VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

MODERNÍ METODY REKONSTRUKCE SATUROVANÝCH AUDIO SIGNÁLŮ

MODERN METHODS OF RECONSTRUCTION OF SATURATED AUDIO SIGNALS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Aneta Mikulášková

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Mgr. Pavel Rajmic, Ph.D.

BRNO 2022

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní program **Audio inženýrství**
specializace Zvuková produkce a nahrávání
Ústav telekomunikací

Studentka: Bc. Aneta Mikulášková ***ID:*** 182626
Ročník: 2 ***Akademický rok:*** 2021/22

NÁZEV TÉMATU:

Moderní metody rekonstrukce saturovaných audio signálů

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s modelováním saturačních zkreslení audio signálů (tvrdý clipping, měkký, simulace elektronky apod.). Dále se seznamte s metodami rekonstrukce signálů, které byly saturací ovlivněny. Pochopte souvislost desaturačních úloh s úlohami dekvantizačními (tedy s případy, kdy signál je degradován skalárním kvantováním). Jakožto základní a výchozí algoritmus uvažujte metodu určenou pro tvrdý clipping, typicky nastávající v nahrávacím řetězci při tzv. přebuzení vstupní škály. Tuto metodu uzpůsobte pro ostatní, obecnější případy saturace. Vzniklé metody implementujte v MATLABu, otestujte, porovnejte na simulovaných i reálných zvukových datech objektivními i subjektivními metrikami a proveďte vyhodnocení, také oproti komerčně dostupným řešením jako např. Izotope RX.

DOPORUČENÁ LITERATURA:

[1] Udo Zoelzer, DAFX - Digital Audio Effects, John Wiley and sons, 2002.
[2] Pavel Záviška, Ondřej Mokrý, and Pavel Rajmic, A proper version of synthesis-based sparse audio declipper, in Acoustics, Speech and Signal Processing (ICASSP), 2019 IEEE, 2019.

Termín zadání: 7.2.2022 ***Termín odevzdání:*** 24.5.2022

Vedoucí práce: prof. Mgr. Pavel Rajmic, Ph.D.

doc. Ing. Jiří Schimmel, Ph.D.
předseda rady studijního programu

ABSTRACT

This thesis presents a variety of soft clipping simulations and their respective de-clipping functions. A restoration method uses a Douglas-Rachford algorithm (already verified for hard de-clip). The algorithm is extended by inverse functions and simultaneous de-quantization is proposed. The simulations are applied to artificial signals, and audio samples, and the proposal is tested in the computing environment MATLAB. A restored signal is later evaluated by objective and subjective methods.

KEYWORDS

Saturation, soft clipping, signal restoration, quantization, de-quantization, de-clip

ABSTRAKT

Práce se zabývá simulacemi a následnou rekonstrukcí měkké saturace, u které dochází při zpracování audio signálů. Metoda obnovy využívá jako výchozí Douglas-Rachfordův algoritmus (již ověřený pro tvrdý ořez signálu). Algoritmus je rozšířen o inverzní funkce a de-kvantizaci. Simulace jsou aplikovány na reálných zvukových datech a návrh je testován ve výpočetním prostředí MATLAB. Obnovený signál je vyhodnocen za pomocí objektivních a subjektivních metod.

KLÍČOVÁ SLOVA

saturace, měkká saturace, rekonstrukce signálu, kvantizace, de-kvantizace, de-saturace

ROZŠÍŘENÝ ABSTRAKT

Diplomová práce se zabývá rekonstrukcí měkce saturovaných audio signálů. V první kapitole teoretické části se nachází stručný úvod do problematiky saturace, jejího původu a vlivu na audio signály. Dále se rozebírají známé typy měkkých saturací, kde se konkrétně jedná o efekty *Distortion*, *Overdrive*, simulaci elektronkového zesilovače a zkreslení záznamem magnetického pásku. Příslušné simulace těchto typů zkreslení jsou matematicky definovány za pomoci nelineárních převodních charakteristik a jejich vliv na signál v časové i kmitočtové oblasti je rozebírán. Je zde též zmíněno a znázorněno zkreslení, při kterém dochází při dynamické úpravě signálu a využití těchto zkreslení v hudební produkci.

Druhá kapitola se zabývá kvantizací, k jakému zkreslení ve spektru dochází při kvantizaci signálu a jakou chybu vnáší A/D převod do problematiky rekonstrukce měkce saturovaných signálů. Pojem de-kvantizace je zaveden a vysvětlena souvislost mezi problematikou de-saturace a de-kvantizace.

Třetí kapitola přibližuje různé přístupy k rekonstruci saturovaných signálů z předešlých výzkumů. Věnuje se restauraci tvrdě saturovaných signálů a popisuje základy metod využívající řídké reprezentace signálů. Definují se pojmy jako syntetický a analytický model rekonstrukce, báze, rámce a využití STFT/Gaborovy analýzy. Následuje formulace minimalizační úlohy pro tvrdou de-saturaci. Výchozími poznatky pro měkkou de-saturaci jsou přístupy použité pro tvrdou de-saturaci, konkrétně využití Douglas-Rachfordova dopředně zpětného algoritmu. Pro měkkou de-saturaci byla upravena projekce na množinu přípustných řešení. Narozdíl od tvrdé de-saturace, kde za pomocí rozdělení signálu do „masek“, tedy vektorů vzorků, které byly saturovány a které jsou věrohodné a neměnné, v případě měkké de-saturace algoritmus minimalizuje počet řešení v rámci intervalů, ve kterých byl původní saturovaný signál kvantován. Na tyto intervaly jsou aplikovány zároveň inverzní saturační funkce.

Čtvrtá kapitola popisuje implementaci výše zmíněného řešení. Saturace a algoritmus je nasimulován ve výpočetním prostředí MATLAB za pomocí funkcí z LTFAT toolboxu. Funkčnost byla nejdříve otestována na simulovaných signálech a poté se testovaly reálné audio data. Porovnávalo se celkem 9 audio vzorků, sestávajících z lidské řeči, sólových nástrojů a hudebních ukázek. Všechny tyto hudební vzorky byly saturovány čtyřmi typy saturací, které byly uvedeny v první části. Rekonstrukce byla posuzována objektivními i subjektivními metodikami, tedy vyhodnocením zlepšení pomocí SDRΔ oproti saturovanému signálu, psychoakustického modelu PEMO-Q a subjektivním poslechovým testem MUSHRA s patnácti respondenty.

Lze říci, že rekonstrukce byla úspěšná u všech typů testovaných vzorků, ať už se jednalo o sólové nástroje či komplexnější hudební nahráky. I při relativně nízkém nakvantování (8 bit) a slyšitelné měkké saturaci audio signálů bylo zlepšení dle

SDRΔ u všech testovaných vzorků okolo 30 dB. Hodnocení dle PEMO-Q využívající ODG stupnici vykazovalo horší výsledky, průměrně -2 (tedy mírně nepříjemné). Respondenti subjektivního ohodnocení MUSHRA zaznamenali mírné změny mezi rekonstruovanými vzorky a referencí, ale u některých rekonstruovaných vzorků docházelo i k jejich záměně s referencí. V porovnání s komerčně využívaným desaturačním software Izotope RX dosahovala testovaná metoda rekonstrukce lepších výsledků, jak při vyhodnocení za pomocí SDRΔ, PEMO-Q tak i u respondentů.

MIKULÁŠKOVÁ, Aneta. *Moderní metody rekonstrukce saturovaných audio signálů*. Brno: Brno University of Technology, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2022, 58 p. Master's Thesis. Advised by prof. Mgr. Pavel Rajmic, Ph.D.

Author's Declaration

Author:	Bc. Aneta Mikulášková
Author's ID:	182626
Paper type:	Master's Thesis
Academic year:	2021/22
Topic:	Moderní metody rekonstrukce saturovaných audio signálů

I declare that I have written this paper independently, under the guidance of the advisor and using exclusively the technical references and other sources of information cited in the paper and listed in the comprehensive bibliography at the end of the paper.

