



University of Applied Sciences

Faculty of Engineering and Economics

Campus Meschede

PROJECT WORK

**Development of an Application to Measure
the Quality of Audio Devices using GNU Radio**

Supervisor: Prof. Dr. Benjamin Buchwitz
Date of Approval: March 1, 2022
Submitted by: Bennet Luke Ladage
Address: 59872 Meschede
Germany
E-Mail: ladage.bennetluke@fh-swf.de

Acknowledgment

I want to thank professor Benjamin Buchwitz for his support in this work, and especially for adopting my Project Work after professor Sigmar Ries passed away unexpectedly who had the idea of this thesis.

He will be missed as a person, teacher, and also for his spontaneous and marvelous sense of humor in his lectures, e-mails or on other events. He taught me the language of Mathematics and he got me enthusiastic about Fourier's theories and Digital Signal Processing. I would like to say thank you for all, but unfortunately I can't do this personally.

Moreover, I want to thank my parents, Erik Hentschel, and Gero Jung for their mental support and their patience to always listen to my confusing thoughts about this project.

Contents

List of Figures	II
List of Tables	IV
List of Abbreviations	V
1. Introduction	1
2. Theory	2
2.1. Fourier's Representation	2
2.1.1. Correctly Scaled Time and Frequency Axis	6
2.1.2. Leakage	6
2.2. FIR and IIR Filter	8
2.3. Quality of Audio Devices	12
3. Implementation and System Concept	17
3.1. Graphical User Interface	18
3.2. Program Structure and Algorithms	22
3.2.1. SNR, THD and Power Estimator	25
3.2.2. Verification of the SNR's Measurement Accuracy	27
3.2.3. The Input Path of the Measuring System	30
4. Measurement Demonstration	32
4.1. Characterization of Sound Cards	32
4.2. Comparison of Two Amplifiers	36
5. Conclusion and Future Work	40
References	42
A. Decibel, Power, Energy and RMS	44

List of Figures

2.1. Cyclic representation of a periodic signal (based on [2, Figure 1.3]).	2
2.2. Fourier's representations (based on [2, Figure 1.5, 1.7, 1.9, 1.11]).	3
2.3. Fourier-Poisson cube (based on [2, Figure 1.22]).	5
2.4. Time and frequency axis for the signal length $N = 8$ (based on [3, p. 53]).	6
2.5. Single-sided and sampled spectrum of a cosine wave without leakage (a) and with leakage (b) (based on [6, Figure 3-10]).	7
2.6. FIR filter with M coefficients (based on [7, Figure 4.4]).	8
2.7. FIR coefficients and Gibbs Phenomenon.	9
2.8. The transfer function of an 89-tap FIR filter, using a Kaiser window $\beta = 5$, has got a stopband attenuation of $\delta_{st} = -54$ dB (single-sided spectrum).	10
2.9. IIR filter (based on [7, Figure 5.5]).	11
2.10. The poles p_1, p_2 and the zeros z_1, z_2 of (a) set the magnitude of the single-sided transfer function in (b). A magnitude of the complex root, close to 1, is a gain (pole) or an attenuation (zero).	12
2.11. The device under test (DUT) is displayed as a black-box and it gets stimulated by a test signal $x_{\text{test}}(t)$ while the resulting output $y_{\text{res}}(t)$ is measured.	13
2.12. Clipping of a sine wave leads to a harmonic distortion.	14
2.13. Linear dynamic range (LDR) of an amplifier (based on [14, Figure 10.1]).	14
 3.1. Demonstration of GNU Radio Companion.	17
3.2. The measurement concept of the application. The device under test (DUT) is treated as a black-box.	18
3.3. This is the GUI to measure the quality indicators. On the left hand side, there are the input, output and data logger settings. The blue signals displays the output's test signal in the middle. Next to this, there are the scopes of the input signal. The footer shows the gain, the THD+N, and the SNR.	19
3.4. All possible tabs of the setting menu (a, b), the data logger settings (c), and an example CSV-file (d, e).	20
3.5. A classical sweep with $B_{\text{sw}} = 20.0$ kHz (a) and a bandpass sweep with $f_{\text{cen}} = 10$ kHz and $B_{\text{bpsw}} = 2$ kHz (b).	20
3.6. Signal Processing of the measurement system's output path.	22
3.7. Kroenecker delta used as a switch where $S_a \in \{1, 2, 3\}$ is the switch's position. .	23
3.8. An IIR filter of cascaded second order sections which provide robustness against quantization errors (based on [7, Figure 5.8]).	25
3.9. Structure of the SNR, THD and Power Estimator with one input $y_q(n)$ and three outputs P_s , $\text{SNR}_{\text{dB},\text{out}}$, and THD+N.	25

3.10. The scatter plot shows the estimated SNR, and the black curve is the theoretical SNR.	29
3.11. Signal Processing of the measurement system's input path.	30
4.1. Measuring of the sound card's peak voltage U_p , using a phone connector. Here, the left channel is measured by the voltmeter V (based on [23, p. 288/289]).	32
4.2. The usage of the classical sweeps leads to the sound cards' transfer functions. . . .	35
4.3. The Fisher equalizer (a) and Wuhzi's low cost amplifier (b) are the DUTs.	36
4.4. Wiring diagram – The PC is connected to the DUT whereby the right channel is measured.	37
4.5. The spectra show the reasons for the different noise and distortion indicators. . . .	38
4.6. Transfer functions of the DUTs. (a) shows a highpass characteristic and (b) behaves like a lowpass.	39

List of Tables

2.1. Kaiser window and maximum stopband ripple δ_{st} (based on [7, p. 279]).	10
3.1. Measurement results of the SQNR testing, having a sample size K and a significance level α	28
4.1. Measurement results of the two sound cards A and B.	33
4.2. Test signal parameters and quality indicators of the DUTs.	37
A.1. Important decibel values to interpret levels faster (based on [6, Table E-1]).	45

List of Abbreviations

A/D	analog-to-digital
BP	bandpass
BR	bandreject
CSV	comma-separated values
D/A	digital-to-analog
DFT	Discrete Fourier Transform
DSP	Digital Signal Processing
DUT	device under test
FD	frequency domain
FIR	finite impulse response
FS	Fourier Series
GUI	graphical user interface
ICFT	Inverse Continuous Fourier Transform
IDFT	Inverse Discrete Fourier Transform
IDTFT	Inverse Discrete-Time Fourier Transform
IFFT	Inverse Fast Fourier Transform
IIR	infinite impulse response
LDR	linear dynamic range
LP	lowpass
NCO	Numerically Controlled Oscillator
PC	personal computer
RMS	root mean square
SDR	Software Defined Radio
SNR	signal-to-noise ratio
SOS	second order section
SQNR	signal-to-quantization noise ratio
SWIG	Simplified Wrapper and Interface Generator
TD	time domain
THD	total harmonic distortion
THD+N	total harmonic distortion plus noise
USRP	Universal Software Radio Peripheral

1. Introduction

This thesis represents a system concept which determines quality indicators of audio devices. It is implemented by GNU Radio, but the concept is described in a way that other software and hardware can realize this system, too. GNU Radio is chosen as the development environment because of its availability for all main operating systems (i.e., Windows, Linux, and Mac) and due to its open-source license. This software has access to the sound card of the personal computer (PC) which is used as an input/output interface for real world signals. Thus, it transfers the PC into a measuring instrument. The second main topic is hence whether the quality of a common sound card is sufficient that the PC is suitable to measure audio devices.

The considered quality indicators are the transfer function, the signal-to-noise ratio (SNR), the total harmonic distortion (THD), the gain and the linear dynamic range (LDR) of audio equipment. If we measure an external audio device, we connect the input of the device under test (DUT) to the sound card's output, and the DUT's output is wired to the sound card's input. This approach is a black-box testing method. For example, equipment like amplifiers, equalizers, mixing consoles, filters, crossover networks, and analog-to-digital (A/D) or digital-to-analog (D/A) converters can be inspected. However, the following test concepts are not adequate for the perceived audio quality of devices which uses low bit rate coding schemes. In addition, the system has a signal generator, a data logger for the quality indicators, an adjustable digital input filter, and a time domain and frequency domain display of the output and input signals. The GNU Radio application can be downloaded via the author's Github page [1].

The next chapter will show the necessary theory to understand the thesis. Chapter 3 deals with the implemented graphical user interface (GUI) and how the system concept works. A measurement demonstration is discussed in chapter 4 where sound cards are characterized as a measuring instrument. Furthermore, there are two amplifiers from different decades (1986 and 2020) which are compared by the measurement results of the application. The last chapter is a conclusion of the thesis and it gives an outlook of the application's future version.

2. Theory

2.1. Fourier's Representation

Fourier's representation is a method to use a linear combination or an integral representation of complex exponentials

$$e^{j(\cdot)} = \cos(\cdot) + j \sin(\cdot) \quad (2.1)$$

to synthesize certain functions [2]. Every kind of a Fourier representation has got a *synthesis equation* and an *analysis equation*. The synthesis equation uses a function of the frequency domain (FD) to synthesize a function in the time domain (TD). The frequency domain is also called *spectrum*. The analysis equation uses a function of the time domain to calculate the corresponding function in the frequency domain. Therefore, the analysis equation is the inverse of the synthesis equation and vice versa.

The Fourier Series (FS) is the special case to synthesize *continuous* and *periodic* functions like the function in fig. 2.1 (a) [2]. A periodic signal $s(t)$ repeats its shape at intervals of the *signal period* T . A useful representation of periodic signals is to wind one period around a circle with

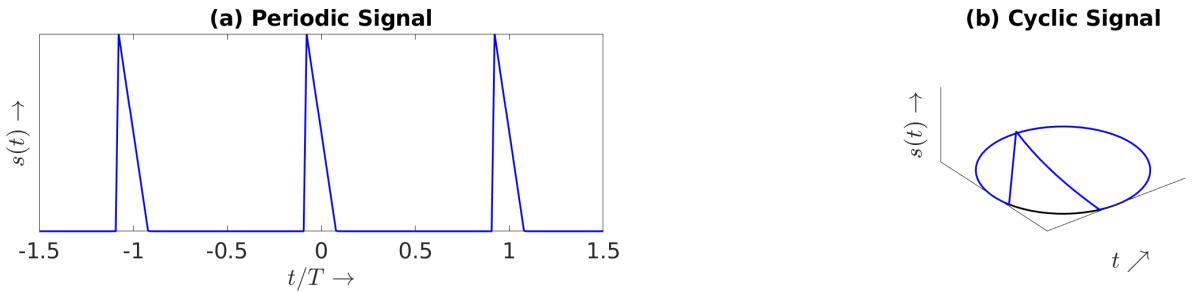


Figure 2.1.: Cyclic representation of a periodic signal (based on [2, Figure 1.3]).

the circumference T [2]. This is displayed in fig. 2.1 (b). Thus, a periodic function can be called *cyclic* [3]. The synthesis equation of the Fourier Series is a linear combination of orthogonal complex exponentials [2]:

$$s(t) = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi kt/T} = \sum_{k=-\infty}^{\infty} |c_k| e^{j\varphi_k} e^{j2\pi kt/T}, \quad k \in \mathbb{Z}. \quad (2.2)$$

If Euler's identity (2.1) is used, the synthesis equation (2.2) can be interpreted as a superposition of cosine waves and sine waves. The complex value c_k is called the k -th *Fourier Coefficient* for $s(t)$ [2]. It is the magnitude $|c_k|$ and phase shift φ_k of the k -th complex oscillation $e^{j2\pi kt/T}$ [4]. Hence, c_k contains everything to represent the spectrum of a cyclic signal: The change in magnitude and phase versus the frequency. The Fourier Coefficients are a sequence therefore the spectrum is called *discrete* [4]. If the magnitude spectrum is displayed, then $|c_k|$ appears at

the frequency

$$f_k = \frac{k}{T} = kf_1, \quad k \in \mathbb{Z}. \quad (2.3)$$

This leads to the space

$$f_1 = \frac{1}{T} \quad (2.4)$$

between each discrete frequency. The frequency f_1 is called the *fundamental* and all other frequencies are called the *harmonics*. If we want to detect the k -th complex oscillation of a time signal, we use the following analysis equation [2]:

$$c_k = \frac{1}{T} \int_0^T s(t) e^{-j2\pi kt/T} dt. \quad (2.5)$$

An example for the Fourier Series is shown in fig. 2.2 (a, b). In (a), there is a periodic and continuous rectangle function in the time domain. Next to it is the corresponding magnitude spectrum which is the Fourier Coefficient's magnitude and a sinc sequence. Moreover, a common

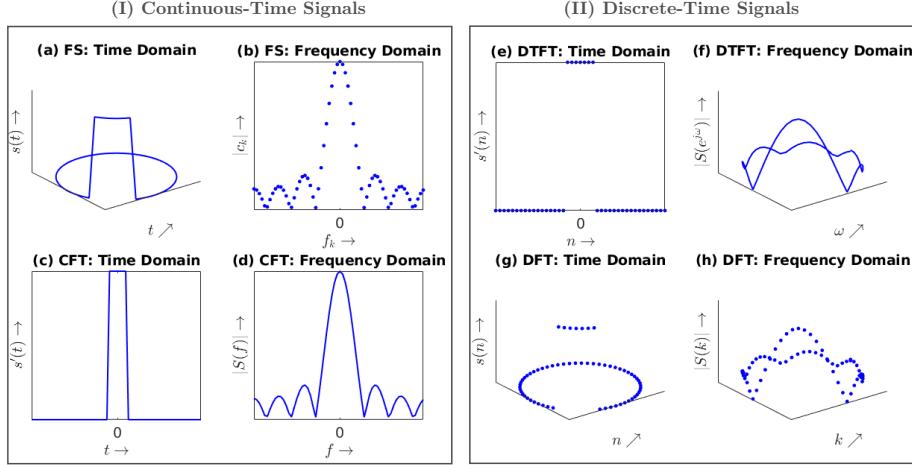


Figure 2.2.: Fourier's representations (based on [2, Figure 1.5, 1.7, 1.9, 1.11]).

notation for the usage of the analysis equation is the following *transformation sign*

$$s(t) \circ \bullet c_k = \dots \quad (2.6)$$

or for the synthesis equation vice versa

$$c_k \bullet \circ s(t) = \dots \quad (2.7)$$

For a *non-periodic* and *continuous* signal $s'(t)$ on the real line \mathbb{R} exists the *Continuous Fourier Transform* (CFT) [2]:

$$S(f) = \int_{-\infty}^{\infty} s'(t) e^{-j2\pi ft} dt \in \mathbb{C} \quad t \in (-\infty; \infty). \quad (2.8)$$

The synthesis equation, associated to the CFT, is called the *Inverse Continuous Fourier Transform* (ICFT):

$$s'(t) = \int_{-\infty}^{\infty} S(f) e^{j2\pi ft} df \quad f \in (-\infty; \infty). \quad (2.9)$$

Thus, the CFT connects a continuous and non-periodic time domain with a continuous and aperiodic spectrum. An example for this is fig. 2.2 (c, d). There is a rectangular pulse $\Pi'(t)$ which has got the sinc function $\text{si}(\pi f)$ as a spectrum [4]. Considering eq. (2.6), we write

$$\Pi'(t) \circ\bullet \text{si}(\pi f) = \frac{\sin(\pi f)}{\pi f}. \quad (2.10)$$

There is another analysis equation for an *aperiodic* and *discrete* signal $s'(n)$ [3]:

$$S(e^{j\omega}) = \sum_{n=-\infty}^{\infty} s'(n)e^{-j\omega n}. \quad (2.11)$$

This is the *Discrete-Time Fourier Transform* (DTFT). The synthesis equation can be written as

$$s'(n) = \frac{1}{2\pi} \int_0^{2\pi} S(e^{j\omega})e^{j\omega n} d\omega \quad (2.12)$$

which is the inverse¹ of eq. (2.11). The signal $s'(t)$ is equidistantly sampled by the *sample period* T_s [5]:

$$s'(t)|_{t=nT_s} = s'(nT_s) =: s'(n), \quad n \in (-\infty; \infty). \quad (2.13)$$

The n indicates the index of the n -th sample. Before sampling, the continuous signal $s'(t)$ must be bandlimited by an analog lowpass filter and the condition

$$f_s = \frac{1}{T_s} > 2f_c \quad (2.14)$$

must hold for the *sampling frequency* f_s to achieve a correct sampling process [5]. Thereby, f_c is the cutoff frequency of the filter. The ω in eq. (2.11) and (2.12) is the *normalized angular frequency* which has got the dimension of an angle and not of s^{-1} [3]. The normalization results from the sampling frequency:

$$\omega = 2\pi \frac{f}{f_s}. \quad (2.15)$$

The DTFT connects a discrete and non-periodic time domain with a continuous and cyclic spectrum. The circle in the spectrum has got an circumference of 2π . An example for this is fig. 2.2 (e, f). If we compare the plot of the FS and DTFT in fig. 2.2, we will notice that they are the opposite of each other.

We do not have a Fourier representation of discrete signals with a discrete spectrum until now. The solution for this is the *Discrete Fourier Transform* (DFT) which connects a discrete and cyclic time domain with a discrete and cyclic frequency domain [2]. This is shown in fig. 2.2 (g, h). The DFT has got the analysis equation

$$S(k) = \sum_{n=0}^{N-1} s(n)e^{-j2\pi kn/N} \quad (2.16)$$

where the N is the number of samples from the discrete signal $s(n)$ [2]. N is also called the signal length of $s(n)$ or $S(k)$. Equation (2.16) always converges due to the finite sum. Therefore, it is always computable. Every limited discrete signal can be wound around a circle hence every

¹Inverse Discrete-Time Fourier Transform (IDTFT).

limited discrete signal is cyclic [2], [3]. This means that the DFT is valid for all limited discrete signals. Moreover, we have to note that the DFT multiplies the samples of the frequency domain² by N [3]. Thus, a discrete cosine wave has got the following amplitudes in the spectrum

$$\mathcal{A}_0 \cos\left(\frac{2\pi}{N}k_0n\right) \circlearrowleft N \frac{\mathcal{A}_0}{2} \left(\delta(k - k_0) + \delta(k + k_0 - N) \right). \quad (2.17)$$

where the \mathcal{A}_0 is the amplitude in the time domain and the $\delta(k - k_0)$ is the *Kronecker delta* or a *discrete Dirac impulse* shifted by k_0 [3]. It has got the following properties:

$$\delta(k - k_0) = \begin{cases} 1 & k = k_0 \\ 0 & k \neq k_0 \end{cases} \quad \forall k, k_0 \in \mathbb{Z}. \quad (2.18)$$

The synthesis equation, associated to the DFT, is the *Inverse Discrete Fourier Transform* (IDFT) [2]:

$$s(n) = \frac{1}{N} \sum_{k=0}^{N-1} S(k) e^{j2\pi kn/N}. \quad (2.19)$$

If we compare the DFT and the FS, then we find some similarities again. So, the DFT's time domain is a sampled version of the FS's time domain and the DFT's spectrum is the discrete spectrum of the FS wound around a circle [2]. It is also possible to change between different Fourier representations using the *Fourier-Poisson cube* which is illustrated in fig. 2.3. There is

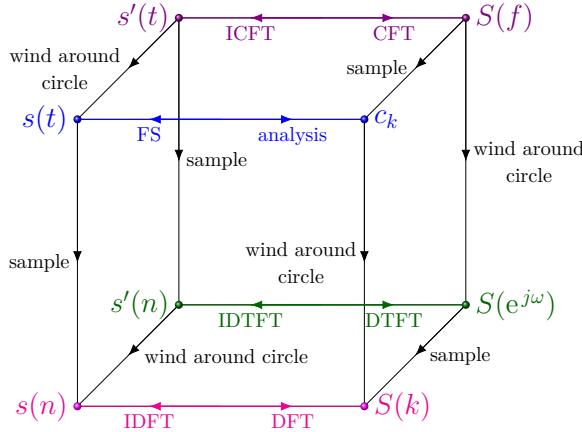


Figure 2.3.: Fourier-Poisson cube (based on [2, Figure 1.22]).

a fast algorithm to calculate the DFT if the condition

$$N = 2^r, \quad r \in \mathbb{N} \quad (2.20)$$

holds [5]. It is called the *Fast Fourier Transformation* (FFT). The fast version of the IDFT is the *Inverse Fast Fourier Transform* (IFFT). If the length $N \neq 2^r$, it is possible to insert zeros at the end of the signal until $N = 2^r$ [5]. This is the method of *zero padding* or *zero filling*.

