau \\ &= 0 + \frac{1}{s}\int_0^\infty e^{-s\tau}f(\tau)\,d\tau \\ &= \frac{1}{s}F(s) \end{aligned} \tag{1.6}$$

If the initial state $f(0)$ can be chosen to be 0, the differentiation becomes a simple multiplication by s, while the integration becomes a division by s. Finally, the most important aspect is, that a simple relation between the input $r(t)$ and the ouput $y(t)$ of a system can be given. The relation between input and the ouput of a system as shown in figure 1.1a is given by the convolution, see e.g. [3]. Assuming the system has an initial state of 0 for $t < 0$, hence $r(t < 0) = 0$ and $g(t < 0) = 0$, one can calculate:

$$y(t) = (r * g)(t) = \int_0^\infty r(\tau)g(t-\tau)\,d\tau \tag{1.7}$$

Applying the Laplace transformation, greatly simplifies this:

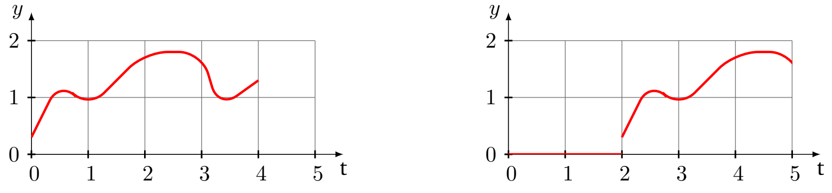
$$\begin{aligned} Y(s) &= \int_0^\infty e^{-st}y(t)\,dt \\ &\overset{1.7}{=} \int_0^\infty \underbrace{e^{-st}}_{e^{-s(t-\tau)}e^{-s\tau}} \int_0^\infty r(\tau)g(t-\tau)\,d\tau\,dt \\ &= \int_0^\infty \int_0^t e^{-s(t-\tau)}e^{-s\tau}g(t-\tau)r(\tau)\,d\tau\,dt \\ &= \int_0^\infty e^{-s\tau}r(\tau)\,d\tau \int_0^\infty e^{-st}g(t)\,dt \\ &= R(s)\cdot G(s) \end{aligned} \tag{1.8}$$

This formula is a lot simpler than the convolution of $r(t)$ and $g(t)$, therefore the use of the Laplace transform has become very popular in control theory.

Another property that is heavily used in control theory is the time delay of functions. To show this property, let $f(t-\theta)$ be

$$g(t) := \begin{cases} f(t-\theta), & t \geq \theta \\ 0, & t < \theta \end{cases} \tag{1.9}$$

The reason for this definition is, that the system must be causal. This means, it is impossible to get data from the future ($t < \theta$). An example is shown in figure 1.2a.



(a) Original signal $f(t)$.

(b) Delayed signal $f(t-2)$.

The Laplace transform of a delayed signal can be calculated as follows:

$$\begin{aligned} \mathscr{L}\left(g(t)\right) &= \int_0^\infty f(t-\theta)e^{-st}\,dt \\ &\overset{1.9}{=} \int_\theta^\infty f(t-\theta)e^{-st}\,dt \\ &\overset{u:=t-\theta}{=} \int_0^\infty f(u)e^{-s(u+\theta)}\,du \\ &= e^{-s\theta}\int_0^\infty f(u)e^{-su} \\ &= e^{-s\theta}F(s) \end{aligned} \tag{1.10}$$

To satisfy the causaulity requirement, the Heaviside function $H(t)$ can be used:

$$\mathscr{L}\left(f(t-\theta)H(t-\theta)\right) = e^{-s\theta}F(s) \tag{1.11}$$

Lastly, the Laplace transform of e^{at}, which is commonly used in differential equations:

$$\mathscr{L}\left(e^{at}\right) = \int_0^\infty e^{(a-s)t}\,dt = \frac{1}{a-s}\left[e^{(a-s)t}\right]_0^\infty = \frac{1}{s-a} \tag{1.12}$$

Using these tools, it is possible calculate the transfer function of a temperature controller. This is done in the next section.

1.4.3. A Model for Temperature Control

In order to describe a closed-loop system, one has to first create a model for the process and the controller involved. A simple model can be derived from the idea, that the system at temperature T_{system} has a thermal capacitance C_{system}, an influx of heat $\dot{Q}_{load}$ from a thermal load and a controller removing heat from the system through a heat exchanger with a resistance of R_{force}. Additionally, there is some leakage through the walls of the system to the ambient environment via $R_{leakage}$. This analogy of thermodynamics with electrondynamics allows to

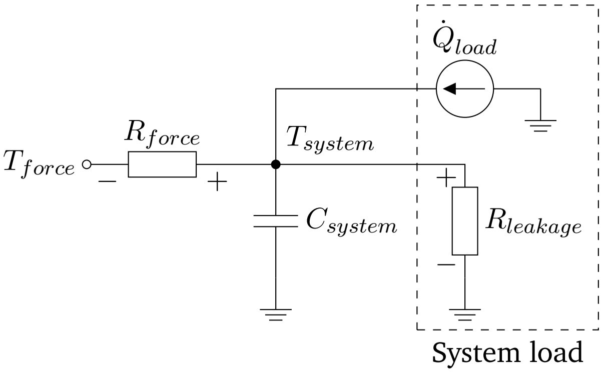
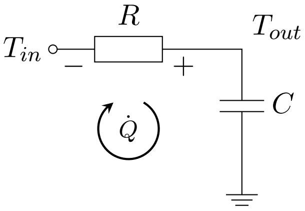


Figure 1.3.: Simple temperature model of a generic system.

create the model shown in figure 1.3. Since this this model is to be used for a room temperature controller, an assumption to simplify it can be made.