As the author, I furthermore declare that, with respect to the creation of this paper, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation § 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll. of the Czech Republic, Section 2, Head VI, Part 4.

Brno ....

author's signature*

*The author signs only in the printed version.

ACKNOWLEDGEMENT

Děkuji vedoucímu diplomové práce prof. Mgr. Pavlu Rajmicovi Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc při zpracování mé diplomové práce, Ing. Pavlu Záviškovi za cenné rady při řešení praktické části práce a svým nejbližším za trpělivost a podporu.

Contents

List of Figures

Introduction

Nowadays, digital signal processing (DSP) is a necessary part of most engineering fields. When it comes to signal processing of telecommunications, sonar, radar, digital image processing, data compression, video coding, audio, or speech, DSP becomes the primary tool in most of these operations. The mathematical operations and principles throughout all mentioned fields can be similar. This thesis presents the importance and the uses of DSP in the last area, audio and speech signals processing.

Although the applications of DSP vary based on specific sectors, it has already surpassed and proven many advantages over analog processing. Nevertheless, there are several occasions where processed signals may lose essential parts of their data. Data loss may occur while converting the signal from analog to digital (ADC), for example, by choosing a low sampling frequency and underestimating the required bit-depth and saturation.

This thesis deals with signal saturation, which typically occurs while recording music, where the input analog signal exceeds the maximum limit of a recording device. If this excess went unnoticed during recording and the soundtrack was not re-recorded, the limited signal cannot be used to work further, as it would cause issues in future processing, mixing, or mastering. This issue also affects musicians who may not want or cannot repeat the recorded part destroyed by unwanted saturation. During recording, conversion, and reproduction, the signal may undergo countless processes, causing nonlinear distortions, and the original idea is more or less deformed. Soft de-clipping serves its purpose of compensating of the real-world devices (such as amplifiers or magnetic recorders) that worsen and impact the audio signal's quality and original intent.

The first chapter explains the origin, impact, and different types of clipping and how they affect audio signal and saturation as a part of artistic expression. Such saturated signals are used in later chapters by the restoration algorithm. The second chapter explains the effect of quantization. In the third chapter, different approaches and ideas behind de-clipping are described. The fourth chapter sums up the MATLAB implementation, results, and various comparative methods.

1 Saturation

1.1 Origin

A signal is considered saturated when a maximum power level of a certain device is exceeded. The output reaches the design limit of the device and cannot be produced above this level, also known as a *threshold*. Any samples exceeding this threshold are cut off, and their data is effectively lost.

Data loss manifests visually in the time domain (as mentioned threshold cutoff) as well as in the frequency domain by producing higher harmonic frequencies. These are the perceivable result of harmonic distortion. The spectrum form depends on the exact type of saturation, as it will be shown and described in the following chapters (1.3.1, 1.3.2, 1.3.5 and 1.3.6) on the simplest shape of a signal: sinusoidal signal.

In the analog world, many sources can cause saturation. Every analog device or electrical circuit consists of components, each of which has its operation point and follows respective nonlinear transfer functions specified by their physical limitation. Saturation occurs when the signal is outside a specific operational range. The output signal of analog devices is limited by their power supply and cannot be amplified further. The same applies to integrated circuits or vacuum tubes where the tube parameters limit the number of transmitted electrons. Clipping also can occur while the transistor is pushed to its maximum amplitude. Transistors are often driven into hard saturation to be used as switches [2]. When the input signal is amplified enough, the output takes a square shape where, in most cases, only two values exist.

In the digital domain, the emergence of hard clips depends on the efficiency of ADC. Input voltage value V is given and described as a maximum that can be converted due to a dynamic range of a digital system [3]. Based on the number of bits, the amount of levels within this range is computed from this value. Every discrete part of a signal exceeding the threshold is converted into a single number. Thus hard clipping is performed.

1.2 Nonlinear Processing

1.2.1 Definition

A short introduction to nonlinear processing is required since all the clipping functions from chap. (1.3) belong to this section.

Every signal processing device, either analog or digital that does not satisfy the condition of linearity, is considered as nonlinear. A system in 1.1 where the input is $x(n)$ and $y(n)$ denotes an output is a linear system [1].

$$x(n) = Ax_1(n) + Bx_2(n) \rightarrow y(n) = Ay_1(n) + By_2(n) \tag{1.1}$$

When the input $x(n)$ of such a system is a continuous sinusoidal signal with known frequency f_1 and amplitude A 1.2:

$$x(n) = A\sin(2\pi f_1 Tn), \tag{1.2}$$

the corresponding output $y(n)$ is also a sinusoid (1.3) where the amplitude A is modified as $A_{out} = |H(f_1)| \cdot A$ by the magnitude response of the transfer function $|H(f_1)|$ and the phase response $\phi_{out} = \phi_{in} + \angle H(f_1)$.

$$y(n) = A_{out}\sin(2\pi f_1 Tn + \phi_{out}), \tag{1.3}$$

On the other hand, the output of the nonlinear system is rather expressed as a sum of sinusoids, as demonstrated in Fig. 1.1: [1]

$$y(n) = A_0 + A_1\sin(2\pi f_1 Tn) + A_2\sin(2 \cdot 2\pi f_1 Tn) + ... + A_N\sin(N \cdot 2\pi f_1 Tn), \tag{1.4}$$

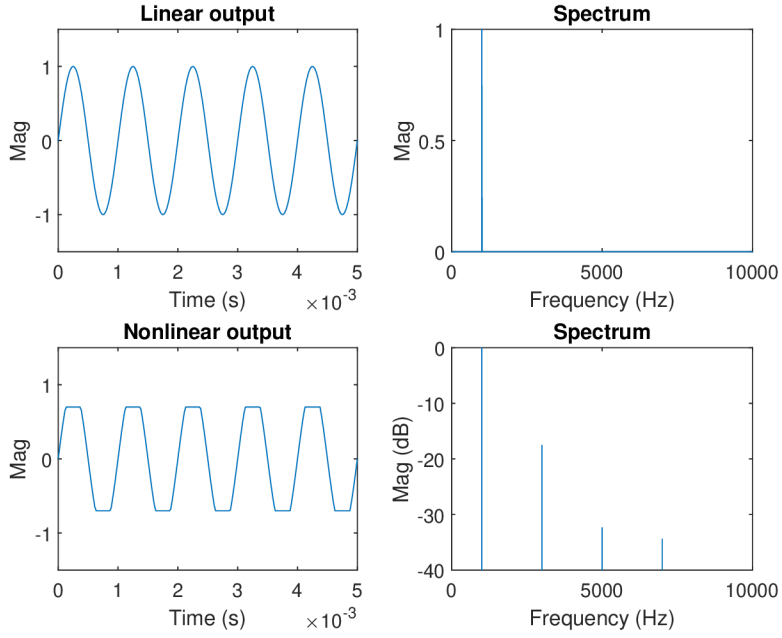

Fig. 1.1: Comparison of the linear and the nonlinear system

In music, nonlinear processing takes the form of dynamic range controllers (mainly controlling the signal envelope where the generation of higher harmonics should be as low as possible), strong harmonic distortions (guitar amps, effect processors) and

modulation effects (such as chorus, vibrato), and many others. Exciters and enhancers are signal processing devices creating higher harmonics and mixing them with the original signal improving its timbre.

1.2.2 Harmonics

The previously shown nonlinear system was an example of hard-clipping distortion. However, this type of distortion can only be a small part of distortion products which occur in standard musical signals. The total harmonic distortion (THD) is one of many ways of indicating nonlinearity. It is defined as the „*square root of the ratio of the sum of powers of all harmonic frequencies above the fundamental frequency to the power of all harmonic frequencies including the fundamental frequency*“. [1] (1.5)

$$THD = \sqrt{\frac{A_2^2 + A_3^2 + ... + A_N^2}{A_1^2 + A_2^2 + ... + A_N^2}}, \tag{1.5}$$

If the input signal consists of the sum of harmonics of a fundamental frequency f_0, the other partial distortion products are also in a row of fundamental's harmonics. The output signal then contains harmonics of amplitudes unrelated to the input amplitudes.