²The samples of the frequency domain are called frequency bins or bins [6]. The signal's energy is split into all samples of the sequence. Thus, a sample is like a bin for discrete energy. But, this is just a possible reason for this name [6].

2.1.1. Correctly Scaled Time and Frequency Axis

When we are using the DFT, we are interested in the physical dimensions of the time domain and the spectrum. Until now, we have just the index n for the time axis of $s(n)$ and the index k for the frequency axis of $S(k)$. If the signal $s(n)$ is equidistantly sampled, the time of the n -th sample will be

$$t_n = nT_s = n \frac{T}{N}, \quad n \in \{0, 1, 2, \dots, N - 1\} \quad (2.21)$$

where T_s is the sample period, N the signal length and T is the signal's period or the *DFT's analysis time* [3]. The term T/N ensures that the whole analysis time is divided in N equidistant samples. Now, we get equally spaced samples in the frequency domain by dividing the sampling frequency f_s by the length N . This leads to the fundamental f_1 which is also the space between each discrete frequency. With this, we obtain the k -th discrete frequency of the spectrum $S(k)$ [3]:

$$f_k = kf_1 = k \frac{f_s}{N}, \quad k \in \{0, 1, 2, \dots, N - 1\}. \quad (2.22)$$

The cyclic time and frequency axis of the DFT or the IDFT are enrolled in fig. 2.4. This picture illustrates eq. (2.21) and (2.22) for a signal length of $N = 8$. The blue or magenta circle indicates

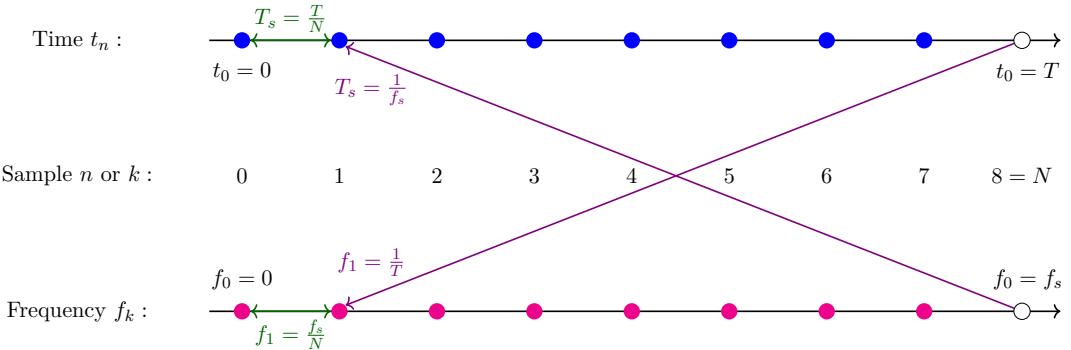


Figure 2.4.: Time and frequency axis for the signal length $N = 8$ (based on [3, p. 53]).

a discrete sample t_n or f_k . The white circle visualizes the return to the first sample 0 when N is reached. The return is necessary due to the circularity of the discrete signal. In addition, the relationship (2.14) between the sample frequency f_s and the sample period T_s is pictured and the relationship (2.4) between the fundamental f_1 and the signal's period T , too.

2.1.2. Leakage

We have to sample a continuous signal for a finite analysis time T to compute its DFT [6]. For example, this continuous signal $x'(t)$ is an audio waveform with a cosine shape and it has got a certain frequency f_x . The finite analysis time can be expressed as a multiplication by a rectangular pulse with the amplitude one and the duration T . This bounds the audio waveform and leads to sinc pulses in the spectrum when the CFT is applied [4]:

$$x'(t) = \cos(2\pi f_x t) \Pi_w(t) \circledast \frac{1}{2} \left(\delta(f - f_x) + \delta(f + f_x) \right) * \text{si}_w(\pi f) \quad (2.23)$$

$$= \frac{1}{2} \left(\text{si}_w(\pi(f - f_x)) + \text{si}_w(\pi(f + f_x)) \right) = X(f) \quad (2.24)$$

Here, the rectangular pulse $\Pi_w(t)$ is called a *window*. Considering the Fourier-Poisson cube in fig. 2.3, we sample the continuous spectrum $X(f)$ and wind it around a circle, or vice versa, to achieve the discrete spectrum $X(k)$ which a computer or other digital device can handle [5]. If the frequency f_x of the cosine wave is

$$f_x \neq m \frac{f_s}{N}, \quad m \in \mathbb{N}, \quad (2.25)$$

then the signal is not sampled at the zeros of the sinc function and *leakage* appears [6]. This means, that the sampled diracs $\delta(f \pm f_x)$ do not fit in one frequency bin and they leak in more bins. This is shown in fig. 2.5 where (a) displays a single-sided spectrum without leakage and (b) with a leaking dirac. Single-sided refers to a spectrum where only the positive frequencies are displayed. The negative frequencies are redundant because they are the complex conjugated and mirrored positive frequencies of a real time signal [3]. Leakage also appears if the analysis time T does not satisfy the following condition [5]:

$$T = mT_x = m \frac{1}{f_x}, \quad m \in \mathbb{N}, \quad T > T_x. \quad (2.26)$$

This means, that multiple periods T_x of the signal $x'(t)$ can occur during the analysis time T , but they must not be truncated. The leakage effect can be reduced by multiplying another

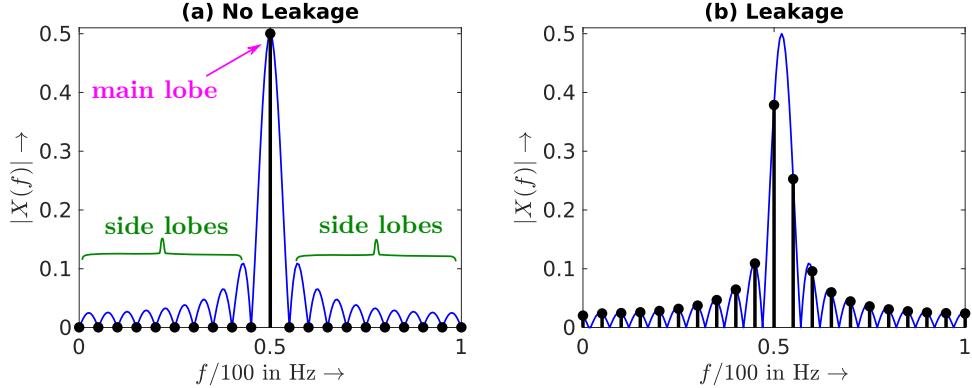


Figure 2.5.: Single-sided and sampled spectrum of a cosine wave without leakage (a) and with leakage (b) (based on [6, Figure 3-10]).

window function $g_w(t)$ before computing the DFT:

$$x'_{\text{win}}(t) = \cos(2\pi f_x t) g_w(t) \circledast \frac{1}{2} \left(\delta(f - f_x) + \delta(f + f_x) \right) * G_w(f) = X_{\text{win}}(f). \quad (2.27)$$

The function $g_w(t)$ minimizes the magnitude of the sinc function's side lobes which are the physical non-existent frequencies due to the leaking [6]. For example, Kaiser, Hamming or Blackman windows are in common use for windowing [6], [7].

2.2. FIR and IIR Filter

A *Finite Impulse Response* (FIR) filter convolves the input sequence $x(n)$ by its *impulse response* $h(n)$ to generate the output sequence $y(n)$ [7]:

$$y(n) = h(n) * x(n) = \sum_{m=-\infty}^{\infty} h(m)x(n-m). \quad (2.28)$$

The impulse response is the output of the filter when the input is a dirac impulse

$$\delta(n) = \begin{cases} 1 & n = 0 \\ 0 & \text{otherwise} \end{cases}. \quad (2.29)$$

The effect of spectral filtering is obtained using the DTFT [7]:

$$y(n) = h(n) * x(n) \circledast X(e^{j\omega})H(e^{j\omega}) = Y(e^{j\omega}). \quad (2.30)$$

Thus, the FIR filter multiplies the input spectrum $X(e^{j\omega})$ by the *transfer function* $H(e^{j\omega})$ of the filter. This complex function describes the change in magnitude and phase versus the frequency. The name *Finite Impulse Response* results from the fact that $h(n)$ always converges [6]. The practical sequence $h(n)$ is finite because the memory of a filter is limited in digital implementations. Hence, a common notation for the practical impulse response is

$$h(n) = (b_0 \ b_1 \ b_2 \ \dots \ b_M) \quad (2.31)$$

where b_n is the n -th coefficient. Then, the convolution of the filter

$$y(n) = x(n) * h(n) = \sum_{m=0}^M b_m x(n-m) \quad (2.32)$$

can be displayed as a digital circuit, as illustrated in fig. 2.6. There are triangles which are

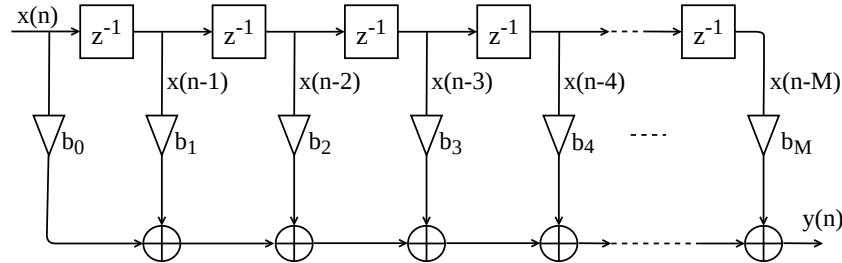


Figure 2.6.: FIR filter with M coefficients (based on [7, Figure 4.4]).

multiplications and the z^{-1} is a delay by one sample (unit delay). Figure 2.6 shows that we need to tap of M separate input values to calculate one output value [6]. Therefore, the name *M -tap FIR filter* is in common use [6]. One can equivalently say that the filter has got M coefficients or the order M . A FIR filter is also always stable due to the finite impulse response [3]. If the impulse response is symmetric, the phase $\arg(H(e^{j\omega}))$ will be linear [6]. A linear

phase leads to a constant group delay

$$\tau_g(\omega) = -\frac{d}{d\omega} \arg(H(e^{j\omega})) \quad (2.33)$$

which is required to have no phase distortion [6], [7]. Then, all frequency components of the input $x(n)$ are delayed by the same time

$$\tau_g = \frac{M}{2f_s} \quad (2.34)$$

before they reach the output of the M -tap filter [6].

Designing a FIR filter can be done by specifying the desired transfer function $H_d(e^{j\omega})$ for a start. An optimal lowpass filter is a rectangular function $\Pi(e^{j\omega})$ as pictured in fig. 2.7 (a). Now, we calculate the IDTFT to obtain the coefficients [6]:

$$H_d(e^{j\omega}) = \Pi(e^{j\omega}) \bullet \circ \text{si}'(n) = h(n). \quad (2.35)$$

Hence, the impulse response is an infinite sinc sequence $\text{si}'(n)$ but the real number of coefficients is limited because a practical system has a limited amount of storage (finite multiplications, additions and unit delays). Thus, a natural window is given which is a rectangular pulse of M samples and an amplitude of one [6]. This is illustrated in fig. 2.7 (b). If we want to calculate the practical transfer function $H(e^{j\omega})$ of a M -tap filter, then we multiply the impulse response by the window. This leads to a convolution of the desired transfer function and the transformed window [6]:

$$h(n)\Pi_w(n) \circ \bullet \Pi(e^{j\omega}) * \text{si}_w(e^{j\omega}) = H(e^{j\omega}). \quad (2.36)$$

The ripples in fig. 2.7 (c) results from the convolution by the window's sinc function $\text{si}_w(e^{j\omega})$. The ripple is called the *Gibbs Phenomenon* which also occurs if the sum of a Fourier Series (2.2) is finite [2], [6]. The highest ripple is the main lobe or passband in this figure. The smaller cones

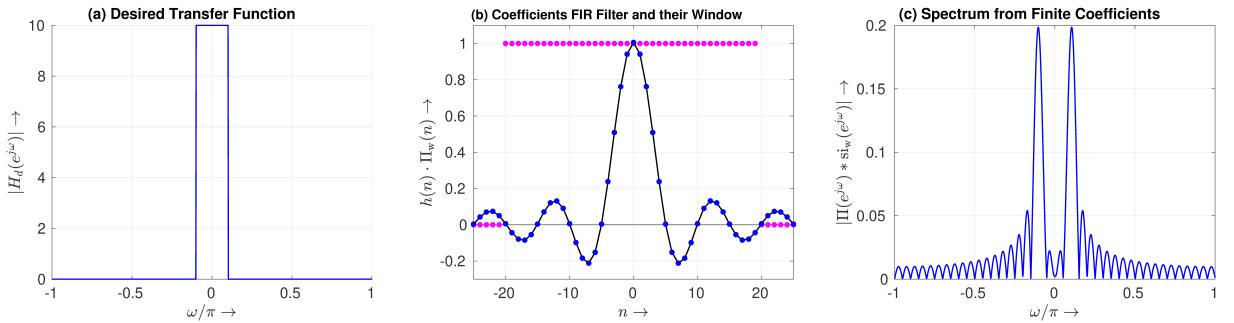


Figure 2.7.: FIR coefficients and Gibbs Phenomenon.

are the side lobes or the stopband. The width of the ripples decreases when more coefficients are used but the height remains approximately the same [7]. The highest sidelobe of the practical transfer function $H(e^{j\omega})$ is the maximum stopband ripple δ_{st} and is always -21 dB due to the use of a rectangular window [7]. Hence, the only way to achieve a stronger stopband attenuation is to use a different window function. The Kaiser window $\kappa_w(n, \beta)$ is an optimal window [7]

whose stopband attenuation δ_{st} is adjusted by the shape parameter β [8]:

$$\beta = \begin{cases} 0.1102 (|\delta_{st}| - 8.7) : & |\delta_{st}| > 50 \text{ dB} \\ 0.5842 (|\delta_{st}| - 21)^{0.4} + 0.07886 (|\delta_{st}| - 21) : & 21 \text{ dB} \leq |\delta_{st}| \leq 50 \text{ dB} \\ 0 : & |\delta_{st}| < 21 \text{ dB} \end{cases}. \quad (2.37)$$

See table 2.1 for a list of often used betas and their corresponding passband ripples. Instead of

Table 2.1.: Kaiser window and maximum stopband ripple δ_{st} (based on [7, p. 279]).

β	2	3	4	5	6	7	8	9
δ_{st} in dB	-29	-37	-45	-54	-63	-72	-81	-90

using a natural window (i.e., a rectangular window), we multiply the impulse response $h(n)$ by the Kaiser window to obtain the practical transfer function $H(e^{j\omega})$ with different ripples [6]:

$$h(n)\kappa_w(n, \beta) \circledast \Pi(e^{j\omega}) * \kappa_w(e^{j\omega}, \beta) = H(e^{j\omega}). \quad (2.38)$$

The transfer function of an 89-tap filter is displayed in fig. 2.8 which is using a Kaiser window $\beta = 5$. The stopband attenuation is $\delta_{st} = -54$ dB there. The cutoff frequency is denoted by ω_c and occurs, by definition, at an attenuation of -3 dB. The passband attenuation is denoted by δ_p and is 3 dB for this filter. The edge frequency ω_{st} of the stopband occurs when the stopband attenuation is reached for the first time. The interval $\Delta\omega = |\omega_{st} - \omega_c|$ is the transmission width of the filter. The minimal filter order M is predictable by the stopband ripple in decibel and the narrowest transition width $\Delta\omega_{\min}$ [8]:

$$M = \left\lceil \frac{|\delta_{st}| - 8}{2.285 \Delta\omega_{\min}} + 1 \right\rceil = \left\lceil \frac{|\delta_{st}| - 8}{2.285 |\omega_{st} - \omega_c|} + 1 \right\rceil = \left\lceil \frac{|\delta_{st}| - 8}{2.285 \left| \frac{2\pi}{f_s} (f_{st} - f_c) \right|} + 1 \right\rceil. \quad (2.39)$$

Here, f_s is the sampling frequency, f_{st} the edge frequency in Hz, and $\lceil \cdot \rceil$ is the rounding to the nearest integer.

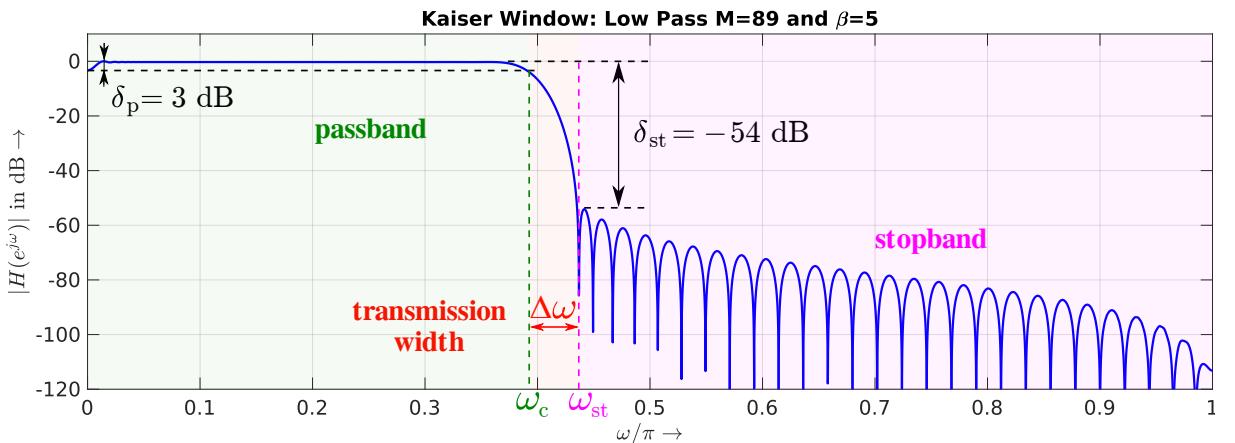


Figure 2.8.: The transfer function of an 89-tap FIR filter, using a Kaiser window $\beta = 5$, has got a stopband attenuation of $\delta_{st} = -54$ dB (single-sided spectrum).