Typically, the room temperature is kept constant. Therefore, the controller will keep T_{system} constant and if the outside temperature and the heat load $\dot{Q}_{load}$ is *reasonably stable*, it is easy to see, that a constant thermal flux must flow through R since it cannot pass through the thermal capacitance C. *Reasonably stable* means that those fluxes can be treated as constant with respect to the temperature controller time constants. This will be further discussed in section **??** with regards to system stability. If this assumption holds, the thermal flux from the system load will only cause a constant offset of T_{force}, since the heat must be removed by the controller through the resistance R_{force}, and the model can be simplified further. Here T_{force} and T_{system} was replaced by T_{in} and T_{out} for better readability.:



This is the classic RC circuit. Now, neglecting the constant thermal flux from the system load and exploiting the analogy of thermodynamics and electrondynamics again, using Kirchhoff's second law, one finds:

$$\sum T_i = 0$$

$$T_{in}(t) - \dot{Q}(t)R - \frac{1}{C}\int \dot{Q}(t)\,dt = 0 \tag{1.13}$$

Taking the Laplace transform, applying equation 1.6 and using $T_{out} = \frac{1}{sC}\dot{Q}(s)$ to replace $\dot{Q}$, equation 1.13 can be written as:

$$T_{in}(s) - \dot{Q}(s)R - \frac{1}{sC}\dot{Q}(s) = 0$$

$$\dot{Q}(s) = \frac{T_{in}(s)}{R - \frac{1}{sC}} = \frac{T_{out}}{\frac{1}{sC}}$$

This allows to calculate the transfer function of the process P:

$$\begin{aligned} P(s) = \frac{T_{out}}{T_{in}} &= \frac{\frac{1}{sC}}{R - \frac{1}{sC}} \\ &= \frac{1}{sRC+1} \\ &= \frac{K}{1+s\tau} \end{aligned} \tag{1.14}$$

with the system gain K and the time constant τ. In case of the RC circuit, the gain is 1, but other systems may have a gain or attenuation factor of $K \neq 1$ in the sensor.

Equation 1.14 is called the transfer function of a first-order model, because its origin is a differential equation of first order. This model describes homogeneous systems, like a room, very well, as can be seen in section **??**, but in order to derive the transfer function including the controller and the sensor some more work is required on the sensor transfer function.

Expanding on figure 1.1b and equation 1.7 the closed-loop transfer function becomes:

$$G(s) = P(s) \cdot S(s) \tag{1.15}$$

and the block diagram becomes

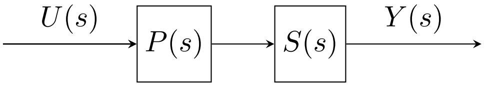


Figure 1.4.: Open loop system with sensor.

The transfer funciton of the sensor can be modeled as a delay line with delay θ and $f(t-\theta) = H(t-\theta)$. A gain of 1 is assumed here, because any system gain is already included in the parameter K. Using equation 1.10 $S(s)$ can be written as

$$S(s) = e^{-\theta s}. \tag{1.16}$$

The full process model including the time delay is:

$$G(s) = \frac{K}{1+s\tau} e^{-\theta s} \tag{1.17}$$

This is called a first-order plus dead-time model (FOPDT) or first-order plus time-delay model (FOPTD). To fit experimental data to this model it is more convenient to transform the transfer function 1.17 into the time domain. To calculate the output response an input $U(s)$ is required. In principal any function can do, but a step function is typically used, for example by Ziegler and Nichols [33] and many others [19, 22, 29, 27, 25, 26, 3]. It is both simple to calculate and apply to a real system. Using equations 1.10 and 1.12, the Heaviside $H(t)$ step function transforms as

$$\mathscr{L}\left(u(t)\right) = U(s) = \mathscr{L}\left(\Delta u H(t)\right) = \frac{\Delta u}{s} \tag{1.18}$$

with the step size Δu. The output $Y(s)$ can then be calculated analytically.

$$
\begin{aligned}
Y(s) &= \frac{\Delta u}{s}\frac{K}{1+s\tau}e^{-\theta s} \\
&= K\Delta u\frac{1}{s(1+s\tau)}e^{-\theta s} \\
&= K\Delta u\left(\frac{1}{s}-\frac{\tau}{s\tau+1}\right)e^{-\theta s} \\
&= K\Delta u\left(\frac{1}{s}-\frac{1}{s+\frac{1}{\tau}}\right)e^{-\theta s}
\end{aligned} \tag{1.19}
$$

To derive $y(t)$, the inverse Laplace transform of $Y(s)$ is required. Unfortunately, this is not as simple as the Laplace transform. Fortunately, using 1.12 while making sure causaulity is guaranteed as shown in 1.11, the simple first order model can easily be transformed back into the time domain.

$$
\begin{aligned}
\mathscr{L}^{-1}\left(Y(s)\right) = y(t) &= K\Delta u\mathscr{L}^{-1}\left(\frac{1}{s}e^{-\theta s}\right) - K\mathscr{L}^{-1}\left(\frac{1}{s+\frac{1}{\tau}}e^{-\theta s}\right) \\
&\overset{1.12}{=} K\Delta u \cdot 1 \cdot H(t-\theta) - \left(e^{-\frac{t-\theta}{\tau}}\right)H(t-\theta) \\
&= K\Delta u\left(1-e^{-\frac{t-\theta}{\tau}}\right)H(t-\theta)
\end{aligned} \tag{1.20}
$$

The time domain solution of the FOPDT model can now be used extract the parameters τ, θ and K from a real physical system using a fit to the measurement data. The parameter Δu is already known, since it is an input parameter. A simulation of the step response of a first-order model with time delay is shown in figure 1.5. Here it can be clearly seen, that the output does not change until the time delay θ has passed and the Heaviside function changes from 0 to 1.

1.4.4. PID tuning rules

We use $\tau_c = \tau$ as suggested by [29, 27] for “*tightest possible subject to maintaining smooth control*“.

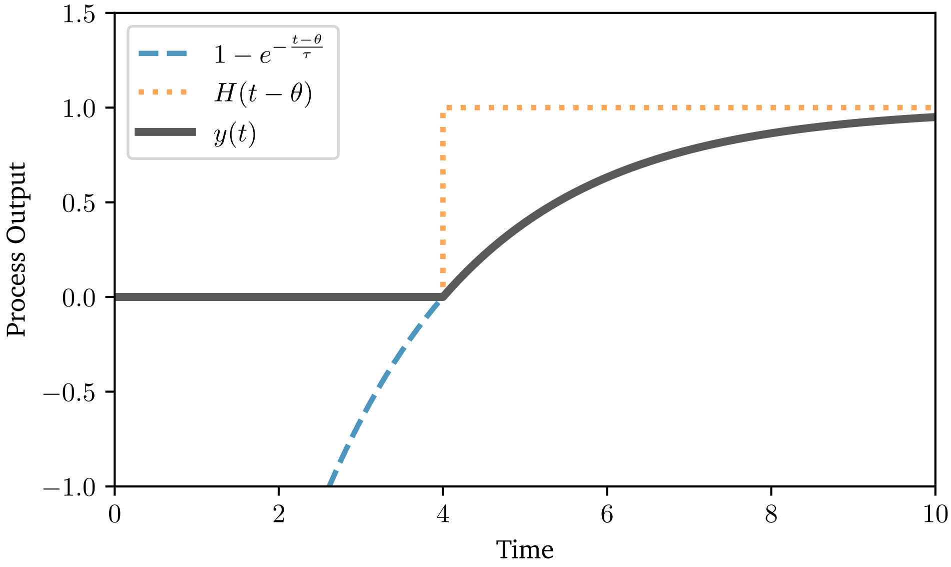


Figure 1.5.: Time domain plot of a first-order plus dead time model, showing induvidual components of the model and the composite function $y(t)$. Model parameters: $K = \Delta u = 1$, $\tau = 2$, $\theta = 4$.