In case input contains non-harmonic partials, the system adds a series of harmonics based on each input partial. The interference between these series generates other respective higher harmonics called *intermodulation products*. Such products are mostly unwelcome because they can blur the signal or provide new harmonic parts which are not precisely in tune. If this happens, the input signal has to undergo proper filtering. [1]

1.2.3 Characteristic Curve

DSP is mainly built on linear time-invariant systems, but commonly used audio reproduction systems and devices have several nonlinearities such as valve amplifiers, tape recorders, or loudspeakers. Such nonlinearities are simulated and modeled by suitable approaches, represented by nonlinear systems without a memory. Static nonlinear curves are directly used on the input signal. The characteristic curves are then simple graphic displays realized as an output to input relation. [1] Since the input signal's bandwidth after nonlinear system processing may exceed half the sampling frequency (thus, the Nyquist theorem is not fulfilled), the input should be first over-sampled to precede any aliasing distortion. [1]

1.3 Clipping types

Clipping is categorized into two main kinds; hard and soft. It is possible to closely define types of saturation based on their respective characteristic curves. Further distinguishing is either as symmetrical or asymmetrical, meaning that the clipped amplitudes are the same in the positive and negative part of the signal (in geometric terms: centrally symmetrical), or only one of those amplitudes is saturated. The part around $y = 0$ is close to or equal to linearity.

1.3.1 Hard Clipping

As mentioned earlier, hard clipping is the most common form of saturation. The amplitude is cut at a certain point, and magnitudes above the specified threshold are assigned with a single maximum value. Thus, peak information is eliminated, and the waveform over the threshold is flattened and clipped. The system's output is determined by condition, where the threshold is θ (1.6).

$$y(n) \begin{cases} \theta, & x(n) \geq \theta, \\ x(n), & -\theta < x(n) < \theta, \\ -\theta, & x(n) \leq -\theta \end{cases} \tag{1.6}$$

As shown in (Fig. 1.2), both amplitudes of the nonlinear characteristic curve are clipped symmetrically, the middle part is linear, and the spectrum produces high order harmonics, thus a characteristic aggressive, digitally distorted sound is produced.

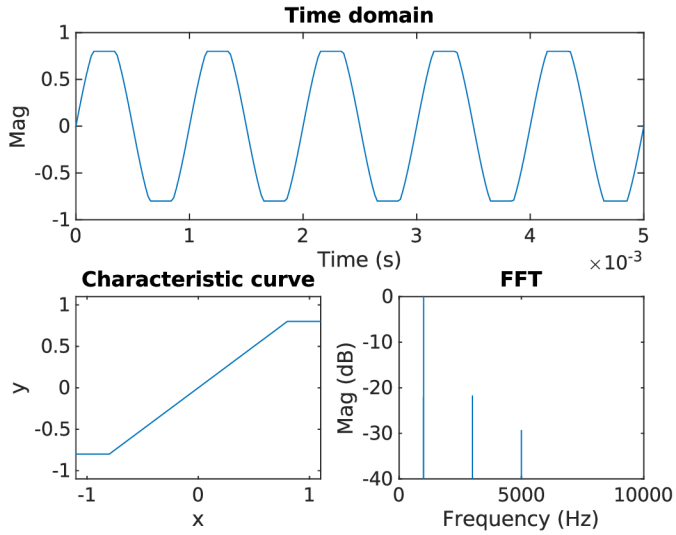

Fig. 1.2: Representation of the hard clipped signal

1.3.2 Soft Clipping

In contrast to a hard clip, a soft clip saturates the amplitude as a smooth curve and rounds its shape before the amplitude reaches the threshold. [1] This smooth amplitude reduction can typically be found in analog systems such as vacuum tube amplifiers or magnetic tapes. Soft clipping favors the human ear more than hard clipping and produces odd-order harmonics. The major difference between the symmetrical soft clipping and hard clipping is that higher harmonics appear to have much greater magnitudes in the case of the hard clip. The amount of data loss is adequate for the amount of compression caused by soft clipping. Moreover, signals saturated by soft clipping are more likely to be reconstructed if their characteristic curves are known. Since no part of the signal was completely lost, reversion is possible by applying a memory-less inverting polynomial of a finite degree. [13]

1.3.3 Distortion

The effect of distortion can be simulated according to [1] by a function:

$$f(x) = \operatorname{sgn}(x)\left(1 - \mathrm{e}^{-|x|}\right) \tag{1.7}$$

Together with an overdrive effect represents a processing device that alters the sound by increasing the input gain. The amount of preceding multiplication of the input x leads to an equivalent amount of distortion. Here, in the Fig. 1.3 is gain = 4.

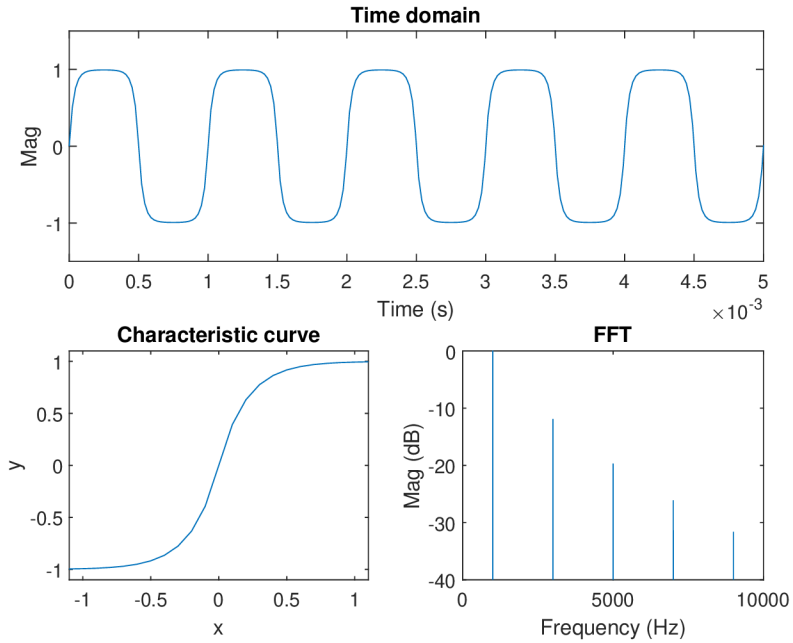

Fig. 1.3: Distortion effect

1.3.4 Overdrive

The modulation function (1.8) is an overdrive effect precisely, which saturates the signal as following:

$$f(x)\begin{cases} 2x, & 0 \leq x \leq \frac{1}{3}, \\ \frac{3-(2-3x)^2}{3}, & \frac{1}{3} \leq x \leq \frac{2}{3}, \\ 1, & \frac{2}{3} \leq x \leq 1 \end{cases} \tag{1.8}$$

Conditions split static characteristic $y = f(x)$ into intervals, where up to $\frac{1}{3}$ of the input x the linear region is multiplied by 2, between $\frac{1}{3}$ to $\frac{2}{3}$ of the input values, the slight distortion is already produced, and above $\frac{2}{3}$ is output set to 1 [1], as is seen on Fig. 1.4, meanings the overdrive function combines both soft and hard clipping.