IIR Filter

An *Infinite Impulse Response* (IIR) filter implements a linear time-invariant system by its *difference equation*

$$y(n) = \sum_{m=0}^M b_m x(n-m) - \sum_{i=1}^L a_i y(n-i) \quad (2.40)$$

which is the discrete equivalent to a linear differential equation with constant coefficients [7]. Here, b_m and a_i are the constant coefficients. Equation (2.40) may be displayed as a digital circuit, as shown in fig. 2.9. The sum over M is a FIR filter which gets connected to the feedback of the output values $y(n)$. The feedback is the sum over L . We obtain the transfer

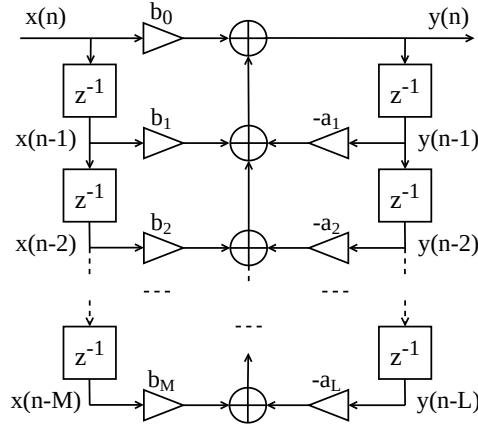


Figure 2.9.: IIR filter (based on [7, Figure 5.5]).

function of the filter by applying the DTFT to eq. (2.40):

$$H(e^{j\omega}) = \frac{Y(e^{j\omega})}{X(e^{j\omega})} = \frac{\sum_{m=0}^M b_m e^{-j\omega m}}{1 - \sum_{i=1}^L a_i e^{-j\omega i}} \bullet \circ h(n). \quad (2.41)$$

Then, the impulse response $h(n)$ can be calculated by the IDTFT of the transfer function. The highest value of M or L sets the *filter order*, for example, it has got the order M if $M > L$ [6].

The feedback leads to an infinite long impulse response hence the name IIR filter. Thus, the filter can be *unstable*, i.e., $h(n)$ diverges [7]. A digital filter is often executed by a computer or microcontroller. As a consequence, the computer can get an overflow or crashes if the filter is unstable. We can calculate the roots p_i of the denominator to check the filter's stability [7]. These roots are called the *poles*. Now, we substitute $z = e^{j\omega}$ in eq. (2.41), exclude b_0 from the numerator, multiply the term by $1 = z^{M-M}/z^{L-L}$, and obtain

$$H(z) = b_0 \frac{(1 + \sum_{m=1}^M \tilde{b}_m z^{-m})}{(1 - \sum_{i=1}^L a_i z^{-i})} \frac{z^{M-M}}{z^{L-L}} = b_0 z^{L-M} \frac{(z^M + \sum_{m=1}^M \tilde{b}_m z^{M-m})}{z^L - \sum_{i=1}^L a_i z^{L-i}} = R(z) \frac{N(z)}{D(z)} \quad (2.42)$$

$$N(z) = 0 \rightsquigarrow z_1, z_2, \dots, z_M \quad z_m \in \mathbb{C} \quad (2.43)$$

$$D(z) = 0 \rightsquigarrow p_1, p_2, \dots, p_L \quad p_i \in \mathbb{C} \quad (2.44)$$

which leads to L poles p_i , M zeros z_m and $r = |L - M|$ additional roots [7], [9]. Here, b_0 is a constant gain and $\tilde{b}_m = b_m/b_0$. The z^{L-M} is just a delay of r samples at the output [9]. This

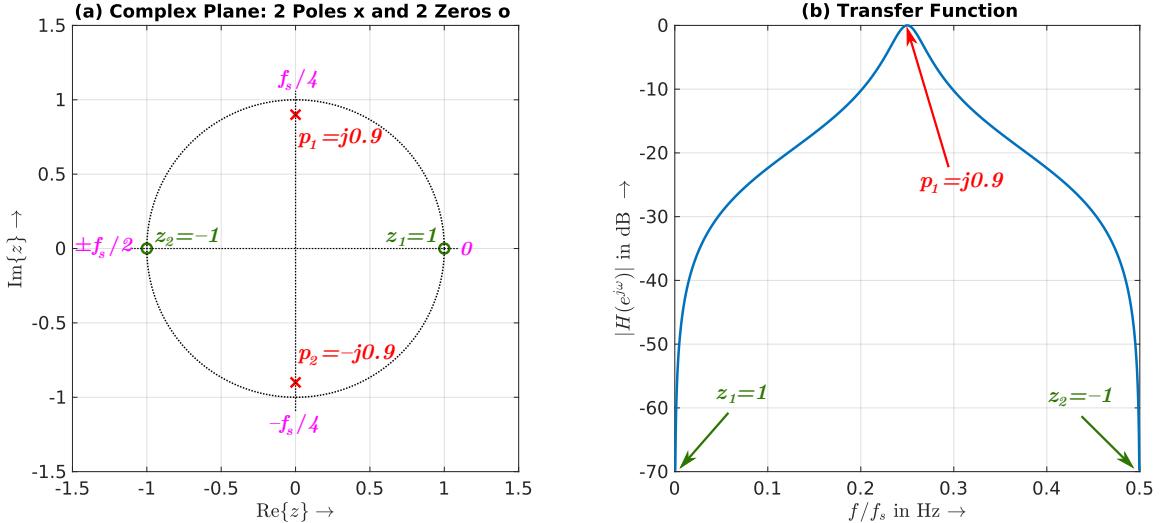


Figure 2.10.: The poles p_1, p_2 and the zeros z_1, z_2 of (a) set the magnitude of the single-sided transfer function in (b). A magnitude of the complex root, close to 1, is a gain (pole) or an attenuation (zero).

leads to r repeated poles ($M > L$) or r repeated zeros ($M < L$) which are all 0. If the condition

$$|p_i| < 1 \quad \forall i \in \{1, 2, \dots, L\} \quad (2.45)$$

holds, the IIR filter will be stable, i.e., $h(n)$ converges. The zeros z_m do not affect the filter's stability [6]. In a practical digital system, the memory for a coefficient a_k or b_k is limited due to the binary representation of numbers. Then, the coefficients will be truncated or rounded when they are saved or calculated. This leads to a shift of the pole's magnitude that maybe $|p_i| \geq 1$ [8]. Then, the system is unstable. This effect is called the *coefficient quantization*.

If we plot the poles and zeros on the complex plane in fig. 2.10, we notice that the pole's magnitude close to 1 is a gain for the transfer function and the zero's magnitude close to 1 is an attenuation [6]. The angle of the root corresponds to the frequency where the gain or attenuation occurs. The angles $\varphi \in [0; \pi]$ are mapped to the frequencies $f \in [0; f_s/2]$ and $\varphi \in [\pi; 2\pi] \mapsto f \in [-f_s/2; 0]$. Moreover, the roots describe the transfer function eq. (2.41), as well [7]:

$$H(z) = b_0 \frac{\prod_{m=1}^M (1 - z_m z^{-1})}{\prod_{i=1}^L (1 - p_i z^{-1})} \rightsquigarrow H(e^{j\omega}) = b_0 \frac{\prod_{m=1}^M (1 - z_m e^{-j\omega})}{\prod_{i=1}^L (1 - p_i e^{-j\omega})}. \quad (2.46)$$

2.3. Quality of Audio Devices

This section describes the quality of audio devices using classical measurement methods. These methods consider black-box testing of devices like amplifiers, equalizers, mixing consoles, and analog-to-digital (A/D) or digital-to-analog converters (D/A) [10]. The following techniques are not adequate for the perceived audio quality of devices which uses low bit rate coding schemes [11]. Black-box testing means that an analog test signal $x_{\text{test}}(t)$ is fed into the input of the *device under test* (DUT) and the resulting output $y_{\text{res}}(t)$ is measured. $y_{\text{res}}(t)$ is directly interpretable as a quality indicator or it can be used to compute certain values which characterize the audio quality. This concept of testing is shown in fig. 2.11.

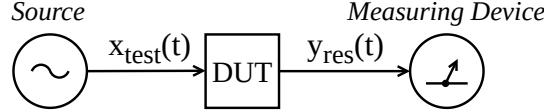


Figure 2.11.: The device under test (DUT) is displayed as a black-box and it gets stimulated by a test signal $x_{\text{test}}(t)$ while the resulting output $y_{\text{res}}(t)$ is measured.

The human ear is able to detect sound waves within the frequency range from 20 Hz to 20 kHz [12]. Thus, an audio signal ranges in this spectrum and an audio device has to operate in this area, too. We determine the frequency characteristic of the DUT by the continuous transfer function $H(f)$. This is a CFT spectrum. We can use the FFT and the Fourier-Poisson cube (fig. 2.3) to calculate this function with a digital device. The requirements of the transfer function distinguish from the device type. For example, an amplifier is expected to have a constant gain over the whole audio range [10]. The gain is the ratio of the output power P_{out} to the input power P_{in} and it is usually displayed in decibels³:

$$G_{\text{dB}} = 10 \log_{10} \left(\frac{P_{\text{out}}}{P_{\text{in}}} \right). \quad (2.47)$$

A typical performance for a consumer preamplifier is a gain which may fluctuate from ± 0.1 dB to ± 0.3 dB [10]. On the other hand, equipment like an analog filter is expected to have no ripple in the passband, a steep transmission width and a highest possible stopband attenuation.

Furthermore, another indicator of the quality is the *effective dynamic range*. This is the output signal's power P_{sig} divided by the power of the output noise P_{noise} for analog devices. This value is commonly given in decibels and it is called *signal-to-noise ratio* (SNR) [6], [10]:

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \left(\frac{P_{\text{sig}}}{P_{\text{noise}}} \right). \quad (2.48)$$

The measurement of the SNR uses typically a sine wave of $f_{\text{test}} = 1$ kHz or $f_{\text{test}} = 0.997$ kHz as a test signal [10], [13]. Then, the power of the sine wave is P_{sig} . P_{noise} is the power of analog noise and any other undesired signals. Analog noise is caused by random motions of charges or charge carriers in the DUT [14]. These motions have different origins, e.g. thermal noise is caused by thermal vibration of bound charges. It increases as the temperature or the bandwidth increases. There are more kinds of analog noise like shot noise, $1/f$ -noise or flicker noise [14] which are all included in P_{noise} . An acceptable level is a $\text{SNR}_{\text{dB}} > 40$ dB, this means that the signal power is 10 000 times greater than the noise level [10]. A satisfactory value is a $\text{SNR}_{\text{dB}} = 70$ dB at 1 kHz [12]. Moreover, the DUT's change of the SNR is captured by the *noise figure* F [14]:

$$F_{\text{dB}} = \text{SNR}_{\text{dB,test}} - \text{SNR}_{\text{dB,res}}. \quad (2.49)$$

The next indicator is the *maximum output level* of the DUT which provides sufficient quality [10]. This level is bounded due to the power supply of the DUT, but the highest possible output level can lead to *clipping*. Clipping cuts off the amplitude of the DUT's input signal in the time domain when a certain threshold of gain is exceeded. For example, a sine wave is clipped on

³A ratio in decibels is often called *level*. See more about the definition of decibels in appendix A.

the left in fig. 2.12. This phenomenon leads to a *harmonic distortion* which generates multiples of the input's fundamental (i.e., the harmonics), as shown on the right in fig. 2.12. Hence, the waveshape of the sine wave changed and the resulting signal is representable as a Fourier Series. If clipping appears, the output power will not increase as the input power increases [14].

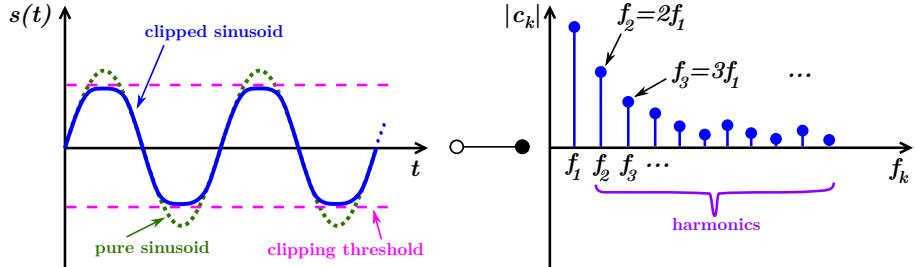


Figure 2.12.: Clipping of a sine wave leads to a harmonic distortion.

Therefore, the name *gain compression* is in common use, too. A related measurement is the *linear dynamic range* (LDR). This is the range where a system acts approximately linear and deterministic, i.e., an amplifier performs the following equation [14]:

$$P_{\text{out}, \text{dBm}} = G_{\text{dB}} + P_{\text{in}, \text{dBm}} \quad \text{or} \quad P_{\text{out}} = GP_{\text{in}} \quad (G \text{ as power ratio}). \quad (2.50)$$

The LDR's lower bound is the output level when the analog noise level is equal to the input level and the LDR's upper bound is the output level when gain compression occurs [14]. This dynamic is visible in fig. 2.13 where the output power in dBm is plotted versus the input power in dBm⁴.

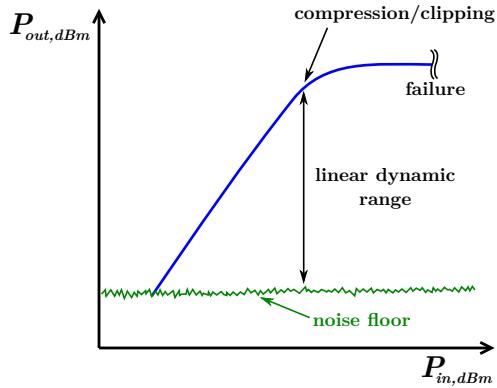


Figure 2.13.: Linear dynamic range (LDR) of an amplifier (based on [14, Figure 10.1]).

Furthermore, there is an indicator for the dimensions of harmonic distortions. It is the *total harmonic distortion* (THD) which is the ratio of the harmonics' root mean square (RMS)⁵ to the fundamental's RMS or its magnitude, respectively [15]. Moreover, it is equivalently defined as the square root of the harmonics' average power \bar{P}_H over the fundamental's average power

⁴See eq. (A.4) of the appendix for the definition of dBm.

⁵See eq. (A.12) of the appendix for the definition of the RMS.

\bar{P}_H . We express the THD of a periodic and continuous signal by applying the Fourier Series [4]:

$$\text{THD} = \sqrt{\frac{\bar{P}_H}{\bar{P}_F}} = \frac{\text{RMS}_H}{\text{RMS}_F} = \frac{\sqrt{\frac{1}{K} \sum_{k=2}^K |c_k|^2}}{\sqrt{\frac{1}{K} \sum_{k=1}^1 |c_k|^2}} = \frac{\sqrt{\sum_{k=2}^K |c_k|^2}}{|c_1|}. \quad (2.51)$$

There are another definitions of the THD, but this one is recommended [15], [16].

Digital Audio Devices

Digital audio devices, e.g. sound cards, have got additional quality indicators, compared to analog devices. These new requirements exist due to the A/D or the D/A conversion. As shown in section 2.1.1, the resolution of the time or frequency depends on the used sampling frequency f_s and the signal length N . Considering eq. (2.21) and (2.22), the resolution increases as f_s or N increases. The sample frequency must satisfy eq. (2.14) that the A/D or D/A process is correct. Common values of f_s are 8 kHz for speech transmission, 44.1 kHz for CDs, 48 kHz for default studio production, and 96 kHz or 196 kHz for high-quality studio production [12].

Then, the resolution of the vertical axis, i.e., the resolution of the amplitudes, are also limited in a digital system. These devices are capable to store finite large or small numbers. Thus, the continuous amplitudes of the discrete-time signal $s_{DT}(n)$ are mapped to discrete amplitudes after the sampling process [12]. This process is called *quantization*. The used amount of bits to display the amplitudes are denoted by b . With this, they are mapped to the set

$$\mathcal{Q} = [-2^{b-1}; 2^{b-1} - 1] \cap \mathbb{Z}, \quad b \in \mathbb{N}, \quad (2.52)$$

by rounding down to the next integer [5]:

$$s_q(n) = \lfloor s_{DT}(n) \rfloor, \quad s_q(n) \mapsto \mathcal{Q}. \quad (2.53)$$

Now, $s_q(n)$ is the quantized version of $s_{DT}(n)$. This mapping leads to rounding errors which are called the *quantization error*. These errors result in *quantization noise* in the DFT spectrum which looks like analog noise [4]. Hence, we define a *signal-to-quantization noise ratio* (SQNR)

$$\text{SQNR}_{\text{dB}} = 10 \log_{10} \left(\frac{P_{\text{sig}}}{P_{\text{qn}}} \right) = 1.76 + 6.02b \quad (2.54)$$

where P_{qn} is the power of the quantization noise and which depends only on the number of bits [6]. This equation is valid if the signal is sinusoidal and reaches the minimum and maximum value of the set \mathcal{Q} . If a signal reaches the set's limits, we say that the signal has got full-scale. The *effective dynamic range* of a digital system differs from an analog one. The range of the dynamic is the SQNR [6]. A value greater than the maximum of \mathcal{Q} is not possible. This leads to the upper bound. The lower bound is the quantization noise. Common values of b are 16 bit ($\text{SQNR}_{\text{dB}} = 98.08 \text{ dB}$) for CDs and 20 bit ($\text{SQNR}_{\text{dB}} = 122.16 \text{ dB}$) or 24 bit ($\text{SQNR}_{\text{dB}} = 146.24 \text{ dB}$) for studio productions [12].

Practical THD in Digital and Analog Systems

Harmonic distortions and noise appear simultaneously in a practical audio system. Hence, we introduce the *total harmonic distortion plus noise* (THD+N). It still is the ratio of the harmonics to the fundamental, as the THD in eq. (2.51), but they are both superimposed by noise [10]. Here, we define noise as the sum of analog noise, undesired signals, and quantization noise for a digital system. The quantization noise is dispensable if the system is analog. If the THD+N is low, it is a good overview that the DUT's quality is satisfactory because the harmonic distortion *and* the noise is low. For example, a typical consumer preamplifier has got a THD+N between 0.01 % and 0.1 %, or a typical CD player has got values from 0.0015 % to 0.01 % [10].

3. Implementation and System Concept

The following measurement application for the audio quality is based on GNU Radio. This is an open-source development toolkit for *Software Defined Radio* (SDR) [17]. GNU Radio is written in C++ and provides especially Digital Signal Processing (DSP) functions for Digital Communication. Also, it is possible to integrate these C++ functions in Python due to the *Simplified Wrapper and Interface Generator* (SWIG) [18]. Hence, there is a Python-based flow chart development environment which simplifies the usage of GNU Radio. It is called the GNU Radio Companion. It is capable to simulate SDR applications and it is able to access the PC's ports.