1.5. Allan Deviation

The Allan variance [1] $\sigma_A^2(\tau)$ is a two-sample variance and used as a measure of stability. The Allan deviation $\sigma_A(\tau)$ is the square root of the variance. Originally, the Allan variance was used to quantify the performance of oscillators, namely the frequency stability, but it can be used evaluate any quantity. In order to define the Allan variance, a few terms need to be defined first. A single measurement value of the time series $y(t)$ can be written as

$$\bar{y}_k(t) = \frac{1}{\tau} \int_{t_k}^{t_k+\tau} y(t)\, dt. \tag{1.21}$$

This is the k-th measurement with a measurement time or integration time τ. The latter term is frequently used for DMMs. t_k is the sampling inverval including the dead time θ

$$t_{k+1} = t_k + T \tag{1.22}$$

with

$$T = \tau + \theta. \tag{1.23}$$

Using this, the standard deviation over N sampled is defined as [1, 4]

$$\sigma_A^2(N, T, \tau) = \left\langle \frac{1}{N-1} \left(\sum_{n=0}^{N-1} \bar{y}_n^2(t) - \frac{1}{N} \left(\sum_{n=0}^{N-1} \bar{y}_n(t) \right)^2 \right) \right\rangle \tag{1.24}$$

The $\langle\ \rangle$ denotes the (infinite time) average over all measurands y_k. Hence for all k.

The Allan variance is a special case of this definition with zero dead-time ($\theta = 0$) and only 2 samples:

$$\sigma_A^2(\tau) = \sigma_A^2(N = 2, T = \tau, \tau) \tag{1.25}$$

$$= \left\langle \frac{\left(\bar{y}_{k+1} - \bar{y}_k\right)^2}{2} \right\rangle \tag{1.26}$$

In practice, no experiment can take an infinite number of samples, so typically the Allan variance is estimated using a number of samples m:

$$\sigma_A^2(\tau) \approx \frac{1}{m} \sum_{k=1}^{m} \frac{\left(\bar{y}_{k+1} - \bar{y}_k\right)^2}{2} \tag{1.27}$$

It can be shown [4], that 1.27 is indeed more usefull than $\sigma_A^2(N \to \infty, T = \tau, \tau)$, becuase $\sigma_A^2(\tau)$ even for $m \to \infty$ converges for processes, that do not have a convergent $\sigma_A^2(N \to \infty, T = \tau, \tau)$.

Additionally, the Allan variance is mathematically related to the two-sided power spectral density $S_y(f)$ [4]:

$$\sigma_A^2(\tau) = 2 \int_0^\infty S_y(f) \frac{\sin^4(\pi f t)}{(\pi f \tau)^2}\, df \tag{1.28}$$

and therefore all processes, that can be seen in the power spectral density can also be seen in the allan deviation. The inverse transform, however, is not always possible as shown by Greenhall [10].

Distinguishing different noise processes using the Allan deviation will be elaborated in the next section.

1.5.1. Identifying Noise in Allan Deviation Plots

It was already mentioned by Allan in [1], that types of noise, whose spectral density follows a power law

$$S(f) = C \cdot f^{\alpha} \tag{1.29}$$

can be easily identified in the Allan deviation plot. The most common types of noise encountered in experimental data and their power coefficients can be found in table 1.1. Since those types of noise is present in any measurement or electronic device, it warants a further discussion to understand their root causes and ideas to minimize them. While not a type of noise, linear drift can also be easily identified in the Allan deviation plot. It is therefore included in table 1.1 as well.

Amplitude noise type	Power-law coefficient α	Allan deviation σ_A
White noise	0	$\propto \tau^{-1/2}$ [2]
Flicker noise	-1	$\propto \tau^{0}$ [2]
Random walk noise	-2	$\propto \tau^{1/2}$ [2]
Burst noise	0 and -2	$\propto \tau^{1/2}$ and $\tau^{-1/2}$
Drift	–	$\propto \tau^{1}$ [11]

Table 1.1.: Power law representations using the Allan variance.

In order to arrive at a good understanding of the features seen in an Allan deviation plot, this section will provide the reader with examples of each type of noise and the corresponding time domain, power spectral density and Allan deviation plot. Since a complete overview is not available in current literature, all required mathematical descriptions and simulation tools will be discussed here. The simulations were done using Python and the source code is linked to in the discussions.

White Noise

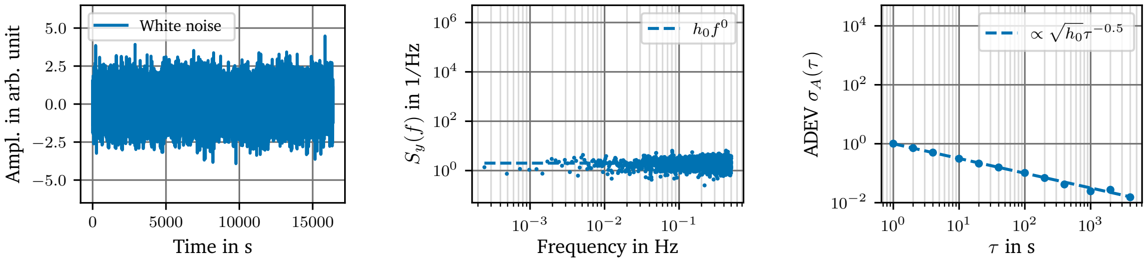
White noise is probably the most common type of noise found in measurement data. Johnson noise found in resistors, caused by the random fluctuation of the charge carriers, is one example of mostly white noise up to bandwidth of 100 MHz, from where on quantum corrections are required [9]. Amplifiers also tend to have a white noise spectrum at higher frequencies. For these reasons, it typically makes up for a considerabe amount of noise in a measurement, unless one measures at very low frequencies. White noise is a series of uncorrelated random events and therefore characterised by a uniform power spectral density, which means there is the same power in a given bandwidth at all frequencies. Another one of its important and often used properties is, that the variance of two uncorrelated variables adds:

$$\sigma_{x+y}^2 = \sigma_x^2 + \sigma_y^2 + \underbrace{2\,\mathrm{Cov}(x,y)}_{\text{uncorrelated}=0} = \sigma_x^2 + \sigma_y^2 \tag{1.30}$$

This allows simple addition rules of variances from different sources, but it must be stressed here, that this property is only valid for uncorrelated sources like white noise, although it is usually incorrectly applied to all measurements in disreagard of the dominant noise present, which ufortunately obscures rather than clarifies the uncertainties involved.