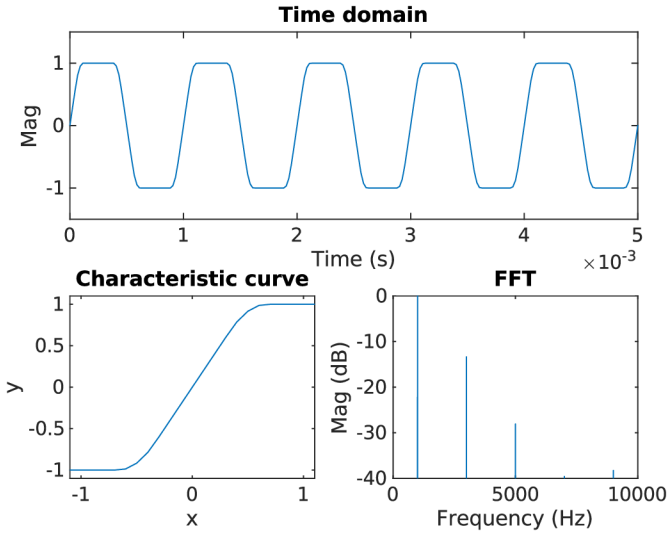

Fig. 1.4: Overdrive effect

1.3.5 Tube Saturation

Valves or vacuum tubes are active components used for amplifying. Despite being replaced by semiconductors, they still play an essential part in hi-fi. The tube saturation was used as a representative of an asymmetrical clipping. The characteristic curve of tubes varies from the type, and should consider the valve amplifier circuit, speaker and cabinet influences. One of the many simulations can be given by 1.9 where Q is a work point that controls the linearity of the transfer function for low input levels, and d is the amount of distortion. [1]

$$f(x)\begin{cases} \frac{x-Q}{1-e^{-d\cdot(x-Q)}}+\frac{Q}{1-e^{d\cdot Q}}, & Q\neq 0, x\neq Q, \\ \frac{1}{d}+\frac{Q}{1-e^{d\cdot Q}}, & x=Q, \end{cases} \tag{1.9}$$

As seen in Fig. 1.5, the negative input values are mostly cut off, and positive values are approximately linear. The signal looks destroyed at first, and there are visible changes in its spectrum, even though this signal modification is not as disturbing to a listener as the symmetrical hard clip. In comparison to the previous figures, the appearance of order higher harmonic is more significant here. After using this function, proper lowpass and highpass filtration is recommended to achieve a more authentic tube sound.

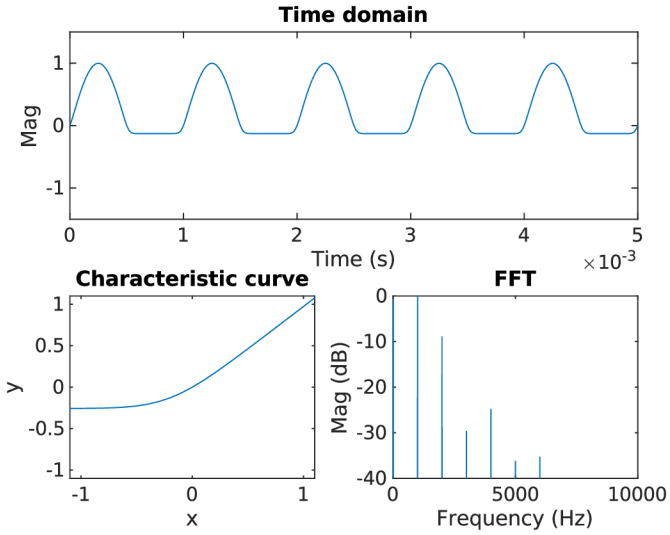

Fig. 1.5: Representation of the tube simulation

1.3.6 Tape Saturation

Magnetic tapes produce soft, symmetrical clipping, which is almost unnoticeable to an average listener, yet it adds harmonics to the final sound, specifically third and other odd-order harmonics. The gain factor is generated based on the characteristic curve of the input level and is used for compressing signals, while for low amplitudes, it works as an amplifier. The tapes gradually progress into distortion. [1] [6]

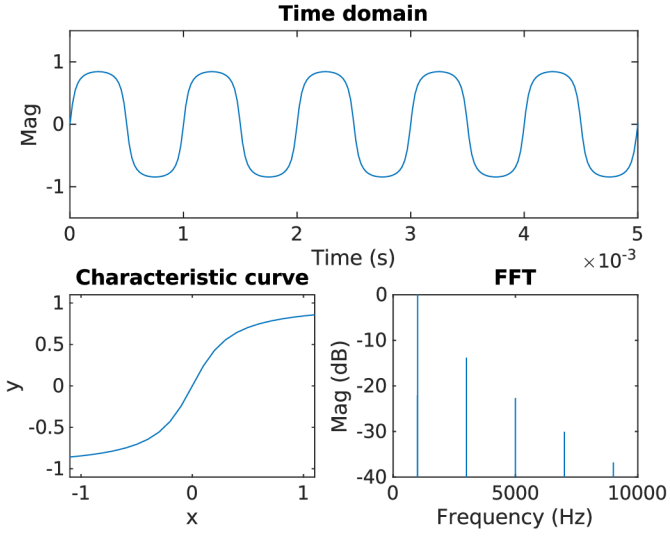

Fig. 1.6: Representation of the tape simulation

As an example of the tape clipping for this thesis, an arctan function was used

for simulation 1.10. The parameter α specifies the gain.[6] In a fig. 1.6, $\alpha = 4$

$$y(n) = 2\pi \cdot \arctan(\alpha \cdot x(n)) \tag{1.10}$$

1.4 Dynamic range processing

Dynamics processing affects the signal dynamics and perceptually modifies its loudness and timbre. The audio signal dynamics are processed by amplifying devices where the analysis of the input signal controls the gain, primarily by compressors, expanders, limiters, and noise gates. Nevertheless, these effects will only be mentioned briefly as the thesis mainly focuses on static characteristics, mapping the instantaneous input to the instantaneous output.

The signal envelope is followed by amplitude detection. Either peak values of the signal or an averaged constant providing RMS value by measuring the signal's power are detected. [1] The dynamic behavior is influenced by parameters such as *attack time* and *release time* for the peak measurement approach and *averaging time* for RMS approach. Based on these parameters signal is then adjusted by a smoothing filter and gain controller. The main signal flow is delayed due to any possible delay in a side branch.

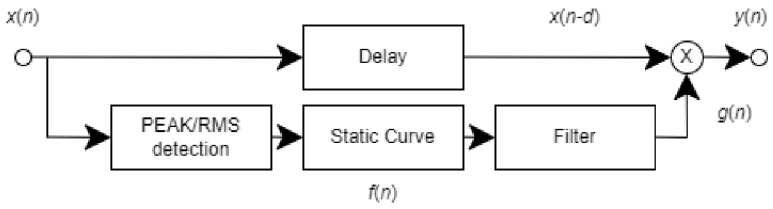

Fig. 1.7: Dynamics processing diagram (redrawn from [1])

1.4.1 Compressor

The compressor's main function is to reduce the signal's dynamic range above the threshold, which increases loudness without affecting the signal's maximum level output [15]. Compared to the limiter, the compressor does not use the infinite slope.

In the dB scale, the output can be expressed as a sum of input and gain. The gain is 0 dB up to a certain threshold (here -30 dB). Above this level, the characteristic follows a compression ratio. The ratio R specifies a relation between input level and output level, which is defined by delta $Y = \frac{1}{R} \cdot \Delta X$. The slope factor S may be defined instead of the ratio. (Fig. 1.8)

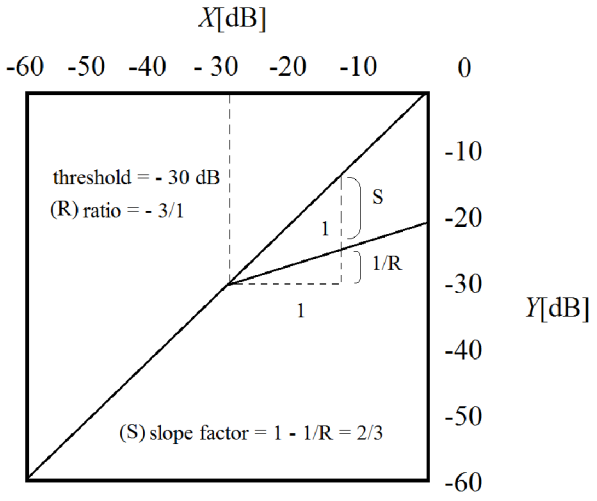

Fig. 1.8: Static characteristic of a compressor (redrawn from [1])

Compressors generally affect high levels or loud parts according to their characteristic curve. Quiet parts are not modified, but due to reducing level peaks and the dynamic range they seem to be much louder and stand out from the signal way more than before processing.