GNU Radio Companion offers beside DSP blocks also blocks for graphical user interfaces (GUI). Furthermore, it is possible to program own blocks in Python which makes the development versatile. As a result, software packages like NumPy or SciPy can be used [19], [20]. The flow chart concept is shown in fig. 3.1. There is a signal source which creates the sine wave $x(n)$.

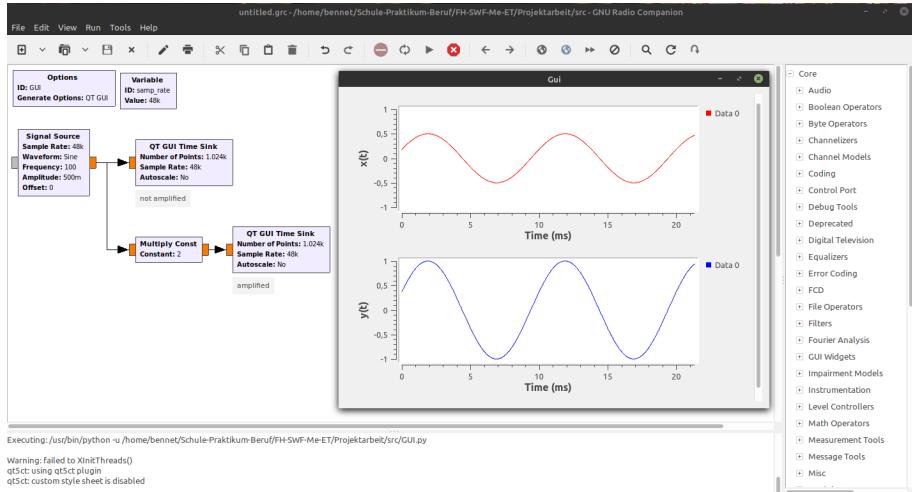


Figure 3.1.: Demonstration of GNU Radio Companion.

This source is connected to time domain scope which is the *QT GUI Time Sink* block. This element generates the scope with the red line in the window of the executed application. Also, there is the *Multiply Const* block connected to the signal source. This multiplies the signal $x(n)$ by 2 and is attached to a time scope, as well. Thus, the gain is visible as the blue sinusoid in the window of the executed flow chart. Moreover, GNU Radio Companion generates a file with Python source code which can be executed without the usage of the development environment. But, Python and GNU Radio must be installed to launch the program. Here, GNU Radio Companion v3.8.5.0-5-g982205bd and Python 3.6.9 are used on a Linux Mint 19.3 operating system.

The measurement concept of the application is visualized in fig. 3.2. The PC runs the GNU Radio application and its GUI elements are controlled by the PC's periphery. Then, the software

has access to the sound card which connects the digital domain with the analog domain and vice versa. The output path of the sound card is a D/A converter with an audio jack, e.g., a 3.5 mm phone connector. The input path has got the same structure, just there is an A/D converter. Furthermore, the output of the sound card is connected by cable to the input jack of the DUT. The test object's output jack is connected to the sound card's input jack. These links lead to the black-box testing concept of the application. Then, GNU Radio generates a test signal that is fed into the DUT which applies its transfer function and changes the test signal. The resulting output is measured by the sound card and analyzed by the DSP algorithms of the main application. This signal processing leads to the indicators of the sound quality, as mentioned in section 2.3. These indicators, the test signal, and the input signal are displayed in the GUI.

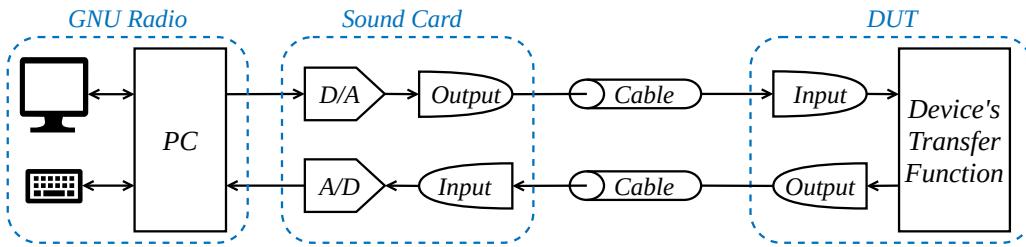


Figure 3.2.: The measurement concept of the application. The device under test (DUT) is treated as a black-box.

3.1. Graphical User Interface

The GUI is shown in fig. 3.3. In the middle, there is the upper scope which displays the test signal's time domain. The vertical axis is scaled that 1 is equal to full-scale. The scope below is the magnitude of the test signal's single-sided FFT spectrum. The frequency domain is displayed as a relative level

$$|S_{\text{dB}}(k)| = 20 \log_{10} \left(\left| \frac{S(k)}{N} \right| / O(k) \right), \text{ unit dB}, \quad (3.1)$$

where $S(k)$ are the complex FFT samples, N is the FFT length, and $O(k) = 1 \forall k \in \mathbb{Z}$ is the (pseudo) reference sequence. The scaling by N cancels the DFT's gain which is shown in eq. (2.17). Thus, a sine wave with an amplitude 1 will have a level of -6 dB in the spectrum if the display via eq. (3.1) is used, no leakage occurs and therefore a rectangular window is applied. It is not 0 dB because the sinusoid's amplitude is divided by 2, according to eq. (2.17). This means that the signal power is split equally into the positive and negative frequency components.

The right scopes show the same domains, but from the input signal. The menu to the right of each time scope offers the ability to zoom in and out (axes), and to stop the image of the streaming audio by setting a trigger. Another option is to autoscale or to freeze the view of the signal (stop). The menu to the right of a FFT scope has got the zoom function (axis options), the trigger, and the freeze function (stop), as well. The trace options are able to hold the maximum or the minimum of the recorded spectrum and it can average the signal (avg). If the avg slider is pushed to the left, the averaging increases and the curve will fluctuate less. The number 2048 represents the FFT length of the FFT menu. It can be changed using the combo box. Below

that, there is the box with the name *Kaiser* which is the Fourier Transform's window due to the leakage. The Kaiser window has got a $\beta = 6.76$ here, i.e., $\delta_{st} = 70$ dB.

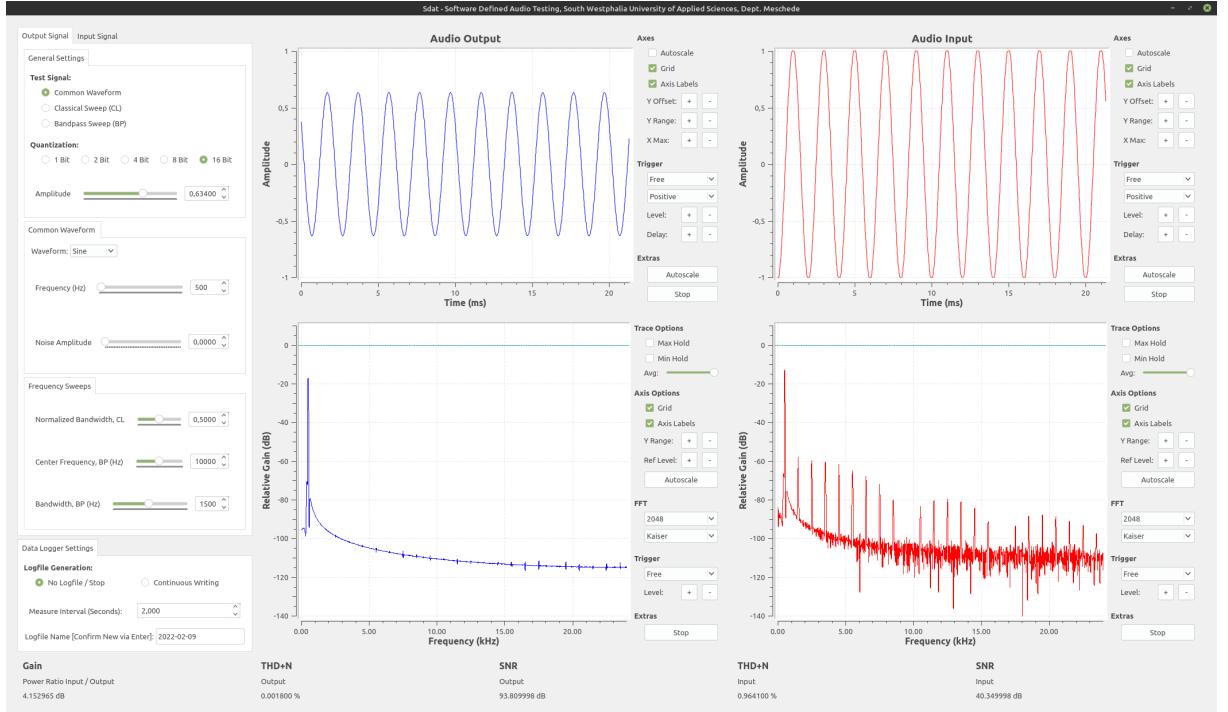


Figure 3.3.: This is the GUI to measure the quality indicators. On the left hand side, there are the input, output and data logger settings. The blue signals displays the output's test signal in the middle. Next to this, there are the scopes of the input signal. The footer shows the gain, the THD+N, and the SNR.

In the footer, there are also indicators of the quality measurement. They are the gain, the THD+N and the SNR. Here, the gain is the power ratio of the input's power to the output's power in decibel. The THD+N and the SNR are displayed twice because one value is for the output and the other for the input.

On the left of fig. 3.3, there is a box with an *output signal* and *input signal* tab. This is the menu to control the parameters of the output and input. All possible tabs are visualized in fig. 3.4. The output settings are in the partial picture (a) of this figure. There, we can choose between 3 test signals, the output's quantization, and the test signal amplitude A_{test} under the box *general settings*. The amplitude is variable and has the range 0 to 1 where 1 is equal to full-scale.

The next box has got the name *common waveform* and adjusts the properties of its namesake. The combo box *waveform* is the GUI element to change the shape whereby the generation of a periodic sine, square, triangle, or saw tooth oscillation is possible. Then, the test signal's frequency is adjustable by the *frequency* slider in the range $f_{test} \in [20.0\text{ Hz}; 20.0\text{ kHz}]$. The test signal may superimposed by white noise whose power is selectable. Here, the definition of the *noise amplitude* A_{wn} is that the noise power and the test signal power are equal if $A_{wn} = A_{test} = 1$. Thus, the SNR is equal to 0 dB. Or, if $A_{test} = 1$ and $A_{wn} = 0.5$, then the $\text{SNR}_{\text{dB}} = 6$ dB which represents an *amplitude ratio* of 2. The name noise amplitude was chosen because the unit of the number has got an amplitude unit and GNU Radio uses internally the same name. It is known that the real amplitude of the white noise fluctuates randomly.



Figure 3.4.: All possible tabs of the setting menu (a, b), the data logger settings (c), and an example CSV-file (d, e).

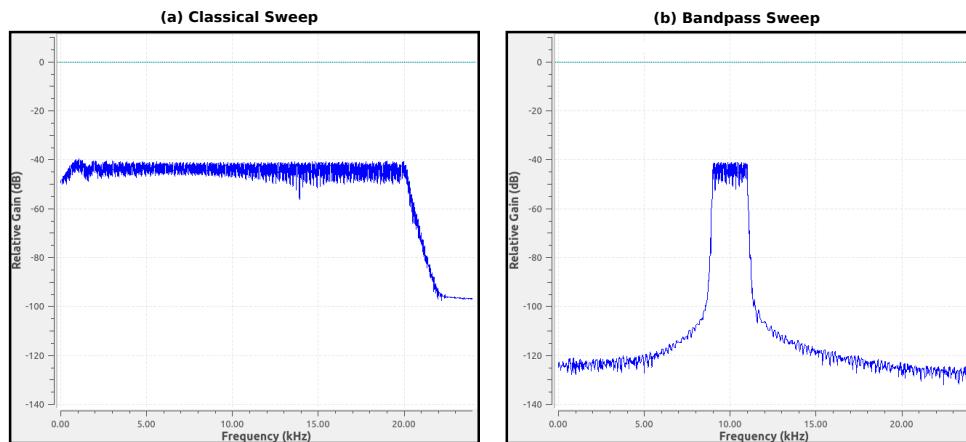


Figure 3.5.: A classical sweep with $B_{sw} = 20.0$ kHz (a) and a bandpass sweep with $f_{cen} = 10$ kHz and $B_{bpsw} = 2$ kHz (b).

If we select the *classical sweep (CL)* option in fig. 3.4 (a), the application will generate a test signal which is a common sweep, as shown in fig. 3.5 (a). The bandwidth of the sweep is

$$B_{\text{sw}} = \tilde{B}_{\text{nz}} 20 \text{ kHz} \quad (3.2)$$

where \tilde{B}_{nz} is the value of the *normalized bandwidth*, *CL* slider under the *frequency sweeps* box. The range of the bandwidth starts with 200 Hz ($\tilde{B}_{\text{nz}} = 0.01$) and ends at 20 kHz ($\tilde{B}_{\text{nz}} = 1$). The other sliders are only for the bandpass sweep in the *frequency sweeps* box.

If we select the *bandpass sweep (BP)* radio button, the system will create a box function with the center frequency f_{cen} and the bandwidth B_{bpsw} in the spectrum. The sweep's lower cutoff frequency is defined as

$$f_{c,\text{lo}} = f_{\text{cen}} - \frac{1}{2} B_{\text{bpsw}} \quad (3.3)$$

and the upper cutoff can be written as

$$f_{c,\text{up}} = f_{\text{cen}} + \frac{1}{2} B_{\text{bpsw}}. \quad (3.4)$$

B_{bpsw} is adjustable by the *bandwidth*, *BP* slider and f_{cen} by the *center frequency*, *BP* slider. The center frequency has got the range $f_{\text{cen}} \in [2 \text{ kHz}; 18.5 \text{ kHz}]$ and the bandwidth $B_{\text{bpsw}} \in [0.1 \text{ kHz}; 3.0 \text{ kHz}]$. These values were chosen to generate a narrowbanded test signal. This is advantageous if just a part of the audio range has to be measured. For instance, a tweeter does not play well low-end frequencies and thus does not require these spectral components as a part of the test signal.

The input settings are shown in fig. 3.4 (b). They contain a filter with gain and the amount of used bits for the quantization. All of these things are only applied to the input signal and not to the output. The input filter can be either a lowpass or a highpass, or just an amplifier (*gain only* option). The gain, the stopband ripple δ_{st} , and the cutoff frequency f_c are changeable by the sliders in the *active filter* box. The gain varies from $G_{\text{in}} = -60 \text{ dB}$ up to $G_{\text{in}} = 60 \text{ dB}$ and the cutoff frequency has got the range $f_c \in [0.5 \text{ kHz}; 20.0 \text{ kHz}]$. The transmission width is 0.5 kHz hence the lower bound results. The theoretical upper bound is $f_{c,\text{max}} = 48 \text{ kHz}/2$, according to eq. (2.14), if the internal sample rate $f_{s,\text{in}} = 48 \text{ kHz}$ is concerned. Thus, the 20.0 kHz cutoff frequency is below the technical limit and still contains the audio range's maximum. This input filter has got a FIR architecture and computes its coefficients on the fly when the cutoff frequency or the attenuation changes. Also, the filter's passband ripple can be varied by the *max. stopband ripple* radio buttons. This filter uses a Kaiser window and therefore the ripples results from the betas in table 2.1 on page 10.

The last menu contains the *data logger settings* which is shown in fig. 3.4 (c). If the radio button *continuous writing* is selected, the application will save the numeric quality indicators in a *comma-separated values* (CSV) file. The commas separate a table column and a newline is equal to a new row of that table. An example for the CSV-file is displayed in fig. 3.4 (d) where one dataset contains the test signal's amplitude (**Asig**), the noise amplitude (**Anoise**), the test frequency (**ftest**), the gain, the output's SNR (**SNRout**), its THD+N (**THDout**), the input's SNR (**SNRin**) and its THD+N (**THDin**). The test frequency has got the unit Hz, the gain and the SNRs are in dB, and the THD+Ns are in %. The representation of the CSV-file as a table is shown in fig. 3.4 (e). Furthermore, the *measurement interval* sets the period when the

measurement results are stored. This time varies from 0.1 s up to 30.0 s. The file will be saved in the directory `sdat_measurements` which is in the same path as the folder of the GNU Radio installation. For example, the path is `/home/userName/sdat_measurements` in Linux Mint 19.3 because the GNU Radio files are stored in `/home/userName/gnuradio`. The default file name is the current date in the format `YYYY-MM-DD.csv`. The name is adjustable by the text field *logfile name*. A new name has to be confirmed via pressing the enter key.

In general, the GUI development is fast to apply because the mentioned elements like the combo boxes, sliders, etc. are already implemented by GNU Radio Companion. The big disadvantage of these implemented elements is that they are not much customizable. For example, if the output signal tab is selected in fig. 3.4 (a), there is no variable which shows the tab's change. Or, a text field triggers only an event if the enter key is pressed. But, the decimal places of the indicator displays are more problematic. Their digits can not be set which may suggest too high measuring accuracy. In general, it is assumed that a decibel value is displayed accurately by maximum 2 digits after decimal point. The only way to change this issue is to modify the open source code and build an own version of GNU Radio, but this is not the goal of this Project Work.