In order to demonstrate the effect of white noise in Allan deviation plots, it was simulated using the excellent *AllanTools* library [31]. The noise generator chosen in the AllanTools library is based on the work of Kasdin and Walter [13]. The full Python program code is published online []. For better comparison, all noise densities are normalized to give an Allan deviation of $\sigma_A(\tau_0) = 1$, with τ_0 being the smallest time interval between measurements.

Figure 1.6 shows a sample of white noise in three different forms. Figure 1.6a is the time series representation. From this sample, the power spectral density was calculated and is shown in figure 1.6b. The dashed line shows the expectation value of the power spectral density and the Allan deviation.



(a) Time domain

(b) Power spectral density

(c) Allan deviation

Figure 1.6.: Different representations of white noise.

From this simulation, several features can be observed. First of all, the power spectral density is flat and constant with $h_0 = 2$, which is in accordance with table 1.1 and the normalization mentioned earlier. Figure 1.6c shows the typical $\tau^{-\frac{1}{2}}$ dependence of white noise in the Allan deviation plot. This immediatly explains, why filtering white noise scales with $\frac{1}{\sqrt{n}}$ with n being the number of samples averaged.

Burst Noise

Burst noise, popcorn noise, or sometimes referred to as random telegraph signal is a random bi-stable change in a signal and is caused by a generation recombination processes. This, for example, happens in semiconductors if there is a site, that can trap an electrons for a prologned period of time and then randomly release it. Impurities causing lattice defects are discussed in this context [15, 24, 6, 14]. Such latttice defects can also be introduced by ion implantation during doping. Fortunately, this type of noise has become less prevalent in modern manufacturing processes, because the quality of the semiconductors has improved. But if a trap site is located very close to an important structure, for example a high precision Zener diode, its effect might be so strong, that it can be clearly seen.

The discussion is split into two parts. First the power spectral density is calculated and then the Allan variance is caclulated using that result.

The spectral density of burst noise caused by a single trap site was derived in [18] by Machlup. The author used the autocorrelation function of the burst noise signal and applied the Wiener-Khinchin (Wiener-Хинчин) theorem, which connects the autocorrelation function with the power spectral density. A more detailed derivation can be found in [32], in this paper the preconditions, like stationarity of the process, are also discussed. The burst noise signal consists of two energie levels, called 0 and 1, split by Δy. Multiple burst noise signals can be superimposed in a real device. This would then result in mutiple levels, but they can be treated separately. The measurement interval over an even number of transitions, so that one ends in the same state as the measurement has started, is the time T. The mean lifetime of the levels is called $\bar{\tau}_0$ and $\bar{\tau}_1$:

$$\bar{\tau}_0 \approx \frac{1}{N}\sum_i^N \tau_{0,i} \qquad \bar{\tau}_1 \approx \frac{1}{N}\sum_i^N \tau_{1,i} \tag{1.31}$$

Figure 1.7 shows a burst noise signal along with the definitions above.

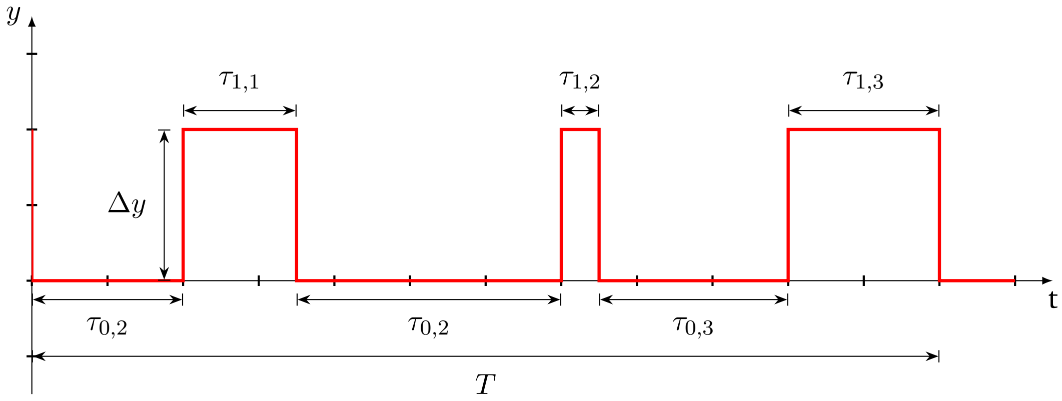


Figure 1.7.: A random burst noise signal.

Using these definitions, one can then derive [18]:

$$R_{xx}(T) = \Delta y^2 \cdot \frac{\bar{\tau}_1 \bar{\tau}_0 e^{-\left(\frac{1}{\bar{\tau}_1}+\frac{1}{\bar{\tau}_0}\right)T}}{(\bar{\tau}_1+\bar{\tau}_0)^2} \quad \text{and} \tag{1.32}$$

$$S(\omega) = 4R_{xx}(0)\frac{\frac{1}{\bar{\tau}_1}+\frac{1}{\bar{\tau}_0}}{\left(\frac{1}{\bar{\tau}_1}+\frac{1}{\bar{\tau}_0}\right)^2+\omega^2} \qquad \omega > 0. \tag{1.33}$$

Note, that the power spectral density is the one-sided version, hence an additional factor of 2 is included. The d.c. term was ommitted here and can usually be neglected, because it is not relevant for calculating the power spectral density as it only contributes a single peak at $\omega = 0$. Using the following definitions of the average time constant and the duty cycle

$$\frac{1}{\bar{\tau}} = \frac{1}{\bar{\tau}_1}+\frac{1}{\bar{\tau}_0} \quad \text{and} \tag{1.34}$$

$$D_i = \frac{\bar{\tau}_i}{\bar{\tau}_1+\bar{\tau}_0} \qquad i \in \{0;1\} \tag{1.35}$$

equations 1.32 and 1.33 can be rewritten to give a more intuitive form:

$$R_{xx}(T) = \Delta y^2 D_1 D_0\, e^{-\left(\frac{1}{\bar{\tau}_1}+\frac{1}{\bar{\tau}_0}\right)T} \tag{1.36}$$

$$S(\omega) = 4R_{xx}(0)\frac{\bar{\tau}}{1+\omega^2\bar{\tau}^2} \tag{1.37}$$

The special case $\bar{\tau}_0 = \bar{\tau}_1$ with $D = \frac{1}{2}$ is the previously mentioned case of random telegraph noise.