1.4.2 Expander

The expander is a system that increases the signal dynamics by attenuating the signal under a certain threshold; the rest remains unchanged. As in the compressor's case, the expander typically employs RMS level detectors of the input level with an averaging time [1] [15]. The ratio is usually applied, e.g., as 1:3, which denotes an upward-type expander amplifying input over the threshold by the value of 3. At a downward-type expander 3:1, the input below the threshold would be decreased accordingly.

The following figure 1.9 demonstrates the possible impact of the compressor and expander effect on a saxophone sample in both time and spectral domain while using parameters: compressor (threshold -25 dB, ratio 10, knee width 5 dB); expander

(threshold -40 dB, ratio 10, attack time 0.01 s, release time 0.02 s). These functions are found within the MATLAB Audio Toolbox.

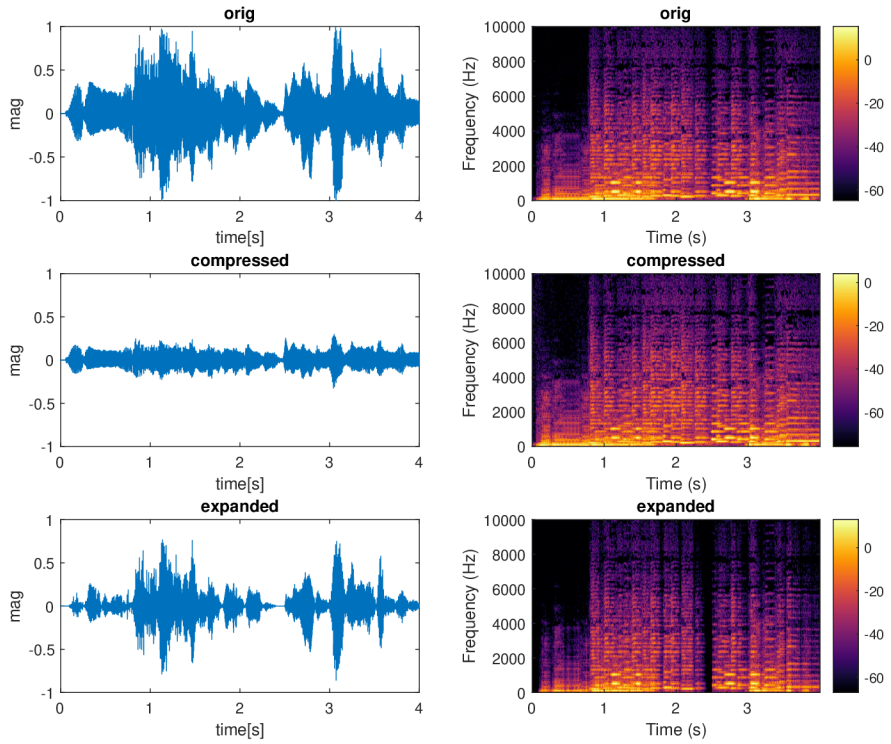

Fig. 1.9: Comparison of the compressor and expander

1.5 Sound and Musical Use

During the past decades, saturation effects have influenced many musical genres heavily. As guitar amplifiers were evolving, characteristic sounds were generated. Analog pedal guitar devices were developed based on saturated valve amps. The first ones were based on the products manufactured by world-known brands such as Mesa Boogie, Vox, and Marshall. [1] These effects were transferred into the digital world with the signal processing implementation. Typically distorted sounds of asymmetrical clipping can be heard in *Fuzz* effects, while *Distortion* resembles the most severe form of clipping. Soft clipping is mainly utilized in *Overdrive* effects and in all analog device simulations, which are commonly employed in the form of plug-ins: additional software used in mixing and mastering DAWs.

2 Quantization

Since signals in DSP have to be processed using finite arithmetic, the analog signal is first digitized, where quantization is an essential part of the process.

The signal is first sampled by the sampling frequency, which has to fulfill the Nyquist-Shannon theorem's condition to get the discrete digital signal (a sequence of numbers). The sampling frequencies are usually 32 kHz (based on the 15 kHz bandwidth of analog stereophonic FM broadcast), 44.1 kHz (CD standard established by the video recorders), 48 kHz (suitable for the video processing), and their respective integer multipliers (such as 88.2 kHz, 96 kHz, and 195 kHz).[15]

The range of finite possible values depends on the ADC's bit-depth to achieve the digital signal. The closest level from this range is assigned for each sample using the rounded signal value. [4] [14] The reproduced signal is never identical to the previous analog signals since the quantized signal is always distorted and affects DSP approaches.

2.1 Quantization Effect

The example of a quantized signal is shown in Fig. 2.1, where a low bit depth was simulated. For w-bit representation, there are 2^w of levels within the specified range where the quantization step is $\Delta = 2^{-(w-1)}$. Quantization error was computed by subtraction of quantized and original signal.

There are two different ways of quantization based on the placement of quantization levels. The quantization step Δ is constant, so the levels are distributed equally over the whole dynamic range as in Fig. 2.1, or the quantization levels are distributed non-uniformly across the range. [4] Professional audio devices utilize ADC with only linear quantization, meaning constant quantization step. Non-linear quantization is known from telecommunications applications. The resolution for devices is 16, 20, and 24-bit. [15] According to formula (2.1), a quantized signal $\mathbf{x}^q \in \mathbb{R}^N$ is computed:

$$(\mathbf{x}^{\mathrm{q}})_n = \mathrm{sgn}^+(\mathbf{x}_n)\Delta\left(\left\lfloor \frac{|x_n|}{\Delta} \right\rfloor + \frac{1}{2}\right) \tag{2.1}$$

where n denotes the sample index and function $\mathrm{sgn}^+(x)$ returns 1 for $x \geq 0$ and -1 for $x < 0$. [4]

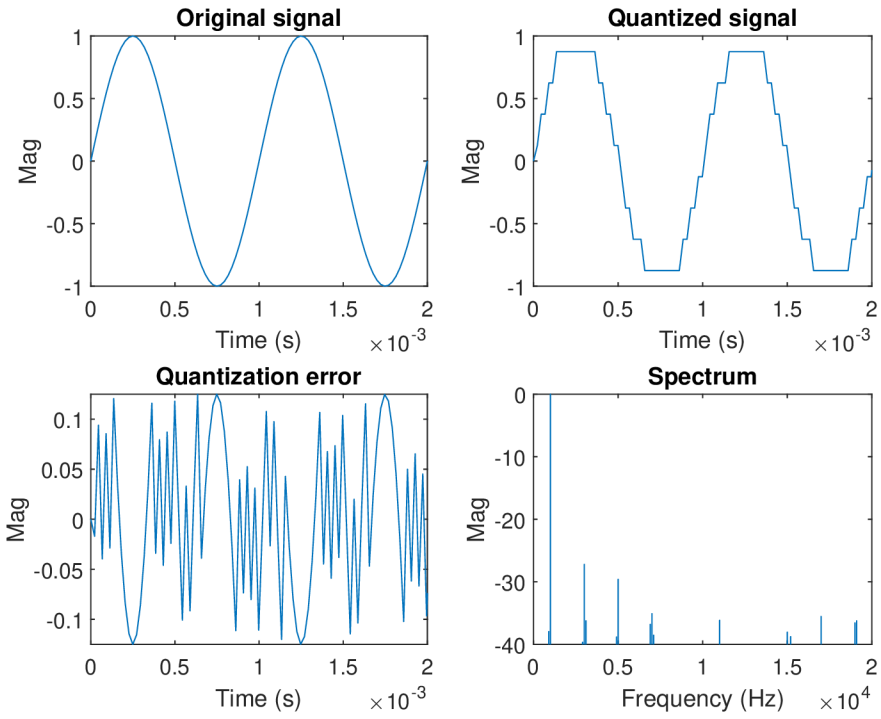

Fig. 2.1: Quantization error, 8 quant. levels (3 bit)