3.2. Program Structure and Algorithms

This section deals with the system concept of the measurement application. First, we discuss

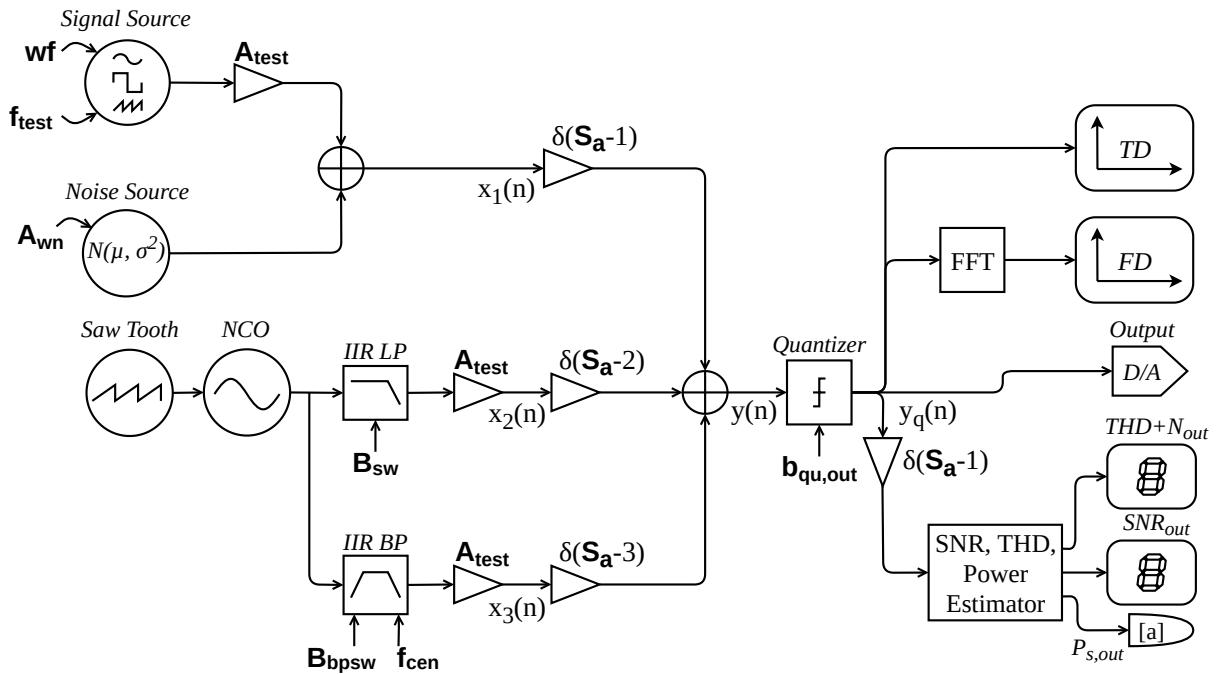


Figure 3.6.: Signal Processing of the measurement system's output path.

the generation of the test signals which were mentioned before. Therefore, the flow chart of the output path is shown in fig. 3.6. Generally speaking, the circles are signal sources, the triangles are multipliers, the circles with the crosses are adders, the rounded rectangles are display elements of the GUI, and the other rectangles are special functions which are explained later. Then, the characters in bold are input elements of the GUI, e.g., a slider value or a combo

box entry. The cone, with the $[a]$, is a connector to the system's input path in fig. 3.11.

In fig. 3.6, there is the signal source for the common waveforms. Its input wf is the waveform of the output setting's combo box in fig. 3.4 (a), i.e., $wf \in \{\text{sine, square, triangle, saw tooth}\}$. f_{test} sets the frequency of the test signal. Then, the waveform is generated with the amplitude 1 and is scaled by the multiplier $\mathcal{A}_{\text{test}}$. The noise source, below, produces gaussian white noise which is added to the common waveform. The \mathcal{A}_{wn} is the noise amplitude and sets the noise power. Its name and function was defined in section 3.1. The following multiplication by the Kronecker delta $\delta(S_a - 1)$ acts as the 1st path of a switch and contains the value of the radio button S_a which selects the test signal:

$$S_a = \begin{cases} 1 : & \text{common waveform } x_1(n) \\ 2 : & \text{classical sweep } x_2(n) \\ 3 : & \text{bandpass sweep } x_3(n) \end{cases}. \quad (3.5)$$

We extract the Kroenecker deltas from the output path to show their switch function. This is shown in fig. 3.7. In this picture, the output $y(n)$ will be equal $x_1(n)$ if the switch's position S_a is equal 1, or $y(n) = x_2(n)$ if $S_a = 2$, or $y(n) = x_3(n)$ if $S_a = 3$. Considering the properties

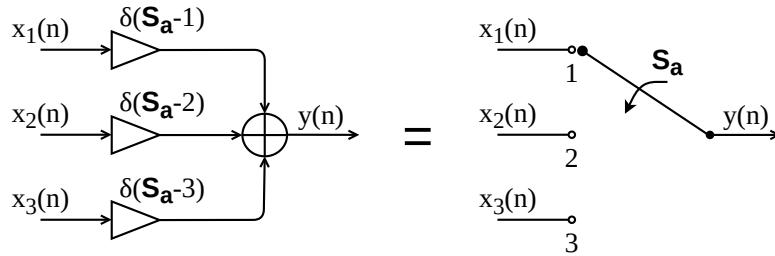


Figure 3.7.: Kroenecker delta used as a switch where $S_a \in \{1, 2, 3\}$ is the switch's position.

of the Kronecker delta in eq. (2.18), we obtain the desired switch function by adding all of the three paths which are multiplied by its corresponding delta:

$$y(n) = x_1(n)\delta(S_a - 1) + x_2(n)\delta(S_a - 2) + x_3(n)\delta(S_a - 3) = \begin{cases} x_1(n) : & S_a = 1 \\ x_2(n) : & S_a = 2 \\ x_3(n) : & S_a = 3 \\ 0 : & \text{otherwise} \end{cases}. \quad (3.6)$$

In GNU Radio, we create a custom block which offers the multiplication by the Kronecker delta for the m -th path of the switch. It uses the heaviside function $\varepsilon(S_a - m)$ for the implementation:

$$\delta(S_a - m) = \varepsilon(S_a - m) - \varepsilon(S_a - (m + 1)), \quad \varepsilon(S_a - m) = \begin{cases} 1 : & S_a \geq m \\ 0 : & S_a < m \end{cases}, \quad m \in \mathbb{Z}. \quad (3.7)$$

A custom block of a switch by common if-else statements leads to strong lagging of the streaming test signals therefore the Kroencker switch is used instead.

Next, the saw tooth source generates a periodic saw tooth of the frequency $f_{\text{saw}} = 2\pi 10 \text{ Hz}$ and with an amplitude $\mathcal{A}_{\text{saw}} = 1$. Then, its output is connected to the *Numerically Controlled*

Oscillator (NCO). This component creates a sine wave with the amplitude 1. Its frequency $f_{\text{nco,out}}$ depends on the amplitude $\mathcal{A}_{\text{nco,in}}$ of the NCO input. Here, the NCO has got a sensitivity of 22 kHz per full-scale amplitude, i.e., the sinusoid's frequency is

$$f_{\text{nco,out}} = 22 \text{ kHz } \mathcal{A}_{\text{nco,in}}, \quad \mathcal{A}_{\text{nco,in}} \in [0; 1]. \quad (3.8)$$

The saw tooth's value of the vertical axis changes linear between 0 and 1. Thus, the NCO input value varies between 0 and 1, too. This leads to a frequency sweep from 0 Hz to 22 kHz at the output of the NCO. If we filter this sweep by the IIR lowpass (LP), we obtain the classical sweep and if we use the IIR bandpass (BP), we get the bandpass sweep. The sensitivity of 22 kHz, instead of 20 kHz, is necessary because the NCO has got a lowpass characteristic itself. That is, the NCO has got approximately a transmission width of 1.0 kHz. We desire a constant amplitude of the sweep hence we choose the sensitivity higher than the audio range to escape the transmission width. The other input values of the filters set their bandwidth (B_{sw} or B_{bpsw}) and the center frequency f_{cen} of the BP. These parameters results from the GUI's sliders which are shown in fig. 3.4 (a). Both filters has got a maximum passband ripple of 0.2 dB, a maximum stopband ripple of 60 dB, and a Chebyshev I characteristic. A Chebyshev I design has got a ripple in the stopband and maintain monotonicity in the passband [7]. IIR filters were chosen because they require fewer multiplications and less memory than a FIR filter with the same requirements [6]. The filter order of the lowpass is 7 and 11 of the bandpass. The order of the BP is higher to achieve a steeper transmission width. We prevent an unstable filter due to coefficient quantization by cascading second order sections (SOS)¹ [8]:

$$H_{\text{cas,sos}}(z) = b_0 \prod_{k=1}^{L_{\text{sos}}} H_{\text{sos},k}(z) = b_0 \prod_{k=1}^{L_{\text{sos}}} \frac{1 + \beta_{1k} z^{-1} + \beta_{2k} z^{-2}}{1 - \alpha_{1k} z^{-1} - \alpha_{2k} z^{-2}}. \quad (3.9)$$

Here, the L_{sos} is the number of SOSs, the β s and α s are the SOS's new coefficients. If we transfer a direct implemented IIR filter, as shown in fig. 2.9 or in eq. (2.41), into a cascade form, we design the filter with common (software) tools and then calculate its roots. We use eq. (2.42) to compute M zeros z_m and L poles p_i . Then, we take 2 zeros and 2 poles to build an 2nd order filter [7], [8]:

$$H_{\text{sos},k}(z) = \frac{(1 - z_{1k} z^{-1})(1 - z_{2k} z^{-1})}{(1 - p_{1k} z^{-1})(1 - p_{2k} z^{-1})} = \underbrace{\frac{1 - (z_{1k} + z_{2k}) z^{-1} + (z_{1k} z_{2k}) z^{-2}}{1 - (p_{1k} + p_{2k}) z^{-1} + (p_{1k} p_{2k}) z^{-2}}}_{\alpha_{1k} \quad \alpha_{2k}}. \quad (3.10)$$

The assumption that $M \leq L$, and the pairing of the roots, leads to $L_{\text{sos}} = \lfloor (L+1)/2 \rfloor$ sections [8]. If the coefficients $b_m, a_i \in \mathbb{R}$, each complex root has got a complex conjugated counterpart hence the pair of z_{1k} and $z_{2k} = z_{1k}^*$, or p_{1k} and $p_{2k} = p_{1k}^*$ are common [7]. Then, $\beta_{1k} = -2\text{Re}\{z_{1k}\}$ and $\beta_{2k} = |z_{1k}|^2$. This holds analogously for the poles. The cascade structure is not extremely sensitive to quantization errors and uses a minimum number of multiplications and unit delays if the structure in fig. 3.8 is used [8]. Further simplifications are obtained if the number of roots is odd. Then, one β_{2k} or α_{2k} will be zero [8]. Hence, the order of the filters is chosen odd.

¹A second order section is a 2nd order IIR filter.

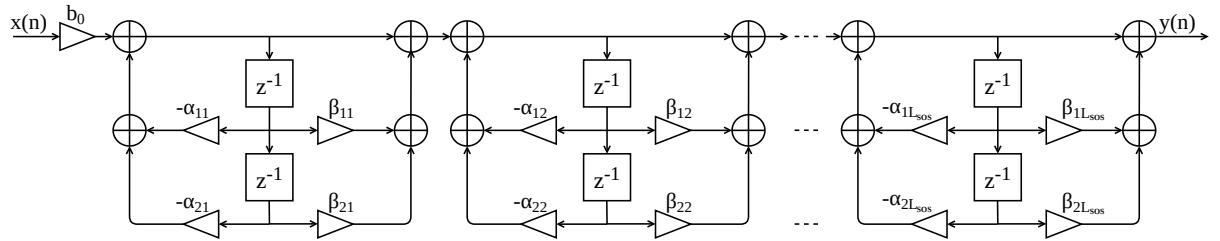


Figure 3.8.: An IIR filter of cascaded second order sections which provide robustness against quantization errors (based on [7, Figure 5.8]).

Now, we return to the flow chart of the output in fig. 3.6 where the output of the Kroecker switch $y(n)$ is one of the 3 test signals, as mentioned before. Then, $y(n)$ is quantized by 1, 2, 4, 8, or 16 bits. This bit number $b_{\text{qu,in}}$ is set by the radio buttons of the GUI in fig. 3.4 (a). After that, the quantized test signal $y_q(n)$ is simultaneously processed by several blocks. The TD block displays the time domain of $y_p(n)$. The FFT block windows the signal and calculates its magnitude spectrum in decibel. The visualization of the magnitude versus the frequency is done by the FD block. Then, the test signal is fed into the D/A converter that the sound card's output provides a time-continuous audio signal (i.e., a time dependent voltage). The next Kroecker delta multiplier ensures that the following SNR, THD and power estimator block is only active when the output is a common waveform. As the name indicates, this block estimates the THD+N_{out} in percent, the SNR_{dB,out} and the signal power $P_{s,\text{out}}$ of the test signal. The THD+N_{out} and the SNR_{dB,out} are numerically displayed by the GUI. This is implied by the special shaped 8 which might be recognized from digital clocks, for example. The signal power is connected to the input path and will be further processed in this part of the system (see section 3.2.3).

3.2.1. SNR, THD and Power Estimator

This estimator is not provided by GNU Radio and therefore a custom block. The flow chart of the estimator is shown in fig. 3.9. First, the quantized sequence $y_q(n)$ is windowed by a

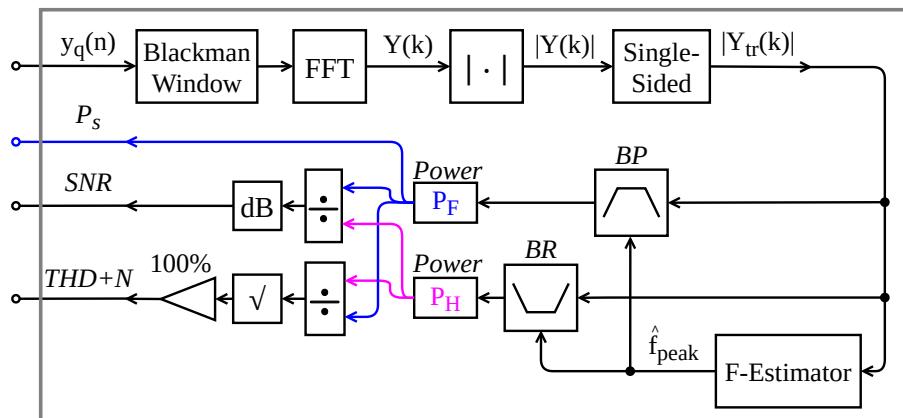


Figure 3.9.: Structure of the SNR, THD and Power Estimator with one input $y_q(n)$ and three outputs P_s , SNR_{dB,out}, and THD+N.

Blackman window to reduce leakage. Then, the FFT is performed with a length $N = 2^{14}$. This high resolution is required due to the special filters which are explained later. The next step is to

compute the magnitude spectrum $|Y(k)|$ which is performed by the $|\cdot|$ block. The magnitudes are necessary because the fundamental estimator (*F-Estimator*) and the power calculator (*Power* blocks) uses this kind of samples. Then, the *Single-Sided* block truncates the samples from $N/2$ up to N due to their redundancy, as mentioned in section 2.1.2. We thus reduce the signal length to $N_{\text{tr}} = N/2 = 2^{13}$ but we discard the half of the signal power, too. This will not effect the results in computing the SNR or THD+N because they are ratios where the same constant factors are canceled, as shown in the following equation:

$$\text{THD} = \sqrt{\frac{\frac{1}{2}P_{\text{H}}}{\frac{1}{2}P_{\text{F}}}} = \sqrt{\frac{P_{\text{H}}}{P_{\text{F}}}}. \quad (3.11)$$

For the same reason, we do not cancel the factor N from the FFT samples, as mentioned in eq. (2.17). The ratio cancels the additional power offset by N , too. This saves a multiplication by $1/N$ with all 2^{13} samples.

Next, the truncated magnitude spectrum $|Y_{\text{tr}}(k)|$ is connected to the *F-Estimator*. This element estimates the fundamental's frequency f_{peak} in $|Y_{\text{tr}}(k)|$. For this purpose, the following estimation algorithm is used, but it will fail if the fundamental has not got the highest amplitude in the spectrum [6]:

1. Determine the index k_{max} of the peak.
2. The index will not belong to the real fundamental due to leakage. Therefore, the following equation computes a correction factor \mathcal{K} which requires a Blackman window:

$$\mathcal{K} = \frac{1.75 (|Y_{\text{tr}}(k_{\text{max}} + 1)| - |Y_{\text{tr}}(k_{\text{max}} - 1)|)}{|Y_{\text{tr}}(k_{\text{max}} + 1)| + |Y_{\text{tr}}(k_{\text{max}})| + |Y_{\text{tr}}(k_{\text{max}} - 1)|}. \quad (3.12)$$

3. The estimated index of the fundamental is $\hat{k}_{\text{peak}} = k_{\text{max}} + \mathcal{K}$.
4. Considering eq. (2.22), the corresponding peak frequency is

$$\hat{f}_{\text{peak}} = \hat{k}_{\text{peak}} \frac{f_s}{N_{\text{tr}}} = \hat{k}_{\text{peak}} \frac{f_s}{N/2}. \quad (3.13)$$

This method will also fail if the majority of the signal's energy does not lie within a bin and if the fundamental's bins are substantially contaminated by leakage from other frequencies [6]. A sine wave satisfies the mentioned conditions as a test signal if the noise peaks are not considerably bigger than the sinusoid's amplitude.

Now, the center frequency of the bandpass (BP) and the bandreject (BR) filter is set by \hat{f}_{peak} . Thus, the BP suppresses the harmonics and passes the fundamental, and the BR vice versa. This leads to a separation of these spectral components in $|Y_{\text{tr},\text{BP}}(k)|$ (fundamental) and $|Y_{\text{tr},\text{BR}}(k)|$ (harmonics). Hence, the fundamental's power P_{F} and the harmonics' power P_{H} are computable by

$$P_{\text{F}} = \frac{1}{N_{\text{tr}}} \sum_{k=0}^{N_{\text{tr}}-1} |Y_{\text{tr},\text{BP}}(k)|^2 \quad \text{and} \quad P_{\text{H}} = \frac{1}{N_{\text{tr}}} \sum_{k=0}^{N_{\text{tr}}-1} |Y_{\text{tr},\text{BR}}(k)|^2. \quad (3.14)$$

The rest of the flow chart shows the SNR and THD+N calculation, i.e., eq. (2.48) and eq. (2.51) are implemented. Furthermore, both filters are not IIR or FIR filters because their sophisticated requirements causes several problems. If we determine the SQNR, the bandreject needs a narrow

bandwidth to suppress only the fundamental and nothing else. In addition, the suppression requires a high attenuation to eliminate the whole energy of the fundamental. The bandpass has got similar requirements, i.e., a narrow bandwidth to pass just the fundamental and a strong attenuation of the harmonics. A FIR filter satisfies these conditions with a stopband ripple $\delta_{st} = -81$ dB (i.e., Kaiser window $\beta = 8$), a transmission width of $\Delta f = 20$ Hz, and a bandwidth $B = 550$ Hz. With that filter, we compute a $SQNR_{dB} \approx 97$ dB at 16 bit quantization which is close to the theoretical value of 98.08 dB (cf. eq. (2.54)). Considering eq. (2.39), the filter order is

$$M = \left\lceil \frac{|-81| - 8}{2.285 \left| \frac{2\pi}{48 \text{ kHz}} 20 \text{ Hz} \right|} + 1 \right\rceil = 12\,204 \quad (3.15)$$

which is not practical. If we assume a constant phase, the group delay through the filter is $\tau_g = 12\,204 / (2 \times 48 \text{ kHz}) = 0.127$ s, according to eq. (2.34). This and the computing time of the coefficients, when the fundamental's frequency changes, slows down the response time of the application and the signal representation of the 4 scopes. An similar IIR bandpass or bandreject causes stability problems due to the narrow bandwidth and is thus no alternative. Hence, the next algorithm is presented to imitate an BP or BR behavior:

1. We calculate the upper cutoff frequency $f_{c,up}$ and lower cutoff $f_{c,lo}$ where 250 Hz is the half of the bandwidth around the fundamental:

$$f_{c,up} = f_{peak} + 250 \text{ Hz} < \frac{f_s}{2}, \quad f_{c,lo} = \begin{cases} f_{peak} - 250 \text{ Hz} : & f_{peak} \geq 250 \text{ Hz} \\ 0 : & \text{otherwise} \end{cases}. \quad (3.16)$$

The first condition $f_{c,up} < f_s/2$ results from eq. (2.14) and $f_{c,lo} = 0$ to prevent negative cutoff frequencies.