$R_{xx}(0)$ can be identified as the mean squared value of y:

$$y_{RMS} = \sqrt{R_{xx}(0)}\,. \tag{1.38}$$

Equation 1.37 is a Lorentzian function and from this it can be easily seen, that a single trap site has a power spectral density, which is proportional to $\frac{1}{f^2}$ at high frequencies and is flat at low frequencies.

With the spectral density in hand, it is now possible to calculate the Allan variance as it was done by Van Vliet and Handel in [28] for the classic example of random telegraph noise where $\bar{\tau}_1 = \bar{\tau}_0$. Do note, that table I given by Van Vliet and Handel shows the total number of events instead of the instantationous number of events typically given. Hence, their notation must be multiplied by $\frac{1}{\bar{\tau}^2}$. For the generic case with $\bar{\tau}_1$, $\bar{\tau}_0$ and the definition of $\bar{\tau}$ given in equation 1.34 one finds for the Allan variance of burst noise:

$$\sigma_A^2(\tau) = R_{xx}(0)\frac{\bar{\tau}^2}{\tau^2}\left(4e^{-\frac{\tau}{\bar{\tau}}} - e^{-\frac{2\tau}{\bar{\tau}}} + 2\frac{\tau}{\bar{\tau}} - 3\right) \tag{1.39}$$

Having arrived at equations 1.37 and 1.39 of the power spectral density and Allan variance, it it now possible to model it. For this purpose, parts of the Python library *qtt* [8] was used. The algorithm written by Eendebak et al. implements continous-time Markov chains to simulate the burst noise signal. The result can be see in figure 1.8. For these simulations one time constants, namely the lifetime of the lower state $\bar{\tau}_0$ was held constant, while the lifetime of the upper state was varied to show the effect of different $\bar{\tau}$. By looking at the time domain in figure 1.8a it can be seen, that the maximum average number of state changes can be observed, when $\bar{\tau}_1 = \bar{\tau}_0$. If $\bar{\tau}_1 > \bar{\tau}_0$ the system will favour the upper, while if $\bar{\tau}_1 < \bar{\tau}_0$ it will favour the lower state instead. This explaines why the noise is strongest for random telegraph noise when $\bar{\tau}_1 = \bar{\tau}_0$, which can also be seen in power spectral density in figure 1.8b. Looking at the Allan deviation in figure 1.8c confirms this, but also shows another interesting implication as it shows an obvious maximum. If the application allows a choice over the sampling interval τ, the effect of the burst noise can mitigated by staying well clear of the maximum.

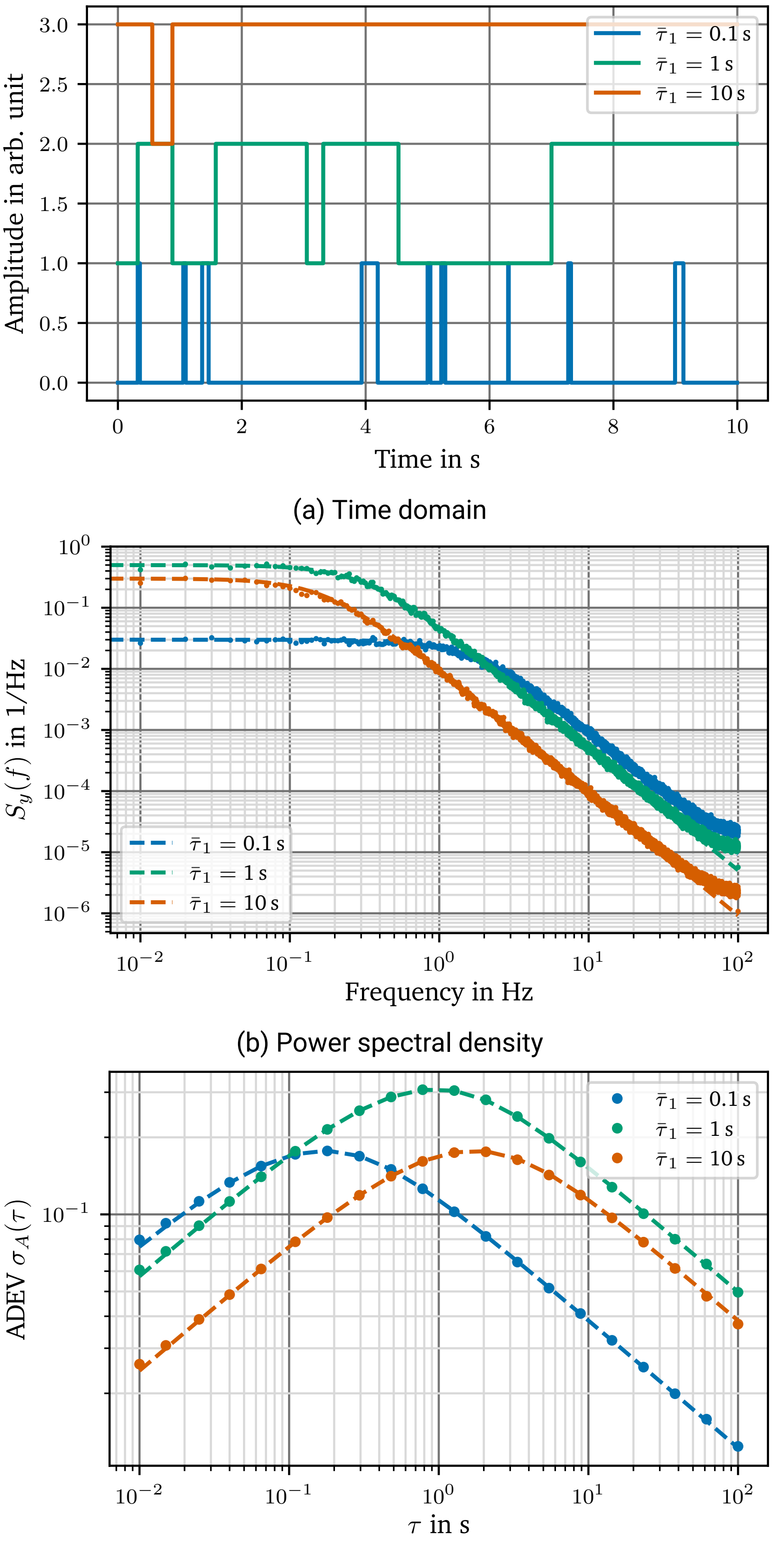
The small deviation from the analytical solution in figure 1.8c at large τ is a typical so called end-of-data error. As it was discussed above, the Allan deviation can only be estimated given a limited number of samples using equation 1.27 and going to longer τ means there are fewer samples to average over.