In electronic or experimental music, quantization is employed in an effect processor called *Bitcrush* to simulate the sound of the synthesizers and drum machines from the past century. Decimating input audio participates in the amount of distortion of the output signal. Decimating the sample rate and bit-depth input audio participates in the amount of distortion of the output signal. [29]

2.2 Dequantization

As implied, dequantization removes the unnecessary distortion from the quantized signal. In practice, hard declipping is related to dequantization since the principle of signal amplitude saturation can be considered a particular type of quantization. Dequantization (also called bit depth expansion) deals with a similar problem as the area of decompressing signals.

The whole process focuses on the task of restoring the original unknown signal. This task is nevertheless ill-posed, meanings there are infinitely many possible solutions, so additional information for solving this problem is required. Because the audio signal has in the time-frequency domain approximately sparse coefficients

(chap. 3.2.1), finding the signal with the sparsest coefficients whose samples do not lie further than $\frac{\Delta}{2}$ from the closest quantization level may be used for signal restoration. [3] Dequantization shares the principle with the hard-declipping algorithm. In the case of the hard clipping restoration, the algorithm operates only above the higher and lower threshold. Dequantization, on the other hand, within specified intervals. Further explanation is provided in the chapter about the soft-decliping approach (Chap. 3.3).

3 Restoration of Clipped Signal

The concept of *declipping* means reconstructing a distorted signal, which was limited, and getting the most similar signal to the original one before clipping. The goal is to get rid of any disturbing components and increase the subjective quality of the signal. [9]

To draw the line between two main modern approaches, the recovery model is considered *supervised* and *unsupervised* [19]. The supervised concept lies within the machine learning, trained on clean audio signals and using the deep neural networks (DNNs). Since they are trained on a specific data set, recovering is more specialized.

Still, most declip approaches are *unsupervised* or *blind* and assume predisposition of what audio signal should be like. They follow a path of choosing the *modeling domain* (time, analysis or synthesis ...), *generic model* (autoregressive model, sparsity ...), *model parameters* (specified from clipped signal, coefficients and the dictionary ...), *a criterion* (links the model parameters and observations to be optimized, e.g., clipped part consistency, reliable part consistency.), *suitable algorithm* (optimizes the model criterion) with a fixed number of iterations or conditions checking the convergence. This overview layout was defined for the evaluation of declipping methods in [19].

During past decades many different approaches to signal declipping have been widely discussed. The linear prediction offers another solution. Every discrete sample after signal discretization is expressed as a linear combination of previous samples; thus, the sample is predictable. [9] Recently, algorithms have achieved the best results using the principle of sparse signal representations. The various methods in audio hard-declipping are summed up briefly in the following section.

3.1 Hard-Declipping Methods

Restoring a Clipped Signal is the first mention of audio de-clip, which is dated back to 1991 when authors Abel and Smith recovered the clipped signal by solving a convex feasibility problem based on the assumption that the underlying signal has limited bandwidth relative to the sampling rate (i.e., that it was oversampled) [16] [12].

During the subsequent years, an article called *Statistical model-based approaches to audio restoration and analysis* involved a statistical model approach where missing

samples were replaced by the most probable ones based on the parametric signal model and Bayesian statistical signal processing. [9]

The first study, which included the sparse signal representation and was tested on general audio signals (not only speech and ultrasonic signals as in previous studies), was released in the article *A constrained matching pursuit approach to audio declipping*. The clipped samples were known beforehand, and the signal was processed in frames (each with a length of 64 ms and 75% overlap) and then utilized by an OLA method (Overlap-add approach). The modified OMP (Constrained Orthogonal Matching Pursuit) algorithm with DCT dictionary found the sparsest solution. The results of SNR showed average signal improvement compared to the distorted signal of around 4.5 dB. [20]

Recovering a Clipped Signal in Sparseland by authors Weinsten and Wakin also followed sparse representation using ℓ_1 minimization and DFT dictionary in algorithm *Reweighted ℓ_1 minimization with clipping constraints* where the sparse vector is weighted in every iteration. The article also incorporated another algorithm with less computation time *Trivial Pursuit with Clipping Constraints*. Using DFT, the Fourier spectrum coefficients are found, and their values are determined from the original spectrum by the least-squares method. If the original signal's sparsity k is known beforehand, the signal is reconstructed by founding the biggest harmonics of the saturated signal. If the number of k is unknown, the greedy algorithm is utilized for the iterative reconstructing of harmonic components. [12]

The study that incorporated for the first time the psycho-acoustic model was by B. Defraene and authors [21] is called *Declipping of Audio Signals Using Perceptual Compressed Sensing*. Finding missing components is done by ℓ_1 minimization and DFT dictionary. The frames are $N = 512$, and the Hanning window is used instead of the square one as in previous approaches. The aspects of human hearing, such as the absolute hearing threshold and masking, were employed in a single parameter *instantaneous global masking threshold*. The algorithm then computes this parameter by the standard of MPEG-1 from each processed frame and is later utilized in the minimization task. The results were evaluated by SNR, PEAQ, ODG, and subjective tests.

In 2014, *Audio Declipping with Social Sparsity* approach by [22] implemented *social sparsity*. To limit the set of feasible solutions, the authors used the hinge squared function and solved the optimization task by the relaxed algorithm (F)ISTA. A tight Gabor frame based on a Hanning window with a length of 1024 samples was used like a dictionary. Also, the authors compare within the article various shrinkage op-

erators (respectively L *Lasso*, WGL *Windowed Group-Lasso*, EW *Empirical Weiner* and PEW *Persistent Empirical Weiner*), and the results were presented on various audio and speech signals using SNR comparison.

Audio Declipping via Nonnegative Matrix Factorization (NMF) from 2015 deals simultaneously with audio inpainting (missing samples in digital audio signals) and declipping. The method's principle is combining of the algorithm GEM (generalized expectation-maximization) used for model parameter estimation and Wiener filtration estimating signal reconstruction by model parameters. The algorithm runs 50 iterations and uses STFT computed via sin windows with 50% overlap and frame length of 1024 samples. The results showed reconstruction comparable with the Social Sparsity approach [9].

The authors (among others S. Kitić, the author of previous researches) present A-SPADE and S-SPADE (*SParse Audio DEclipper*) in their article *Sparsity and Cosparsity for Audio Declipping: A Flexible Non-convex Approach*. After proper adaptation of parameters, the introduced algorithm is suitable for both synthesis and analysis data model. Models are identical in case their dictionaries are square and inverse matrices. Also, the analysis model requires a tight frame as a dictionary. Testing proved S-SPADE among A-SPADE, Social Sparsity, and C-IHT the most satisfactory algorithm with the longest computation time (due to the iterative projection). On the other hand, A-SPADE was the fastest of the tested algorithms with just a bit worse results. [24] [9] As a dictionary was used DGT (Discrete Gabor Transform) with the Hamming window (length of 1024 samples and 75% overlap).

A recent study [25] provides the solution for the de-clipping problem using a dictionary learning algorithm. While other approaches in the sparsity-based field use fixed dictionaries (DCT or Gabor), dictionary learning has already proved previously successful results in audio inpainting or denoising. [25]

The latest successful finding has been described in a paper [23] from 2022, where the authors enhanced the sparsity-based inconsistent audio de-clipping method (SS PEW - namely Social Sparsity with Persistent Empirical Wiener shrinkage) through cross-fading the signal's region. Research challenges the best performing de-clipper NMF, specifically in calculation time, is around 15 times shorter.