2. The corresponding bins of the upper and lower cutoff are computable using eq. (2.22):

$$k_{c,up} = \left\lfloor N_{tr} \frac{f_{c,lo}}{f_s} \right\rfloor, \quad k_{c,lo} = \left\lfloor N_{tr} \frac{f_{c,lo}}{f_s} \right\rfloor. \quad (3.17)$$

3. We imitate the bandreject by setting all samples to zero from $k_{c,lo}$ up to $k_{c,up}$:

$$|Y_{tr,BR}(k)| = \begin{cases} 0 : & k \in [k_{c,lo}; k_{c,up}] \\ |Y_{tr}(k)| : & \text{otherwise} \end{cases}. \quad (3.18)$$

4. The imitated bandpass is the inverse of the bandreject:

$$|Y_{tr,BP}(k)| = \begin{cases} |Y_{tr}(k)| : & k \in (k_{c,lo}; k_{c,up}) \\ 0 : & \text{otherwise} \end{cases}. \quad (3.19)$$

3.2.2. Verification of the SNR's Measurement Accuracy

First, we measure the full-scaled SQNR to judge the accuracy of the SNR, THD and Power estimator. We use a sine wave of $f_{test} = 1$ kHz as a test signal. Its amplitude is 1 that we can apply eq. (2.54) as a reference. Table 3.1 shows the measurement results. The $SQNR_{dB,theo}$ is the mentioned reference value and the $SQNR_{dB,meas}$ is the measured signal-to-quantization

Table 3.1.: Measurement results of the SQNR testing, having a sample size K and a significance level α .

b_{qu} in bit	Waveform	f_{test} in kHz	$\mathcal{A}_{\text{test}}$	$\text{SQNR}_{\text{dB,theo}}$ in dB	$\text{SQNR}_{\text{dB,meas}}$ in dB	Δ in dB	K	α
1	sine	1	1	7.78	10.22 ± 0.00	-2.44	100	5 %
2	sine	1	1	13.80	15.09 ± 0.00	-1.29	100	5 %
4	sine	1	1	25.84	25.26 ± 0.19	0.58	100	5 %
8	sine	1	1	49.92	50.08 ± 0.09	-0.16	100	5 %
16	sine	1	1	98.08	97.80 ± 0.09	0.28	100	5 %

noise ratio. We use the data logger to store the results. The $\text{SQNR}_{\text{dB,meas}}$ is the **SNRout** column of the CSV-file. This column is denoted by $v(n)$ and is a sequence $v(n) = (v_0 \ v_1 \ v_2 \ \dots \ v_K)$ because multiple measurement values are saved. Thereby, K is the sample size. Some SQNR numbers are fluctuating and therefore multiple records are necessary to display its *arithmetic mean* \bar{v} and its confidence interval $[v_{\text{lo}}; v_{\text{up}}]$. Here, v_{lo} is the interval's lower bound and v_{up} is the upper bound. We compute the confidence interval of the true but unknown SQNR via the average \bar{v} and the *measurement error* Δv [21]:

$$[v_{\text{lo}}; v_{\text{up}}] = \bar{v} \pm \Delta v = \left(\frac{1}{K} \sum_{m=1}^K v_m \right) \pm \left(t_{1-\frac{\alpha}{2}}(K-1) \frac{\text{sd}_v}{\sqrt{K}} \right) \quad (3.20)$$

$$= \left(\frac{1}{K} \sum_{m=1}^K v_m \right) \pm \left(t_{1-\frac{\alpha}{2}}(K-1) \sqrt{\left[\frac{1}{K-1} \sum_{m=1}^K (v_m - \bar{v})^2 \right] / K} \right). \quad (3.21)$$

Here, sd_v is the vector's *standard deviation* and the $t_{1-\frac{\alpha}{2}}(K-1)$ is the *Student's t-distribution* with $K-1$ degrees of freedom at a significance level α [21]. Every measurement uses a sample size $K = 100$ and $\alpha = 5\%$ here. Table 3.1 shows that the level deviation

$$\Delta = \text{SQNR}_{\text{dB,theo}} - \text{SQNR}_{\text{dB,meas}} \quad (3.22)$$

between the theoretical and the measured SQNR's mean is relative small for 8 bit and 16 bit. As the quantization bit number decreases the error increases. The standard deviation is conspicuously 0 at the 1 bit and 2 bit resolution. This means that the same SQNR is measured 100 times.

Next, we check the SNR estimation when white noise is present. Thus, we implement the algorithm as a Python script and simulate different variations of a sinusoid test sequence which is superimposed by noise. For this, we define the noise amplitude $\mathcal{A}_{\text{wn}} = 0.1$ for the gaussian white noise signal. The sine wave's frequency is fixed at $f_{\text{test}} = 1 \text{ kHz}$ here. Its amplitude is changed from the values

$$\mathcal{A}_{\text{test}}(n) = (0.01 \ 0.02 \ 0.03 \ \dots \ 9.98 \ 9.99 \ 10.0) \quad (3.23)$$

that the SNR of different amplitude ratios $\frac{\mathcal{A}_{\text{test}}(n)}{\mathcal{A}_{\text{wn}}}$ are estimatable. The varying of $\mathcal{A}_{\text{test}}(n)$ leads

to $K_{\mathcal{A}} = 999$ estimations, and to the ratio limits

$$\left. \frac{\mathcal{A}_{\text{test}}(n)}{\mathcal{A}_{\text{wn}}} \right|_{\min} = 0.11 \quad \text{and} \quad \left. \frac{\mathcal{A}_{\text{test}}(n)}{\mathcal{A}_{\text{wn}}} \right|_{\max} = 100.1. \quad (3.24)$$

We compute also the theoretical SNR level by

$$\text{SNR}_{\text{dB},\text{theo}}(n) = 20 \log_{10} \left(\frac{\mathcal{A}_{\text{test}}(n)}{\mathcal{A}_{\text{wn}}} \right), \text{ unit dB.} \quad (3.25)$$

Then, the deviation of the SNR estimation is calculated, and stored as the sequence

$$\Delta(n) = \text{SNR}_{\text{dB},\text{theo}}(n) - \text{SNR}_{\text{dB},\text{esti}}(n) \quad (3.26)$$

where $\text{SNR}_{\text{dB},\text{esti}}(n)$ is the estimated SNR. Now, we obtain the deviation's confidence interval by using the concept of eq. (3.21):

$$[\Delta_{\text{lo}}; \Delta_{\text{up}}] = (0.48 \pm 0.02) \text{ dB.} \quad (3.27)$$

Here, the sample size is $K_{\mathcal{A}}$ and a significance level $\alpha = 5\%$ is used. This result shows approximately the accuracy of the estimation.

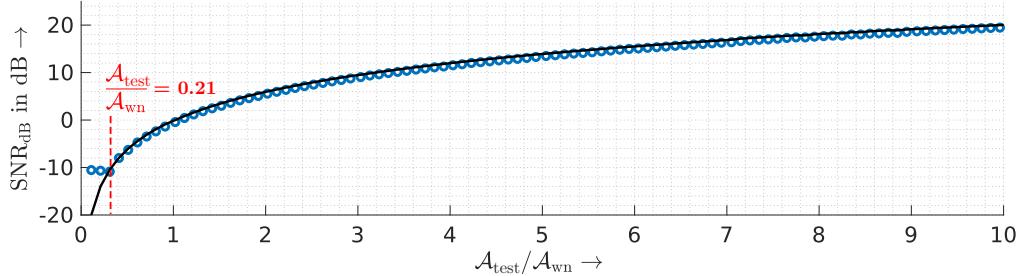


Figure 3.10.: The scatter plot shows the estimated SNR, and the black curve is the theoretical SNR.

The estimated SNR sequence $\text{SNR}_{\text{dB},\text{esti}}(n)$ and the theoretical SNR are visualized in fig. 3.10. There are not all amplitude ratios shown, just the values from 0.11 up to 10.0 because the deviation do not increase more with greater ratios. If the whole signal $\text{SNR}_{\text{dB},\text{esti}}(n)$ is printed, we will not observe well the outliers under $\frac{\mathcal{A}_{\text{test}}}{\mathcal{A}_{\text{wn}}} = 0.21$. This is the failure of the fundamental estimation algorithm, as mentioned before in section 3.2.1. If the wrong fundamental is used for the bandpass and the bandreject filter, the SNR and THD+N estimation will also fail.

The shown verification of the measurement accuracy is just a brief approach. In a future work, a statistical model is desired as an analysis tool for the measurement accuracy. For example, a linear regression model [22] of the form

$$\text{SNR}_{\text{dB}} = \lambda_0 + \lambda_1 \log_{10} \left(\frac{\mathcal{A}_{\text{test}}}{\mathcal{A}_{\text{wn}}} \right) \quad (3.28)$$

or

$$\text{SNR}_{\text{dB}} = \mu_0 + \mu_1 \log_{10} (\mathcal{A}_{\text{test}}) + \mu_2 \log_{10} (\mathcal{A}_{\text{wn}}) \quad (3.29)$$

are conceivable. The expected coefficients are $\lambda_0 = \mu_0 = 0$, $\lambda_1 = \mu_1 = 20$, and $\mu_2 = -20$

because they lead to the same equation as the theoretical SNR, cf. eq. (3.25). For instance, if the coefficients are close to the theoretical values, this is an indicator of a high measurement accuracy. The definition of close to the theoretical values has to be defined also in the future work.

3.2.3. The Input Path of the Measuring System

The input path, in fig. 3.11, is similar to the output path in fig. 3.6. First, the input signal is

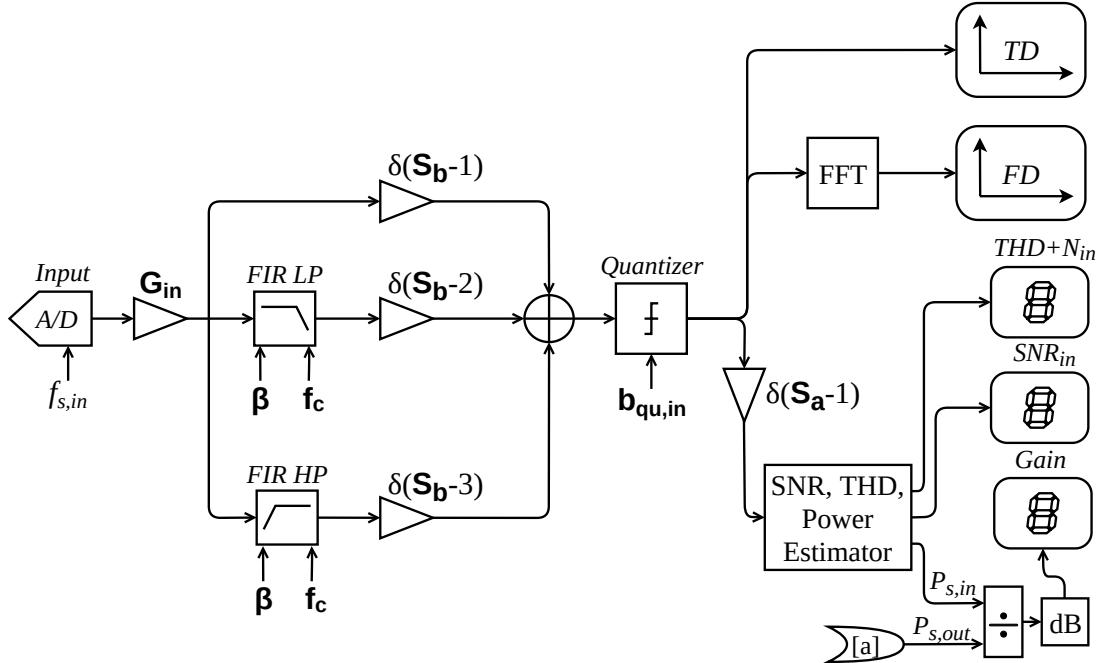


Figure 3.11.: Signal Processing of the measurement system's input path.

sampled with the frequency $f_{s,in} = 48 \text{ kHz}$. Next, the signal can be amplified by the input gain G_{in} . Its value is set by the gain slider in fig. 3.4 (b). Then, the next three paths are building a Kronecker switch, as shown in fig. 3.7. This switch's position S_b is used to change between different input filters. Its states are defined as

$$S_b = \begin{cases} 1 : & \text{no filter, i.e., input gain only} \\ 2 : & \text{input gain plus lowpass filter} \\ 3 : & \text{input gain plus highpass filter} \end{cases} . \quad (3.30)$$

S_b is the value of the input filter radio button in fig. 3.4 (b). The lowpass or highpass are FIR filters based on a Kaiser window. Their stopband ripple can easily be adjusted by changing the shape parameter β via the GUI. This and the linear phase justify the choice of the FIR structure. The linear phase is advantageous because it does not add distortion to the signal when the SNR or THD+N is measured. The maximum attenuation of both filters is $\delta_{st} = -90 \text{ dB}$ and their transmission width $\Delta f = 0.5 \text{ kHz}$. If the internal sampling rate of 48 kHz and the maximum stopband ripple are used, they will be both a 555-tap filter which results in a group delay of 5.781 ms. These are the greatest possible values because the filter order M decreases if the sampling frequency or attenuation decreases, cf. eq. (2.39). The rest of the flow chart has got

the same function as the input path. The quantizer set the input quantization to 1, 2, 4, 8, or 16 bit. This parameter is changeable by the GUI, too. Then, the time domain (TD), spectrum (FD), THD+N, SNR, and the gain are displayed. Here, the gain display is the ratio of the fundamental's input power $P_{s,in}$ to the fundamental's output power $P_{s,out}$. The relationship is swapped compared to eq. (2.47) because $P_{s,in}$ represents the output power of the DUT, and $P_{s,out}$ is the DUT's input power, as shown in fig. 3.2. The in and out indices refers to the PC's ports, or to the input and output paths, respectively. $P_{s,in}$ results directly from the SNR, THD and power estimator while $P_{s,out}$ results from the connector [a] which refers to the estimator of the output structure in fig. 3.11.

4. Measurement Demonstration

This section shows some measurements which are possible by the application. First, we capture the quality indicators of two sound cards, i.e., we judge our measurement equipment. Afterwards, we compare the quality of two amplifiers.

4.1. Characterization of Sound Cards

At a start, we determine the sound card's maximum sampling frequency $f_{s,\max}$ and the maximum quantization bits $b_{qu,\max}$ by using system tools of the operating system. These findings and all other measurement results are shown in table 4.1. There is the front audio connector's sound card of the motherboard MSI Z170-A PRO which will be denoted by *sound card A* in the following text. Then, we use the term *sound card B* for the sound card of the Lenovo Y520-15IKBN 80WK00LGGE. They are both internal sound cards of personal computers.

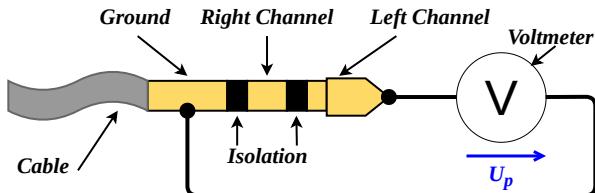


Figure 4.1.: Measuring of the sound card's peak voltage U_p , using a phone connector. Here, the left channel is measured by the voltmeter V (based on [23, p. 288/289]).

Then, the devices' peak voltages are identified that we are able to assign a physical unit to our test signals. A phone connector cable is connected to the audio's output jack which is $d = 1\text{ m}$ long, is made of copper, and has got a 24 AWG specification [24]. The open-circuit voltage is measured, as illustrated in fig. 4.1. Here, the voltmeter is a *MASTECH MS8229* [25] whose AC mode is used to measure the peak voltage. This mode measures the root mean square voltage $\text{RMS}_{u,\text{out}}$ which leads to the peak voltage of the output

$$U_{p,\text{out}} = \sqrt{2} \text{ RMS}_{u,\text{out}}, \text{ unit V(Volt)}, \quad (4.1)$$

if a sine wave is applied as a test signal [26]. Below a $\text{RMS}_u = 0.4\text{ V}$, the voltmeter has got an absolute resolution of 0.1 mV , a relative measurement error of $\Delta U\% = 0.8\%$, and works only in the frequency range from 40 Hz up to 400 Hz [25]. Hence, a sine wave of 50 Hz is used as a test signal in the GNU Radio application. The audio output is set to 50% by the PC's settings to match the settings with the next test scenarios. We set also the amplitude to full-scale (i.e., 1) that the sine wave's peak value is equivalent to the output's peak voltage $U_{p,\text{in}}$. This

measurement made possible that the amplitude $\mathcal{A}_{\text{test}} \in [-1; 1]$ is assignable to a physical voltage U_{test} :

$$U_{\text{test}} = U_{\text{p,out}} \mathcal{A}_{\text{test}}, \text{ unit V.} \quad (4.2)$$

We measure $\text{RMS}_{\text{u,A}} = 0.156 \text{ V} \pm 0.8\%$ for sound card A and $\text{RMS}_{\text{u,B}} = 0.174 \text{ V} \pm 0.8\%$ for B. Equation (4.1) leads to the peak voltages in table 4.1. It is desired that the identifications of $f_{\text{s,max}}$, $b_{\text{qu,max}}$, and $U_{\text{p,out}}$ are going to be determinable by the GNU Radio application in a future version of the software.

Table 4.1.: Measurement results of the two sound cards A and B.

	Sound Card A	Sound Card B
Device Name	MSI Z170-A PRO	Lenovo Y520-15IKBN 80WK00LGGE
$f_{\text{s,max}}$ in kHz	48	48
$b_{\text{qu,max}}$ in bit	16	16
$U_{\text{p,out}}$ in V	0.221 ± 0.002	0.246 ± 0.002
Waveform	sine	sine
f_{test} in kHz	1	1
$\mathcal{A}_{\text{test,lo}}$	0.0001	0.002
	$(0.0221 \pm 0.0002) \text{ mV}$	$(0.492 \pm 0.004) \text{ mV}$
$\mathcal{A}_{\text{test,up}}$	0.079	0.02
	$(17.459 \pm 0.158) \text{ mV}$	$(4.920 \pm 0.040) \text{ mV}$
Sample Size K	100	100
Significance Level α	5 %	5 %
LDR in dB	53.95 ± 0.47	12.93 ± 0.41
THD+N _{in} in %	0.2092 ± 0.0134	23.6362 ± 2.5152
$G_{\text{dB,up}}$ in dB	21.98 ± 0.00	11.62 ± 0.07
F_{dB} in dB	22.53 ± 0.50	51.65 ± 0.45

Now, we connect the sound card's output to its input. We are using the same 24 AWG cable for this, as before. We keep the identical 1 kHz sinusoid as a test signal but we adjust its amplitude until the input signal has got full-scale without clipping. This value is the upper amplitude $\mathcal{A}_{\text{test,up}}$ in the table 4.1. Here, the PC's output audio level is set to 50 % and the input level, too. We choose equal levels to have no additional gain of the computer's software. Both values are not set to 100 % because clipping arises at any test signal amplitude then.