The burst noise equations can used to gain further insight into more types of noise. The first one is Shot noise, which is commonly found in photodetectors and lasers. Here, electrons or photons are created at discrete intervals resulting in an instantationous signal. This means, that the lifetime of the upper level is very short in comparison to the lower level ($\tau_1 \ll \tau_0$). Using a Taylor series, equation 1.33 then becomes:

$$\begin{aligned} S_{Shot}(\omega) = S_{\tau_1 \ll \tau_0}(\omega) &\approx 4\Delta y^2 \frac{\tau_1}{\tau_0} \frac{\frac{1}{\bar{\tau}_1}}{1 + \omega^2 \left(\frac{1}{\bar{\tau}_1}\right)^2} \\ &= 4\Delta y^2 \frac{\tau_1^2}{\tau_0} \frac{1}{1 + \omega^2 \tau_1^2} \qquad (1.40) \\ &\overset{\omega \ll 1/\tau_0}{\approx} 4\Delta y^2 \frac{\tau_1^2}{\tau_0} \qquad (1.41) \end{aligned}$$

For the typical case, $\omega \ll 1/\tau_0$, this results in a white spectrum as observed in photodetectors and lasers.

The other interesting case is, if many trap sites with different time constants are contributing to the noise. This case is discussed in the next section.



(a) Time domain

(b) Power spectral density

(c) Allan deviation

Figure 1.8.: Different representations of burst noise for different $\bar{\tau}_1$ and fixed $\bar{\tau}_0 = 1\,\mathrm{s}$.

Flicker Noise

Flicker noise is also called $\frac{1}{f}$-noise and it can be observed in many naturally occuring phenomenen and its origin is not clear, although there have been many explanations. An overview can be found in [20, 23]. This work concentrates on flicker noise in electronic devices. For example in thick-film resistors it was shown to extend over at least 6 decades without any visible flattening [21]. One explanation for flicker noise in transistors is the existance of generation-recombination noise or burst noise discussed in the previous section. If there are many uncorrelated trap sites, that contribute to the total noise, the envelope of the noise spectral density changes from $\frac{1}{f^2}$ to $\frac{1}{f^1}$. This is shown in figure 1.9

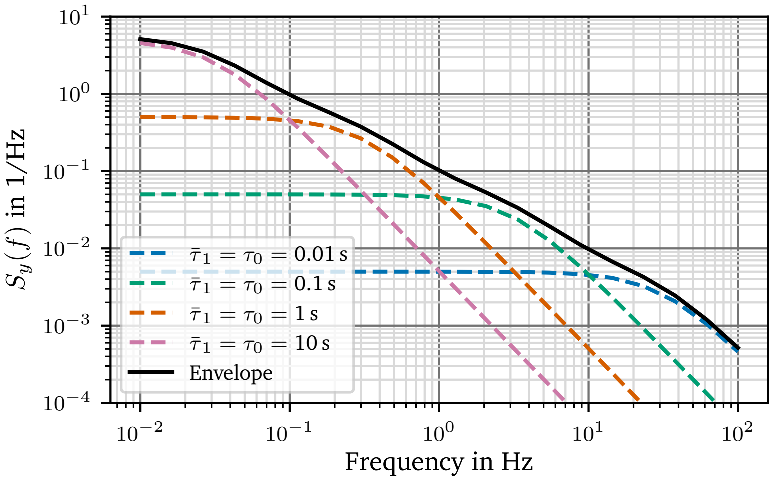


Figure 1.9.: Multiple overlapping Lorentzian noise sources forming a $\frac{1}{f}$-like shape.

Mathematically this can be motivated using the assumption that the number of trap sites N with a certain time constant τ is inversely proportional to the time constant:

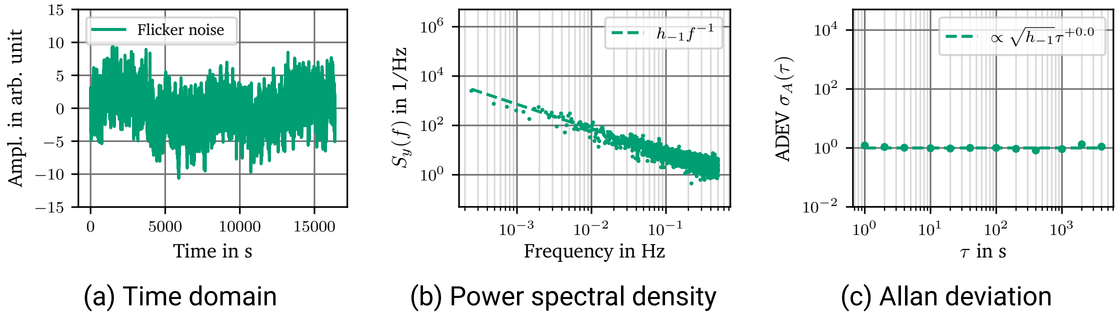
$$N(\tau) \propto \frac{1}{\tau} \tag{1.42}$$

This assumption is physical, because it is unreasonable, that a trap site can store an electron indefinitely and also it is reasonable that there is little correlation as trap sites spread over a large area in a semiconductor do not influence each other. Using equation 1.37 from the previous section, multiplying the weight function 1.42 and integrating over all possible storage times gives:

$$\begin{aligned} S(\omega) &= \lim_{t\to\infty} \int_0^t N(\tau)\, 4R_{xx}(0) \frac{\tau}{1+\omega^2\tau^2}\, d\tau \\ &= 4R_{xx}(0)\, C \lim_{t\to\infty} \int_0^t \frac{1}{1+\omega^2\tau^2}\, d\tau \\ &= \frac{4R_{xx}(0)\, C}{\omega} \lim_{t\to\infty} \arctan \tau\omega \Big|_0^t \\ &= \frac{4R_{xx}(0)\, C}{\omega} \cdot \frac{\pi}{2} \\ &= \frac{2\pi R_{xx}(0)\, C}{\omega} \end{aligned} \tag{1.43}$$

C is the proportionality constant and it can be seen, that for many distributed trap sites, a noise spectrum of $\frac{1}{f}$ is found.