3.2 ℓ_1 Optimization

The following section will outline the basics of sparse signal representation, sparse synthesis, proximal operators, the corresponding algorithm, and other necessary resources for solving the de-clipping problem. The latter mentioned were taken from the hard declip implementation, and are also necessary for utilizing the thesis's primary focus; the soft declip.

Since there is an infinite number of solutions, the declipping task is ill-conditioned. Moreover, it leans on further signal information. Sparsity-based approaches approximate the signal by a few synthesis coefficients. Finding the sparse solutions is NP-hard, so a suitable approximation algorithm is used (greedy-type, convex minimization, or a combination of both). [3] The thesis focuses on a convex formulation of declipping in an STFT/Gabor domain.

For a better explanation of the following sections, the symbols, and abbreviations are described.

The scalar values are represented by italic letters (such as m, N), vectors are described by bold letters - $\mathbf{x}$, $\mathbf{y}$ and considered as column vectors in which the first element's index starts with number one, i.e., $\mathbf{y} = [y_1,\ y_2, y_3, ...y_n]$. The total number of elements of the vector or a set is marked as an absolute value, e.g., $|\mathbf{x}|$ = {-6, 0, 2, 7} = 4. [5]

Capital bold letters stand for matrices ($\mathbf{A}$, $\mathbf{B}$) and $\overline{c}$ italic letters with an overline for complex numbers.

A vector support is defined by $\text{supp}(\mathbf{x}) = \{i|x_i \neq 0\}$ [5] as a set of indexes where the vector contains nonzero coefficients, i.e. for a signal $\mathbf{x} = [x_1, \ldots, x_8]$ = [0, 0, 3, 0, 4, 2, 0, 5] is $\text{supp}(\mathbf{x})$ = {3, 5, 6, 8} and $|\text{supp}(\mathbf{x})| = 4$. [5] The vector space is labeled by double struck capital letters ($\mathbb{C}$) and is considered as a nonempty space.

3.2.1 Signals Sparsity

For systems of linear equations, there is an infinite number of solutions, and sparse solutions are a subset of those. While searching for the sparsest solution, finding the vector with the lowest number of nonzero coefficients is required [7].

For solving the following equations, the norm of a vector needs to be defined. The vector's norm is a real function that describes a nonzero vector's value by assigning

the positive real number to the vector. The ℓ_p norm of a vector $\mathbf{x} \in \mathbb{C}^N$ is defined as (3.1) and marked by $\|\mathbf{x}\|$ [7].

$$\|\mathbf{x}\|_p := \left(\sum_{i=1}^{N} \|x_i\|^p \right)^{\frac{1}{p}} \; s.t. \; 1 \leq p < \infty, \|\mathbf{x}\|_0 := |\mathrm{supp}(\mathbf{x})|, \tag{3.1}$$

The best-known form of the ℓ_p norm is probably the Euclidean norm for $p = 2$, defining euclidean spaces. If $p = 1$, ℓ_1 norm (in formulas indicated by $\|\cdot\|_1$) represents the complete sum of a vector's absolute coefficient's values. The definition of ℓ_0 norm and $\|\cdot\|_0$ is the total number of vector's nonzero coefficients. This thesis uses ℓ_1 norm as a part of convex optimization methods (chap. 3.2.2). A definition of a k-sparse vector has the most of the k nonzero coefficients, so the vector $\mathbf{x} \in \mathbb{C}^N$ is then k-sparse if meeting the condition: [5]

$$\|\mathbf{x}\|_0 \leq k, \tag{3.2}$$

3.2.2 Frames and sparse synthesis

A discrete signal can be expressed as a product of the coordinate vector and a matrix. The process of converting the signal to this form is called *transformation* (i.e., Fourier transform), where the matrix is then the *transformation matrix*.

A *sparse synthesis* applies that $\mathbf{Ax} = \mathbf{y}$, the $\mathbf{y}$ is any signal, $\mathbf{A}$ is the transformation matrix, and $\mathbf{x}$ is an assumed sparse vector of coordinates. It is called synthesis because the resulting vector $\mathbf{y}$ is formed from various components.

In contrast, in the *sparse analysis* (or *co-sparse analysis*), the signal $\mathbf{y}$ is not formed from single vectors; instead, the analysis of signal $\mathbf{y}$ produces sparse vector $\mathbf{z}$, then the $\mathbf{\Omega y = z}$. Since both synthesis and analysis models are identical if their dictionaries are square and inverse matrices, the expression $\mathbf{A = \Omega^{-1}}$ is valid. [9] [24]

The vectors are usually finite in signal processing; thus, linear algebra is used. The vector space $\mathbb{V}$ is a non-empty set consisting of basic elements; vectors. The vector $\mathbf{x} \in \mathbb{V}$ can be expressed by a linear combination of a generator system $\mathbf{E}$ (a subset of vector space $\mathbb{V}$), where $\mathbf{E}$ is a matrix with generating vector within each column. The vector $\mathbf{x}$ can have multiple representations since there are more generating vectors than the dimension of the vector space. If the vector $\mathbf{x}$ can be expressed in its generator system $\mathbf{E}$ as

$$\mathbf{x} = c_1\mathbf{e}_1 + c_2\mathbf{e}_2 + ... + c_n\mathbf{e}_n = \mathbf{E}\mathbf{c}, \tag{3.3}$$

then the scalars c_i represent the coordinates of $\mathbf{x}$ in $\mathbf{E}$, corresponding with the previously mentioned synthesis.

The *basis* is described as the smallest set of vectors whose linear combination can express any vector of $\mathbb{V}$ - the minimal generator system $\mathbf{E}$. The dimension of $\mathbb{V}$ equals the number of its basis vectors [9]

If other vectors are added to the basis of $\mathbb{V}$, the subset of generators is created, which is bigger than the $\mathbb{V}$ dimension. Therefore generators are linearly dependent and still may represent any vector in vector space $\mathbb{V}$. Such a set is called the frame. Frames provide a redundant, stable way of representing a signal. [8] [9]

For a redundant generator system in $\mathbb{V}$ to form a frame, the set of vectors has to follow a formula (3.4) and a condition $0 < A \leq B < \infty$, where A and B defines frame bounds and frame elements $\mathbf{F}_k$ are called atoms.

$$A\|\mathbf{x}\|^2 \leq \sum_{k\in\mathbb{I}} |\langle \mathbf{x}, \mathbf{F}_k\rangle|^2 \leq B||\mathbf{x}||^2, \forall \mathbf{x} \in \mathbb{V} \tag{3.4}$$

If bounds are $A = B = 1$, a Parseval tight frame is achieved, which is used later in the implementation part of this thesis.

The Fourier Transform shows the spectrum characteristic of the whole signal at once. This fact does not correspond with the human hearing's ability to differentiate spectral changes concerning time. The usage of a Short Time Fourier Transform (STFT) is then essential; thus, Gabor analysis is used. A time-frequency analysis in $L^2(\mathbb{R})$ is based on translation and modulation operators. Based on operators, the Gabor analysis represents the function of a vector space $f \in L^2(\mathbb{R})$ as a superposition of translated and modulated versions of windowed, generator, fixed-function $g \in L^2(\mathbb{R})$, also known as a window function (Gaussian window for infinite time range signal and Hann's, Hamming or Blackman's window is most frequently for STFT usage). A set of such functions is called the *Gabor system.* (3.5) [9]

$$\left\{\mathrm{e}^{j2\pi mbx} g\left(x - na\right)\right\}_{m,n\in\mathbb{Z}} \tag{3.5}$$

The system with function $g \in L^2(\mathbb{R})$ and the translation and modulation parameters a and b create the frame in the space of $L^2(\mathbb{R})$. [7]

The form of displaying Gabor coefficients is the *spectrogram*, the frequency-time dependency graph with sufficient color bar denoting the values of the coefficients.