The next step is to capture the measurement results using the data logger. Due to the fluctuating measurements, we record multiply values of the SNR, THD, etc. Hence, we apply a recording interval of 1 s and save $K = 100$ datasets. This leads to an observation period of 100 s for the device. One column of the resulting CSV-file is a measuring sequence and the m -th column is denoted by $v_m(n)$. Then, we calculate the arithmetic mean \bar{v}_m of each sequence $v_m(n)$ and displays its confidence interval in the form $\bar{v}_m \pm \Delta v_m$, as shown in eq. (3.21). This is the meaning of the \pm signs in table 4.1. We assume a significance level $\alpha = 5\%$ there.

Two important indicators are the maximum THD+N_{in} and the gain $G_{\text{dB,up}}$ before the compression point is reached. This is the input's THD+N because we want to judge the signal change from the whole signal path. This means that the D/A conversion is the path's start and

the end is the quantized audio which is stored digitally. The current full-scaled test signal leads to these values.

Furthermore, we use the input's $\text{SNR}_{\text{dB,in,up}}$ of this measurement series to calculate the LDR. This SNR is associated to the upper bound of the LDR. The lower bound occurs if the input signal-to-noise ratio $\text{SNR}_{\text{dB,in,up}} \approx 0 \text{ dB}$. Then, another measurement is required to obtain the amplitude $A_{\text{test,lo}}$ where this behavior occurs. We keep the parameters of the test signal the same, obviously except for the amplitude, in order to determine $A_{\text{test,lo}}$. In general, the LDR is expressible as the difference of the upper and lower SNR:

$$\begin{aligned} \text{SNR}_{\text{dB,in,up}} - \text{SNR}_{\text{dB,in,lo}} &= P_{\text{dB,in,up}} - P_{\text{dB,noise}} - (P_{\text{dB,in,lo}} - P_{\text{dB,noise}}) \\ &= P_{\text{dB,in,up}} - P_{\text{dB,in,lo}} \\ &= \text{LDR}. \end{aligned} \quad (4.3)$$

This leads to the same level difference, as shown in fig. 2.13, when both signals have got an identical noise power. If we assume white analog noise with the same bandwidth and the same device temperature, and no other interference, then they are equal [14]. The $\text{SNR}_{\text{dB,in,lo}} \approx 0$ results in

$$\text{LDR} = \text{SNR}_{\text{dB,in,up}} - \underbrace{\text{SNR}_{\text{dB,in,lo}}}_{\approx 0} \approx \text{SNR}_{\text{dB,in,up}}. \quad (4.4)$$

Thus, the measured input SNR is approximately the same as the LDR. Therefore, the LDR is determined and displayed in table 4.1, too.

The noise figure is the difference of two average SNR values, i.e., the test signal $\overline{\text{SNR}}_{\text{dB,test}}$ minus the input $\overline{\text{SNR}}_{\text{dB,in}}$. This leads statistically to an error propagation when the confidence interval is given [27]:

$$[F_{\text{lo}}; F_{\text{up}}] = \bar{F}_{\text{dB}} \pm \Delta F_{\text{dB}} = \left(\overline{\text{SNR}}_{\text{dB,test}} - \overline{\text{SNR}}_{\text{dB,in}} \right) \pm \left(t_{1-\frac{\alpha}{2}}(K-1) \frac{\text{sd}'_F}{\sqrt{K}} \right). \quad (4.5)$$

The propagated standard uncertainty sd'_F is expressible as

$$\begin{aligned} \text{sd}'_F &= \sqrt{\left(\frac{\partial \bar{F}_{\text{dB}}}{\partial \overline{\text{SNR}}_{\text{dB,test}}} \right)^2 (\text{sd}_{\text{SNR}_{\text{dB,test}}})^2 + \left(\frac{\partial \bar{F}_{\text{dB}}}{\partial \overline{\text{SNR}}_{\text{dB,in}}} \right)^2 (\text{sd}_{\text{SNR}_{\text{dB,in}}})^2} \\ &= \sqrt{(\text{sd}_{\text{SNR}_{\text{dB,test}}})^2 + (\text{sd}_{\text{SNR}_{\text{dB,in}}})^2} \end{aligned} \quad (4.6)$$

where $\text{sd}_{\text{SNR}_{\text{dB,test}}}$ and $\text{sd}_{\text{SNR}_{\text{dB,in}}}$ are the standard deviations of corresponding arithmetic means [27].

Comparison of the Sound Cards

The 16 bit quantization of the sound cards shows CD-quality in this area, but the 48 kbit sample rate indicates default studio production quality. Both peak voltages $U_{\text{p,out}}$ are too small to drive a passive device under test. Passive means that the DUT has no external power supply. For example, if we use a an Altec Lansing N1209-8A crossover network as a DUT, we are not able to measure any signal at the DUT's output because the attenuation of the passive electronic components suppresses completely the voltage. This limits the application to active DUTs when

these sound cards are utilized. This problem is probably solved if an external sound card is used which has got an own power supply. An additional amplifier, connected between the sound card's output and DUT's input, is another option to drive passive circuits. But, its deterioration in quality must be negligibly small to not effect much the measurement results. An external signal generator is another option, but the $\text{THD} + \text{N}_{\text{out}}$, $\text{SNR}_{\text{dB},\text{out}}$, and G_{dB} are not computable directly by the GNU Radio application then. Additional measuring equipment, or a signal generator which has got these features integrated, is required in this case.

The sound card A's LDR is (53.95 ± 0.47) dB, i.e, the LDR is averagely 44.43 dB below the theoretical limit of the $\text{SQNR}_{\text{dB}} = 98.08$ dB. Hence, a DUT with a greater LDR can not be characterized completely. A's $\text{THD} + \text{N}_{\text{in}}$ is $(0.2092 \pm 0.0134)\%$ and thus exceeds the often used reference of 0.1 % [4], [10]. The SNR of the whole signal path decreases by (22.53 ± 0.10) dB, according to the noise figure F_{dB} in table 4.1. The results of $\text{THD} + \text{N}_{\text{in}}$ and F_{dB} indicates that the signal path's quality loss is not negligible if this sound card is utilized as an measuring instrument. If we compare the device B's LDR, $\text{THD} + \text{N}_{\text{in}}$, and F_{dB} in table 4.1, we will notice that all values are worse than A's. Therefore, we use sound card A as a measuring device in the next section.¹

We considered only the device's behavior at 1 kHz, but we want to check their overall frequency characteristics, too. According to this, we use the classical sweep with an bandwidth of 20 kHz

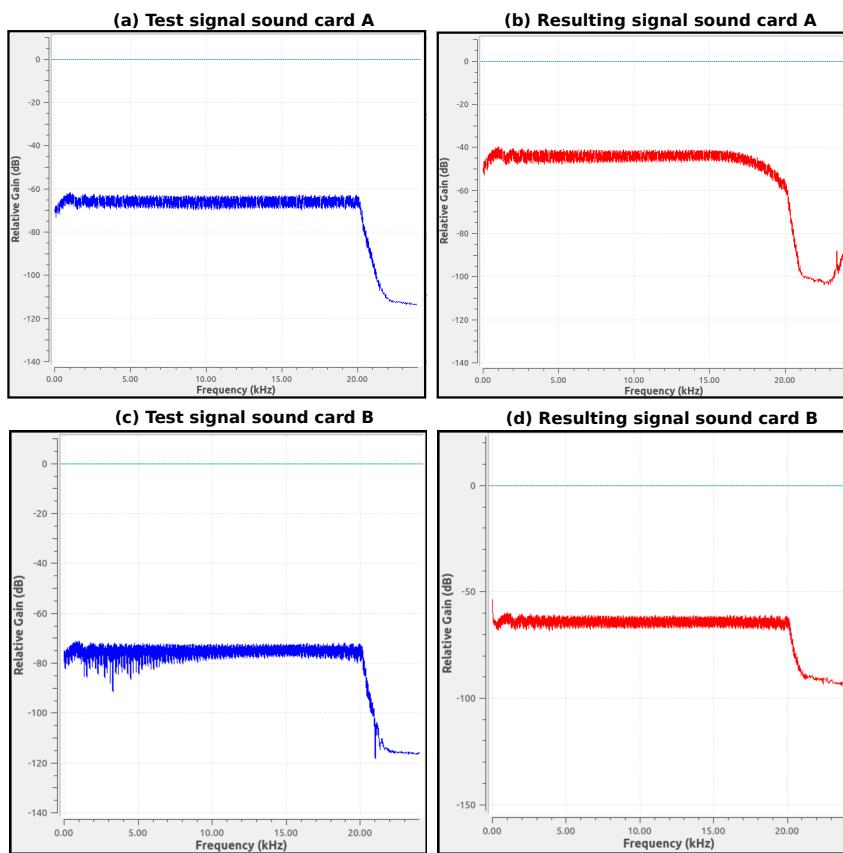


Figure 4.2.: The usage of the classical sweeps leads to the sound cards' transfer functions.

as a test signal which yields the transfer function. We adjust the amplitude to a value where no clipping occurs at the input. We set $A_{\text{test}} = 0.06$ for A and $A_{\text{test}} = 0.02$ for B in this case.

¹Other sound cards were not available at the time where the measurement phase of the project took place.

The PC's output is still connected to the input by the 1 m phone connector cable. Figure 4.2 shows the test signal on the left (a, c) and the resulting sequences on the right (b, d). Here, the offset of both resulting signals are the signal path's gain. In all 4 partial pictures, there is a ripple between 0 kHz and approximately 3 kHz which is the transient of the NCO. These spectral range should be thus investigated closer by a common waveform if a part of this range is especially interesting. It is desired that the ripple is going to be smoothed in a future version of the application. A's transfer function displays a lowpass behavior, as visualized in fig. 4.2 (b). It starts roughly at 16 kHz. Hence, this is the threshold until the magnitude drop is negligible. This LP characteristic suggests an analog filter of the D/A or the A/D conversion. It is probably an anti-aliasing filter [8] which sets the cutoff frequency to fulfill the condition (2.14) of the sampling rate. B's transfer function is flat in the subframe (d). The decreasing magnitude, at 20 kHz, results simply from the test signal's shape. B's flatness do not compensate its other quality indicators because it does not decrease the noise or the harmonics of the sound card. Therefore, sound card A is still used as the test instrument, but its lowpass behavior has to be kept in mind.

Furthermore, it is to be noted that the SNR or THD is frequency-dependend [10]. Therefore, a diagram THD versus the frequency, or SNR versus the frequency, is desired to archive a deeper noise and distortion analysis in a future version of the system.

4.2. Comparison of Two Amplifiers

We demonstrate some possible measurements by comparing two DUTs from different decades. The first one is a Fisher CA-223 amplifier and equalizer is from 1986 [28]. Its front is shown in fig. 4.3 (a). It has got 4 inputs where just one is switched at a time (tape mon., CD, tuner, phono



Figure 4.3.: The Fisher equalizer (a) and Wuhzi's low cost amplifier (b) are the DUTs.

buttons) and its outputs are four terminals for speaker wire (left+, left-, right+, right-). In the middle, there is the loudness slider which sets the overall gain of the amplifier. The equalizer are the sliders on the left which can amplify or attenuate specific frequency bands. The bands are split into 63 Hz, 250 Hz, 1 kHz, 4 kHz, and 16 kHz center frequencies. Next to the 16 kHz slider, there is the control element to change the balance between the right and left channel. The low cost amplifier is a Wuzhi Audio ZK-502C and is displayed in fig. 4.3 (b). It has got a phone connector input, a volume knob for the gain, a Bluetooth receiver, terminals for speaker wire and is from the 2020s.

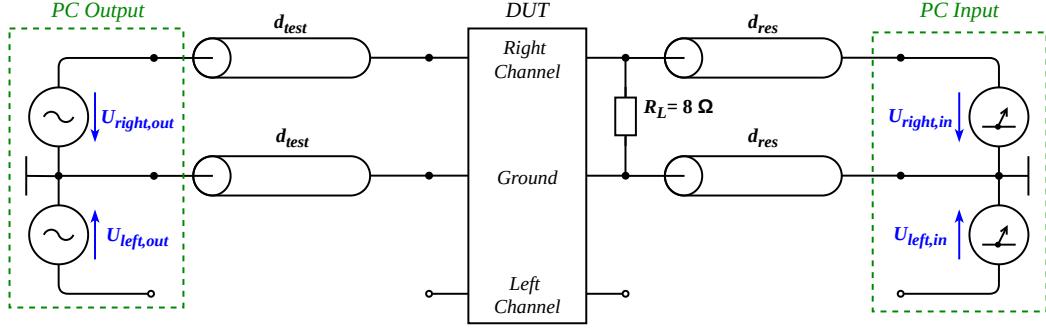


Figure 4.4.: Wiring diagram – The PC is connected to the DUT whereby the right channel is measured.

Both DUTs are connected to the PC as shown in fig. 4.4 when the right channel is measured. It is always checked one channel at a time. But, the two channels on both devices behave approximately the same that we will not distinguish between the left and right channel in the following text. The wire, from the PC's output to the DUT's input, is d_{test} long. The wire has got the length d_{res} which routes the resulting signal. For every DUT, the cable lengths are in table 4.2. d_{test} differs because a phone connector-to-RCA cable must be used as a connection due to the RCA inputs of the Fisher CA-223. Moreover, a load resistor $R_L = 8 \Omega$ is connected in parallel to the DUT's output. This simulates a real load like a speaker and is especially recommended for the equalizer [28].

Table 4.2.: Test signal parameters and quality indicators of the DUTs.

	Fisher CA-223	Wuzhi ZK-502C
Waveform	sine	sine
f_{test} in kHz	1	1
\mathcal{A}_{test}	0.01	0.01
	(2.21 ± 0.02) mV	(2.21 ± 0.02) mV
Gain Setting	loudness slider 5.5 all equalizer bands 0 dB	knob's notch -135°
d_{test} in m	1.5	1.0
d_{res} in m	1.0	1.0
Sample Size K	100	100
Significance Level α	5 %	5 %
SNR _{dB,in} in dB	20.62 ± 0.12	11.94 ± 0.13
THD+N _{in} in %	9.3422 ± 0.1418	25.3831 ± 0.3657
G_{dB} in dB	33.19 ± 0.01	32.16 ± 0.01
F_{dB} in dB	37.74 ± 0.17	46.07 ± 0.15

Then, a sine wave of frequency $f_{test} = 1$ kHz with the amplitude $\mathcal{A}_{test} = 0.01$ is chosen as a test signal for both devices. The small amplitude is chosen to create a margin that prevent the sound card from clipping when the DUTs are set to higher gains. As mentioned before in section 4.1, we take the data logger, record $K = 100$ measurements with a measuring interval of 1 s, calculate the confidence intervals in the form $\bar{v} \pm \Delta v$ ($\alpha = 5\%$), and print the results in table 4.2. We increase the Fisher's loudness slider just before clipping occurs and the Wuzhi's gain knob is set analogously. The slider has got the value 5.5. The notch of the Wuzhi's gain

knob is rotated by -135° . Here, 0° is the right point of the horizontal line when the side with the amplifier input is observed and the black plate, in fig. 4.3, is the top.

The Fisher CA-223 performs better in noise and harmonic distortion categories because the SNR is higher, the THD+N and the noise figure F_{dB} are less compared to the Wuzhi, as shown in table 4.2. These are not the absolute limits of the devices. For example, if we increase the

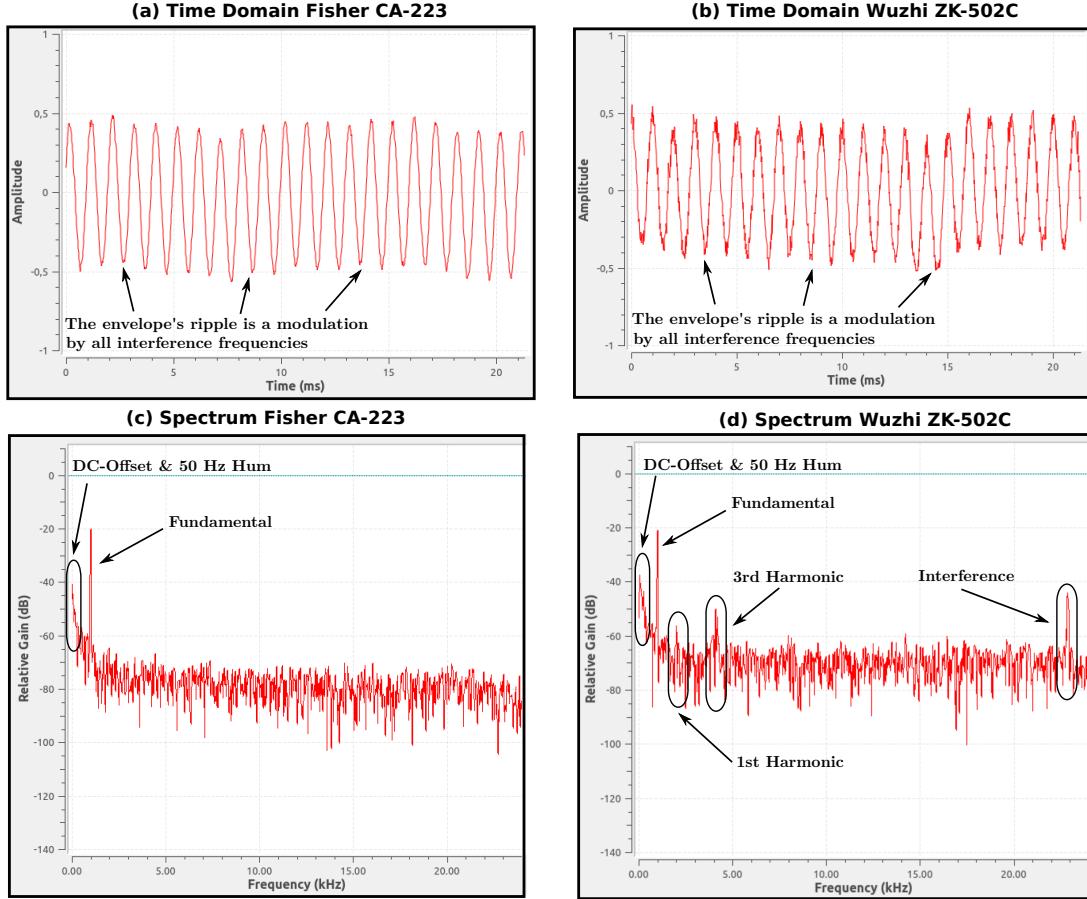


Figure 4.5.: The spectra show the reasons for the different noise and distortion indicators.

input power, the SNR will increase also because the white noise part does not depend on the signal power [14]. Other noise parts increase possibly too but the SNR rises in total. Hence, it is important to note that we do a relative comparison here. This also explains why the Fisher's label “No more than 0.5 % total harmonic distortion.” is misleading in fig. 4.3 (a). It is just obtainable if the test signals yields $P_{\text{test}} = 20 \text{ W}$ [28]. If we apply the output's peak voltage, we need an input resistance of

$$R_{\text{in}} = \frac{U_{\text{p,out}}^2}{2P_{\text{test}}} = \frac{(0.221 \text{ V})^2}{2 \times 20 \text{ W}} = 1.22 \text{ m}\Omega \quad (4.7)$$

to fulfill this condition [26]. This resistance is atypical for an amplifier circuit with transistors or operational amplifiers [29]. Therefore, the condition does not hold. This shows a limit of the sound cards as a measuring device, again.