Again, using the *AllanTools* library [31], flicker noise was simulated to give an impresion of its properties.



(a) Time domain (b) Power spectral density (c) Allan deviation

Figure 1.10.: Different representations of flicker noise for different $\bar{\tau}_1$ and fixed $\bar{\tau}_0 = 1\,s$.

The small wiggles at longer τ are typical end-of-data errors caused by spectral leakage, because there are insufficient samples to average over [12]. As it was discussed above, the Allan deviation can only be estimated given a limited number of samples using equation 1.27. This leads to the fact, that at $\frac{\tau}{2}$ there are only 2 samples left, so there no averaging possible to improve the estimate of the Allan deviation. This leads to oscillations at low frequencies or large τ.

The coefficients given here are using the assumption, that the Allan deviation is the appropriate measure for the sample data. This might not always be the case, because the Allan deviation assumes a dead time of $\theta = 0$. This problem was extensively discussed by Barnes et al. [4] and even special models were developed to account for the algorithms of modern frequency counters [7]. It is therefore important to discuss typical measurement settings for voltmeter to estimate errors that arise from those settings. Typical settings, that affect the dead time of a voltmeter are auto-zeroing and line synchronization. Auto-zeroing is typically done by adding additional measurements to the normal input integratiion cycle. These measurements are a zero measurement to correct for offset drift and a measurement of the reference voltage to correct for gain errors. The implementation details and type of measurements are manufacturer dependant and must be determinded for every multimeter used.

1.6. Temperature Controller

1.6.1. Tuning of a PID controller

The number of emperical algorithms to determine a set of PID parameters (K_p, K_i, K_d) are numerous. In this work only the most common algorithms and a few notable exceptions will be presented.

1.6.2. Design

Bibliography

[1] D.W. Allan. "Statistics of atomic frequency standards". In: *Proceedings of the IEEE* 54.2 (1966), pp. 221–230. DOI: 10.1109/PROC.1966.4634.

[2] David W. Allan. "Should the classical variance be used as a basic measure in standards metrology?" In: *IEEE Transactions on Instrumentation and Measurement* IM-36.2 (1987), pp. 646–654. DOI: 10.1109/TIM.1987.6312761.

[3] K.J. Åström and R.M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2010. ISBN: 9781400828739. URL: https://fbswiki.org/wiki/index.php/Feedback_Systems:_An_Introduction_for_Scientists_and_Engineers.

[4] James A. Barnes et al. "Characterization of Frequency Stability". In: *IEEE Transactions on Instrumentation and Measurement* IM-20.2 (1971), pp. 105–120. DOI: 10.1109/TIM.1971.5570702.

[5] Patrick Baus. *TinkerforgeAsync*. Version 1.2.0. July 2021. URL: https://github.com/PatrickBaus/TinkerforgeAsync.

[6] K.B. Cook and A.J. Brodersen. "Physical origins of burst noise in transistors". In: *Solid-State Electronics* 14.12 (1971), pp. 1237–1242. ISSN: 0038-1101. DOI: 10.1016/0038-1101(71)90112-2. URL: https://www.sciencedirect.com/science/article/pii/0038110171901122.

[7] Samuel T. Dawkins, John J. McFerran, and Andre N. Luiten. "Considerations on the measurement of the stability of oscillators with frequency counters". In: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 54.5 (2007), pp. 918–925. DOI: 10.1109/TUFFC.2007.337.

[8] Pieter Eendebak et al. *qtt*. 2022. URL: https://github.com/QuTech-Delft/qtt (visited on 11/03/2022).

[9] Nathan Flowers-Jacobs et al. *The NIST Johnson noise thermometry system for the determination of the Boltzmann constant*. en. Dec. 2017. DOI: 10.6028/jres.122.046. URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=923576.

[10] C.A. Greenhall. "Spectral ambiguity of Allan variance". In: *IEEE Transactions on Instrumentation and Measurement* 47.3 (1998), pp. 623–627. DOI: 10.1109/19.744312.

[11] Charles Greenhall. "Removal of drift from frequency stability measurements". In: *Telecommunications and Data Acquisition Progress Report* (Nov. 1981).

[12] David Howe. "Interpreting Oscillatory Frequency Stability Plots". en. In: 2002 IEEE Intl. Freq. Cont. Symp , New Orleans, LA, Jan. 2002. URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=105279.

[13] N.J. Kasdin and T. Walter. "Discrete simulation of power law noise (for oscillator stability evaluation)". In: *Proceedings of the 1992 IEEE Frequency Control Symposium*. 1992, pp. 274–283. DOI: 10.1109/FREQ.1992.270003.

[14] Arthur Kay. *Analysis and Measurement of Intrinsic Noise in Op Amp Circuits, Part VIII: Popcorn Noise*. Technote VIII. Texas Instruments Incorporated, Feb. 2008.

[15] Arthur Kay. *Operational Amplifier Noise: Techniques and Tips for Analyzing and Reducing Noise*. Elsevier Science, 2012. ISBN: 9780080942438.

[16] Dan Kegel. *The C10K problem*. May 1999. URL: `https://web.archive.org/web/19990508164301/http://www.kegel.com/c10k.html`.

[17] Paul J Leach, Michael Mealling, and Rich Salz. "A Universally Unique IDentifier (UUID) URN Namespace". In: *RFC* 4122 (2005), pp. 1–32. URL: `https://datatracker.ietf.org/doc/html/rfc4122`.

[18] Stefan Machlup. "Noise in Semiconductors: Spectrum of a Two-Parameter Random Signal". In: *Journal of Applied Physics* 25.3 (1954), pp. 341–343. DOI: `10.1063/1.1721637`. URL: `https://doi.org/10.1063/1.1721637`.

[19] A.S. McCormack and K.R. Godfrey. "Rule-based autotuning based on frequency domain identification". In: *IEEE Transactions on Control Systems Technology* 6.1 (1998), pp. 43–61. DOI: `10.1109/87.654876`.

[20] Edoardo Milotti. "1/f noise: a pedagogical review". In: *arXiv e-prints* (Apr. 2002).

[21] B. Pellegrini et al. "$\frac{1}{f^\gamma}$ noise in thick-film resistors as an effect of tunnel and thermally activated emissions, from measures versus frequency and temperature". In: *Phys. Rev. B* 27 (2 Jan. 1983), pp. 1233–1243. DOI: `10.1103/PhysRevB.27.1233`. URL: `https://link.aps.org/doi/10.1103/PhysRevB.27.1233`.