Moreover, the cause of the noise performance can be further investigated if the resulting signal's spectrum is observed. It is displayed for the Fisher in fig. 4.5 (c) and for the Wuzhi in fig. 4.5 (d). These spectra are saved when the measurements of table 4.2 are recorded. If

we compare the devices' fundamentals, we notice that they differ slightly. The peak is roughly -20 dB. Hence, the gain is approximately the same, as the numbers show: $G_{\text{dB},\text{Fisher}} = (33.19 \pm 0.01)$ dB versus $G_{\text{dB},\text{Wuzhi}} = (32.16 \pm 0.01)$ dB. The two spectra have got a DC-offset, which is the peak at 0 Hz, and noise around the 50 Hz area. This is probably 50 Hz hum of the power supply. These interferences lead to an amplitude modulation which is visible in fig. 4.5 (a) or in fig. 4.5 (b), respectively. The THD+N is worse for the Wuzhi ((25.3831 ± 0.3657) % vs. (9.3422 ± 0.1418) %) because its frequency domain contains always a first and third harmonic, a narrow-banded interference, and the overall noise level is raised compared to the Fisher's spectrum. The two harmonics never disappear, regardless what the gain knob's position is. As mentioned before, the used sound card A has got a lowpass characteristics, i.e., the interference is actually higher than shown at 23 kHz. But, this is outside the audio range and thus is not noticeable. All these interferences are superimposed visible in the time domain, too. Here, the Wuzhi's signal is differently modulated as compared to fig. 4.5 (a) and the curve is not as smooth as the Fisher's one.

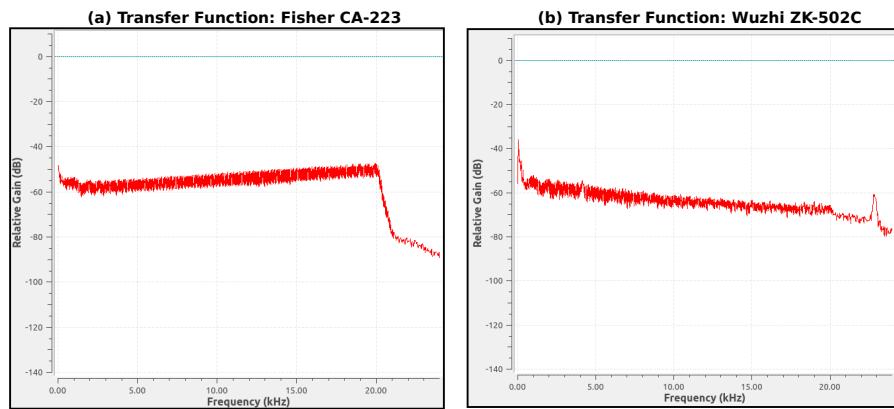


Figure 4.6.: Transfer functions of the DUTs. (a) shows a highpass characteristic and (b) behaves like a lowpass.

To get the overall spectral behavior, we use again a classical frequency sweep, as displayed in section 4.1. This leads to the transfer functions in fig. 4.6. In the partial picture (b), the Wuzhi's spectrum shows a continuous decreasing lowpass behavior. The interference is also visible at 23 kHz, likewise the DC-offset and the 50 Hz hum. The 1st harmonic is not extremely perceivable due to the ripple of the test signal at the frequencies below 3 kHz. The other harmonic at 4 kHz is slightly visible as a small notch. The Fisher amplifier has got a highpass characteristic with this equalizer configuration. Therefore, no amplifier has got a constant gain over the whole audio range. However, the Fisher CA-223 is adjustable by its internal equalizer that the curve can be lifted if necessary.

As a result, the quality indicators suggests that the Fisher amplifier performs with a higher quality than the Wuzhi ZK-502C although their manufacturing date differs approximately 34 years.

5. Conclusion and Future Work

We developed an application that uses the PC as a signal generator and analyzer for audio signals. The shown measuring concept makes it possible to calculate the quality indicators SNR, THD+N, gain, noise figure, and the LDR. Thus, a SNR, THD+N and power estimator has been developed that do not use computationally expensive FIR filters or unstable IIR filters. On the other hand, this estimator requires a large FFT length of $N = 2^{14}$.

The SQNR estimation is close to the theoretical values for a 4, 8, and 16 bit quantization. The deviation to the theory increases if the signal is quantized by 1 or 2 bit. The SNR estimation, when white noise is present, fails if the fundamental estimations fail, too. This occurs if the ratio of the signal amplitude to the noise amplitude falls below 0.21, in the shown simulation. The SNR estimation's accuracy is captured by the confidence interval of the deviation between the theoretical and estimated SNR. This interval is (0.48 ± 0.02) dB. In a future work, it is desired that the accuracy is indicated by a more complex statistical model.

Furthermore, it is possible to analyze audio equipment through a graphic representations. This is realized by a time scope and a FFT spectrum. The transfer function of the DUT is visible if the classical sweep is used as a test signal. But, there is a ripple between 0 kHz and approximately 3 kHz due to the transient of the NCO. This leads to a vague transfer function in this area. This is going to be smoothed in a future version of the system concept.

Moreover, the range of the quality indicators can be further extended. For example, the delay through the DUT, or the latency of the A/D and D/A converter, are not covered yet. It is also desired that the maximum sample rate, the maximum quantization and the peak voltages of the converters are directly determinable by the application.

Besides that, the SNR or THD+N measurements are only observed at a fixed frequency. Thus, a diagram SNR versus the frequency, or THD+N versus the frequency, is requested in a future work. This leads to a deeper quality analysis because both values are frequency-dependant. In addition, an import function for recorded audio is desired. This is usable if the quality of the DUT's input path, or its output path, is supposed to be analyzed separately. For example, a device with a recording ability saves the test signal which passed the equipment's input path. Then, the import function can be applied which calculates the quality indicators of that imported sequence.

GNU Radio Implementation

The implementation of the system concept by GNU Radio Companion is beneficial because a lot of Digital Signal Processing functions are given, and the creation of own Python blocks extends the development possibilities. The GUI elements are also fast to apply, but they are not much adjustable. For example, the decimal places of a number display are not changeable. This suggests too high measuring accuracy.

Sound Cards as a Measuring Instrument

The measurements show that the sound card has got limits, used as a measuring instrument. The first limit is their quality loss which is not negligibly here. For example, the SQNR bounds theoretically the maximum recordable LDR of the audio device. In practice, the card's LDR is the maximum which is measurable. A DUT with a greater LDR can thus not be characterized completely. This behavior must be kept in mind if a DUT is measured, and other quality indicators, like a lowpass transfer function, too.

The second limit is the low output voltage which impedes the characterization of passive DUTs because the test signals have not enough power to drive the device. The other effect of this is that some devices do not receive their maximum input power which will not lead to the DUT's maximum SNR or THD. Hence, just relative comparisons of audio devices are possible. However, this is sufficient to judge which audio device performance better, as shown in section 4.2. This problem is possibly solved through an active sound card, or an amplifier whose deterioration in quality is negligibly small to not effect much the measurement results.

Overall, an internal sound card is usable as a measuring instrument, but their limits have to be determined that the measurement results have got a correct interpretation.

References

- [1] B. L. Ladage. (Feb. 25, 2022). “Sdat – Software Defined Audio Testing, A GNU Radio Application to Measure the Quality of Audio Devices.” 17:15 UTC+1, [Online]. Available: <https://github.com/bladage/Sdat>.
- [2] D. W. Kammler, *A first course in Fourier analysis*. New Jersey: Prentice-Hall, Inc., 2000.
- [3] H. Schulze, *Signale und Systeme*, lecture notes South Westphalia University of Applied Sciences, Meschede, Apr. 2021.
- [4] S. Ries and M.-T. Roeckerath-Ries, *Ingenieurmathematik 2 für Elektrotechniker, Maschinenbauer und Wirtschaftsingenieure, Studienbuch*, 1. Auflage. Wissenschaftliche Genossenschaft Südwestfalen eG, 2019.
- [5] S. Ries, *Digitale Signalverarbeitung, Ab Sommersemester 2020*, lecture notes South Westphalia University of Applied Sciences, Meschede, Mar. 2020.
- [6] R. G. Lyons, *Understanding Digital Signal Processing*, 3rd ed. New Jersey: Prentice Hall, Inc., 2011.
- [7] L. B. Jackson, *Digital Filters and Signal Processing, with MATLAB Exercises*, 3rd ed. Norwell: Kluwer Academic Publishers, 1996.
- [8] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 3rd ed. Pearson Education, 2010.
- [9] K.-D. Kammeyer et al., *Digitale Signalverarbeitung, Filterung und Spektralanalyse mit MATLAB-Übungen*, 5. Auflage, K.-D. Kammeyer and K. Korschel, Eds. Stuttgart Leipzig Wiesbaden: B. G. Teubner, Nov. 2002.
- [10] B. Metzler, *Audio Measurement Handbook*, Second edition for PDF. Beaverton: Audio Precision, Inc., Jan. 2005.
- [11] International Telecommunication Union, “Method for objective measurements of perceived audio quality,” Recommendation ITU-R BS.1387-18, 2001.
- [12] K. Beuth et al., *Nachrichtentechnik*, 4., vollständig überarbeitete und aktualisierte Auflage, ser. Elektronik. Würzburg: Vogel Business Media GmbH & Co. KG, 2016, vol. 7.
- [13] R. Cabot et al., “Aes standard method for digital audio engineering - measurement of digital audio equipment,” Audio Engineering Society, Revision AES17-1998 (r2004), Mar. 2004.
- [14] D. M. Pozar, *Microwave Engineering*, 4th ed. John Wiley & Sons, Inc, 2012.
- [15] D. Shmilovitz, “On the definition of total harmonic distortion and its effect on measurement interpretation,” *IEEE Transactions on Power Delivery*, vol. 20, no. 1, pp. 526–528, 2005.

- [16] Deutsches Institut für Normung, “Elektroakustische Geräte – Teil 3: Verstärker,” Standard DIN EN IEC 60268-3:2018, Feb. 2019.
- [17] GNU Radio Homepage. (Jan. 25, 2022). 16:30 UTC+1, [Online]. Available: <https://www.gnuradio.org/about/>.
- [18] B. Siva Kumar Reddy and K. Subramanya Chari, “Software Defined Radio: USRP N210 with GNU Radio,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 5, pp. 1804–1809, Mar. 2020.
- [19] NumPy Homepage. (Jan. 26, 2022). 11:26 UTC+1, [Online]. Available: <https://numpy.org/>.
- [20] SciPy Homepage. (Jan. 26, 2022). 11:27 UTC+1, [Online]. Available: <https://scipy.org/>.
- [21] L. Fahrmeir et al., *Statistik, Der Weg zur Datenanalyse*, 8., überarbeitete und ergänzte Auflage. Berlin Heidelberg: Springer-Verlag, 2016.
- [22] S. Chatterjee and A. S. Hadi, *Regression Analysis By Example*, 5th ed. John Wiley & Sons, Inc., 2012.
- [23] A. Friesecke, *Die Audio-Enzyklopädie, Ein Nachschlagewerk für Tontechniker*, 2nd ed. Berlin Boston: DE GRUYTER SAUR, 2014.
- [24] ASTM International, “Standard specification for standard nominal diameters and cross-sectional areas of awg sizes of solid round wires used as electrical conductors,” Standard ASTM B258-02, 2002.
- [25] MASTECH, *MS8229 Digital Multimeter Operation Manual*, 00-05-3502, 2016.
- [26] V. Metha and R. Metha, *Basic Electrical Engineering, For B.E./B.Tech. and Other Engineering Examinations*, 6th ed. New Delhi: S. Chand and Company PVT. LTD., 2012.
- [27] Joint Committee for Guides in Metrology, “Evaluation of measurement data – Guide to the expression of uncertainty in measurement,” Standard JCGM 100:2008, Sep. 2008.
- [28] Fisher Hi-Fi Europa Vertriebs GmbH, *Service Manual Fisher CA-223*, München, Apr. 1986.
- [29] U. Tietze and C. Schenk, *Halbleiter-Schaltungstechnik*, 10. Auflage. Berlin Heidelberg New York: Springer-Verlag, 1993.

A. Decibel, Power, Energy and RMS

A ratio in decibel is often called level. It *always* represents a *power ratio* [6]. Hence, a *relative level* between two signals is defined as

$$L_{\text{dB}} = 10 \log_{10} \left(\frac{P_x}{P_y} \right), \text{ unit dB,} \quad (\text{A.1})$$

where P_x is the power of signal $x(\cdot)$ in Watt and P_y is the power of signal $y(\cdot)$. If we measure a magnitude \mathcal{M} , an amplitude \mathcal{A} or a root mean square RMS instead of the power P , and want to display our measurements in dB, then the fraction in eq. (A.1) must be a power ratio, still. Fortunately, we can square the mentioned values and obtain a power unit [6]:

$$\mathcal{M}^2 \sim P, \mathcal{A}^2 \sim P \text{ and RMS}^2 \sim P. \quad (\text{A.2})$$

Hence, we write eq. (A.1) as

$$L_{\text{dB}} = 10 \log_{10} \left(\frac{\mathcal{M}_x^2}{\mathcal{M}_y^2} \right) = 20 \log_{10} \left(\frac{\mathcal{M}_x}{\mathcal{M}_y} \right), \text{ unit dB,} \quad (\text{A.3})$$

to calculate the correct level. We can use \mathcal{A} or RMS instead of \mathcal{M} in eq. (A.3), too.

An *absolute power level* is referred to a reference value like one milliwatt:

$$L_{\text{dBm}} = 10 \log_{10} \left(\frac{P_x}{1 \text{ mW}} \right), \text{ unit dBm.} \quad (\text{A.4})$$

Then, the decibel unit is related to the reference, i.e., 0 dBm is equal to 1 mW because

$$P_x = 1 \text{ mW} \cdot 10^{L_{\text{dBm}}/10} = 1 \text{ mW} \cdot 10^0 = 1 \text{ mW} \quad (\text{A.5})$$

holds. Instead, 0 dB means that the two compared signals has got the same power.

Decibel of Discrete Signals

The energy of a discrete signal $x(n)$ is defined as

$$E_x = \sum_{n=0}^{N-1} |x(n)|^2. \quad (\text{A.6})$$

Now, we divide eq. (A.6) by the signal length N and obtain the average power of the signal [6]:

$$\bar{P}_x = \frac{1}{N} E_x = \frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2. \quad (\text{A.7})$$

If we take (A.6) and (A.7), we will see that the level of the average discrete power and discrete energy is the same:

$$L_{\text{dB}} = 10 \log_{10} \left(\frac{\bar{P}_x}{\bar{P}_y} \right) \quad (\text{A.8})$$

$$= 10 \log_{10} \left(\frac{\frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2}{\frac{1}{N} \sum_{n=0}^{N-1} |y(n)|^2} \right) \quad (\text{A.9})$$

$$= 10 \log_{10} \left(\frac{\sum_{n=0}^{N-1} |x(n)|^2}{\sum_{n=0}^{N-1} |y(n)|^2} \right) \quad (\text{A.10})$$

$$= 10 \log_{10} \left(\frac{E_x}{E_y} \right), \text{ unit dB.} \quad (\text{A.11})$$

Also, we can calculate the square root of eq. (A.7) and get the root mean square

$$\text{RMS}_x = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2} \quad (\text{A.12})$$

which has got the same unit as a discrete magnitude or amplitude [6]. Therefore, eq. (A.3) comes into play:

$$L_{\text{dB}} = 20 \log_{10} \left(\frac{\text{RMS}_x}{\text{RMS}_y} \right) \quad (\text{A.13})$$

$$= 20 \log_{10} \left(\frac{\sqrt{\frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2}}{\sqrt{\frac{1}{N} \sum_{n=0}^{N-1} |y(n)|^2}} \right) \quad (\text{A.14})$$

$$= 20 \log_{10} \left(\sqrt{\frac{\bar{P}_x}{\bar{P}_y}} \right) \quad (\text{A.15})$$

$$= 10 \log_{10} \left(\frac{\bar{P}_x}{\bar{P}_y} \right), \text{ unit dB.} \quad (\text{A.16})$$

This shows that the $20 \log_{10}(\cdot)$ ensures a power ratio if the RMS is used. In addition, table A.1 is printed to interpret levels and their corresponding ratios faster.

Table A.1.: Important decibel values to interpret levels faster (based on [6, Table E-1]).

L_{dB}	P_x/P_y	$\frac{\mathcal{M}_x}{\mathcal{M}_y}$, $\frac{\mathcal{A}_x}{\mathcal{A}_y}$ or $\frac{\text{RMS}_x}{\text{RMS}_y}$
-20 dB	1/100	1/10
-10 dB	1/10	$1/\sqrt{10} \approx 0.32$
-6 dB	1/4	1/2
-3 dB	1/2	$1/\sqrt{2} \approx 0.7$
0 dB	1	1
3 dB	2	$\sqrt{2} \approx 1.4$
6 dB	4	2
10 dB	10	$\sqrt{10} \approx 3.2$
20 dB	100	10
30 dB	1 000	$\sqrt{1 000} \approx 32$
40 dB	10 000	100

Declaration of Originality

I confirm that the submitted Project Work

**“Development of an Application to Measure
the Quality of Audio Devices using GNU Radio”**

is original work and was written by me without further assistance. Appropriate credit has been given where reference has been made to the work of others. The thesis was not examined before, nor has it been published.

I understand that this work will also be reviewed electronically for compliance with academic standards and may qualify as an attempt to cheat if plagiarism is found.

Persönliche Erklärung

Ich versichere hiermit, dass ich meine Projektarbeit

**“Development of an Application to Measure
the Quality of Audio Devices using GNU Radio”**

selbstständig und ohne fremde Hilfe angefertigt habe und dass ich alle von anderen Autoren wörtlich übernommenen Stellen wie auch die sich an die Gedankengänge anderer Autoren eng anlegenden Ausführungen meiner Arbeit besonders gekennzeichnet und die Quellen zitiert habe.

Die Arbeit habe ich in dieser oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Hochschulprüfung noch nicht vorgelegt.

Mir ist bekannt, dass diese Arbeit auch auf elektronischem Wege auf Einhaltung wissenschaftlicher Standards überprüft wird und im Falle eines Plagiats als Täuschungsversuch qualifiziert werden kann.

Meschede, March 1, 2022

Bennet Luke Ladage