[22] David W. Pessen. "A new look at PID-controller tuning". In: *Journal of dynamic systems, measurement, and control* 116.3 (1994), pp. 553–557.

[23] W. H. Press. "Flicker noises in astronomy and elsewhere." In: *Comments on Astrophysics* 7.4 (Jan. 1978), pp. 103–119.

[24] Jr. Puckett Jason Niles. "An Electrical and Statistical Study of Burst Noise". PhD thesis. California Institute of Technology, 1971.

[25] G.J. Silva, A. Datta, and S.P. Bhattacharyya. *PID Controllers for Time-Delay Systems*. Control Engineering. Birkhäuser Boston, 2007. ISBN: 9780817644239. URL: `https://link.springer.com/book/10.1007/b138796`.

[26] Guillermo J. Silva, Aniruddha Datta, and S.P. Bhattacharyya. "PI stabilization of first-order systems with time delay". In: *Automatica* 37.12 (2001), pp. 2025–2031. ISSN: 0005-1098. DOI: `10.1016/S0005-1098(01)00165-0`. URL: `https://www.sciencedirect.com/science/article/pii/S0005109801001650`.

[27] Sigurd Skogestad. "Simple analytic rules for model reduction and PID controller tuning". In: *Journal of Process Control* 13.4 (2003), pp. 291–309. ISSN: 0959-1524. DOI: `10.1016/S0959-1524(02)00062-8`. URL: `https://www.sciencedirect.com/science/article/pii/S0959152402000628`.

[28] Carel M. Van Vliet and Peter H. Handel. "A new transform theorem for stochastic processes with special application to counting statistics". In: *Physica A: Statistical Mechanics and its Applications* 113.1 (1982), pp. 261–276. ISSN: 0378-4371. DOI: `10.1016/0378-4371(82)90019-X`. URL: `https://www.sciencedirect.com/science/article/pii/037843718290019X`.

[29] R. Vilanova and A. Visioli, eds. *PID Control in the Third Millennium: Lessons Learned and New Approaches*. Advances in Industrial Control. Springer London, 2012. ISBN: 9781447124245. DOI: 10.1007/978-1-4471-2425-2_5.

[30] Richard Von Mises. *Über Aufteilungs-und Besetzungswahrscheinlichkeiten*. na, 1939.

[31] Anders E.E. Wallin. *AllanTools*. 2022. URL: https://github.com/aewallin/allantools (visited on 10/17/2022).

[32] Y. Yamamoto. *Fundamentals of Noise Processes*. Cambridge University Press, 2004. Chap. 9. ISBN: 9780521817486. URL: https://www.nii.ac.jp/qis/first-quantum/e/forStudents/lecture/index.html.

[33] J. G. Ziegler and N. B. Nichols. “Optimum Settings for Automatic Controllers”. In: *Journal of Dynamic Systems, Measurement, and Control* 115.2B (June 1993), pp. 220–222. DOI: 10.1115/1.2899060. eprint: https://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/115/2B/220/5546571/220_1.pdf.

A. Appendix

A.1. Multimeter Settings for the Comparison Test

All were configured for maximum stability and similar conversion times using the following settings via SCPI. For better readability, all commands are shown unabridged.

HP 3458A

```
PRESET NORM;  # reset the device
TARM HOLD;  # stop readings
BEEP;
OFORMAT ASCII;  # return text
TRIG AUTO;  # trigger when ready
NRDGS 1,AUTO; # take 1 reading
DCV 10;
AZERO ON;  # enable autozero
NDIG 9;
NPLC 100;
FIXEDZ OFF;  # High input impedance
TARM AUTO;  # enable readings
```

Keithley Model 2002

```
*CLS;  # clear events and errors
*RST;  # reset all settings
*OPC?;  # wait until device is reset
:INITiate:CONTinuous OFF;  # disable continuous initiation
:ABORt;  # place K2002 in idle
:SYSTem:AZERo:STATe ON;  # enable autozero
:SYSTem:AZERo:TYPE SYNChronous;  # azero for every reading
:SYSTem:LSYNc:STATe ON;  # line sync
:SENSe:VOLTage:DC:RANGe:UPPer 20;
:SENSe:VOLTage:DC:DIGits 9;
:SENSe:VOLTage:DC:NPLCycles 10;
:SENSe:VOLTage:DC:AVERage:COUNt 4;  # the averaging length
:SENSe:VOLTage:DC:AVERage:TCONtrol REPeat;  # filter type
:SENSe:VOLTage:DC:AVERage:ADVanced:STATe OFF;
:SENSe:VOLTage:DC:AVERage:STATe ON;  # Enable averaging
:FORMat:DATA REAL,64;  # read data as doubles
:FORMat:ELEMents READing;  # only return the reading
:FORMat:EXPonent HPRecision;  # Scientific notation
:INITiate:CONTinuous ON;  # Enable continuous triggering
```

Keysight 34470A

```
:SYSTem:BEEP;
:ABORt;
*RST;
*CLS;
:CONFigure:VOLTage:DC;
:SENSe:VOLTage:RANGe 10;
:SENSe:VOLTage:ZERO:AUTO ON;  # enable autozero
:SENSe:VOLTage:NPLCycles 100;
:SENSe:VOLTage:IMPEdance:AUTO ON;  # High input impendance
:FORMat:DATA ASCii,9;  # return 9 digits ASCII
```

Keithley DMM6500

```
SYSTem:BEEPer 500, 0.2;
ABORt;
*RST;
*CLS;
:SENSe:FUNCtion:ON ”VOLTage:DC”;
:SENSe:VOLTage:DC:RANGe:UPPer 10;
:SENSe:VOLTage:DC:LINE:SYNC ON;
:SENSe:VOLTage:DC:AVERage:COUNt 9;  # the averaging length
:SENSe:VOLTage:DC:AVERage:TCONtrol REPeat;  # filter type
:SENSe:VOLTage:AZERo:STATe ON;  # enable autozero
:SENSe:VOLTage:DC:NPLCycles 10;
:SENSe:VOLTage:INPutimpedance AUTO;  # High input impedance
:SENSe:VOLTage:DC:AVERage:STATe ON;  # Enable averaging
:FORMat:DATA ASCii;  # read data as double instead of text
:FORMat:ASCii:PRECision 16;  # return 16 digits ASCII
:DISPlay:VOLTage:DC:DIGits 6;  # set the screen to 6 digits
```