3.2.3 Convex Optimization Methods

Finding the sparsest solution requires defining the minimization task. In a system of linear equations $\mathbf{Ax} = \mathbf{y}$, the vector $\mathbf{x}$ is the wanted, an appropriate solution, and a representation of the vector $\mathbf{y}$. If $\mathbf{x}$ contains the least nonzero coefficients, it is the sparsest one. The vector $\mathbf{y} \in \mathbb{C}^m$ is a known result, and $\mathbf{A} \in \mathbb{C}^{mxN}$ is a matrix called a *dictionary*, where $m < N$. Then the problem is specified by:

$$\min_x \|\mathbf{x}\|_0 \; s.t. \; \mathbf{Ax} = \mathbf{y}, \tag{3.6}$$

In practice, statement (3.6) is inefficient because every $\mathbf{x}$ that meets the condition of $\mathbf{Ax} = \mathbf{y}$ is called feasible solution - feasible representation of the vector $\mathbf{y}$, meaning based on the linear algebra that under the above conditions on the matrix $\mathbf{A}$, there is an infinite number of feasible solutions that form an affine space. [7] Therefore a compromise between computational accuracy and time was found. The processed signal is noisy and causes a deviation δ while the computation of $\mathbf{Ax} = \mathbf{y}$. In most cases, p is considered $p = 2$. [5]. Because the ℓ_0 norm is not convex, it can not be used for convex optimization methods and proximal algorithms. The closest convex optimal solution is choosing the ℓ_1 norm instead. If the noise is present in the signal, the relaxation in the statement (3.6) is considered. The norm $||\cdot||_0$ is replaced by $||\cdot||_1$ (3.7).

$$\min_{\mathbf{x}} ||\mathbf{x}||_1 \; s.t. \; ||\mathbf{Ax} - \mathbf{y}||_2 < \delta \tag{3.7}$$

In practice, this solution does not always provide the sparsest results, but in most cases, the solution of ℓ_0 - minimization and ℓ_1 - minimization is almost identical. [7]

The hard clipped signal's restoration problem needs to be solved by convex formulation. For declipping itself, the sparse synthesis in Short-Time Fourier Transform (STFT or also known as Gabor) is used. If it is assumed that discrete signal $\mathbf{y}$ was hard clipped, then $\mathbf{y}$ can be separated into several parts - to the samples above threshold θ_{H}, the value of θ_{H} is assigned, and the samples below θ_{L} are substituted by θ_{L}. Any sample between θ_{H} and θ_{L} is called reliable because it corresponds with the sample of the original non-clipped signal of the same index. The signal $\mathbf{y}$ is sparse and formally written by $\mathbf{y} \approx G\mathbf{c}$ as a product of the linear Gabor synthesis operator

G and a vector $\mathbf{c}$. After employing the ℓ_1 norm, the optimization hard-declipping problem is formulated as [3]:

$$\arg\min_{\mathbf{c}} ||\mathbf{c}||_1 \text{ subject to } \begin{cases} M_{\mathrm{R}} G\mathbf{c} = M_{\mathrm{R}}\mathbf{y}_c, & \text{reliable samples,} \\ M_{\mathrm{H}} G\mathbf{c} \geq \theta_{\mathrm{H}}, & \text{samples clipped over the threshold,} \\ M_{\mathrm{L}} G\mathbf{c} \leq \theta_{\mathrm{L}}, & \text{samples clipped below the threshold} \end{cases} \tag{3.8}$$

$M_{\mathrm{R}}, M_{\mathrm{H}}$, and M_{L} represent masks: projection operators of clipped and reliable samples. Denoting the sets R, H, and L according to the three conditions of (3.8) as:

$$R = \{\mathbf{c}|M_{\mathrm{R}} G\mathbf{c} = M_{\mathrm{R}}\mathbf{y}_c\}, H = \{\mathbf{c}|M_{\mathrm{H}} G\mathbf{c} \geq \theta_{\mathrm{H}}\}, L = \{\mathbf{c}|M_{\mathrm{L}} G\mathbf{c} \leq \theta_{\mathrm{L}}\} \tag{3.9}$$

then masks indexes of a set $R \cup H \cup L$ are selected. The formulation (3.8) then searches for approximately sparse coefficients and generate signal consistent with the time-domain constraints [3]

However, there is no known way of effectively solving this (3.8) mathematical problem. Therefore after rewriting it in an unconstrained form (3.10) (where the variable may take any value, unlimited form), this problem can be sufficiently solved by a *proximal splitting* algorithm.

$$\arg\min_{\mathbf{c}\in\mathbb{C}^N} ||\mathbf{c}||_1 + \iota_R(\mathbf{c}) + \iota_H(\mathbf{c}) + \iota_L(\mathbf{c}), \tag{3.10}$$

Indicator function ι_C is formally written as 3.11:

$$\iota_C(\mathbf{x}) = \begin{cases} 0 & if\ \mathbf{x} \in C \\ +\infty & if\ \mathbf{x} \notin C \end{cases} \tag{3.11}$$

where ι_C values are 0 if the function's argument belongs to the set C and $+\infty$ if the argument lies outside the set C, where $C \subset \mathbb{R}^N$.

For solving (3.10) by proper proximal algorithm, the proximal operator of a convex function f for every $\mathbf{x} \in \mathbb{R}^N$ is defined as a solution of minimization task:

$$\underset{f}{\mathrm{prox}}(\mathbf{x}) \ = \arg\min_{\mathbf{y}\in\mathbb{R}^N} f(\mathbf{x}) + \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 \tag{3.12}$$

The essential proximal operators are, in our case, operators for the ℓ_1 norm of a vector and for set's indicator function ι_C.

The proximal operator for ℓ_1 of a vector $\mathbf{x}$, formally written as $\lambda\|\mathbf{x}\|_1$, is *soft thresholding*, where the threshold is $\lambda > 0$ and in the case of symmetrical clipping is defined as $\text{soft}_\lambda(\mathbf{x})$. Soft thresholding takes the vector argument and performs elementwise mapping [3] [9].

$$\text{soft}(\mathbf{x}) \begin{cases} \mathbf{x} - \underline{\lambda}, & if\ \mathbf{x} < \underline{\lambda}, \\ 0, & if\ \mathbf{x} \in \langle \underline{\lambda}, \overline{\lambda} \rangle, \\ \mathbf{x} - \overline{\lambda}, & if\ \mathbf{x} > \overline{\lambda} \end{cases} \tag{3.13}$$

The proximal operator of the indicator function $\iota_C(\text{prox}_{\iota_C}\mathbf{x})$ is then the projection onto set C $\text{proj}_C(\mathbf{x})$. The projection of the C set then moves $\mathbf{x} \notin C$ to the closest place from C and $\mathbf{x} \in C$ [9].

3.2.4 Douglas - Rachford Algorithm

The unconstrained form can include many optimization problems where the sum of convex functions is minimized. The basic unconstrained form is the following (3.14):

$$\underset{x \in \mathbb{R}^N}{\text{minimize}}\, f_1(x) + \ldots + f_m(x) \tag{3.14}$$

where $f_1, \ldots f_m$ are convex functions of $\mathbb{R}^N$.

The declipping task may be solved upon other approaches by a *forward-backward* algorithm that deals with the task in (3.15).

$$\min_{x \in \mathbb{R}^N} f_1(x) + f_2(x) \tag{3.15}$$

Both functions should be convex, and one of these is differentiable with a β-Lipschitz continuous gradient ∇f_2 [26]. If meeting the latter conditions, it is possible to find at least one solution for $\gamma \in (0, +\infty)$ in (3.16)

$$\mathbf{x} = \text{prox}_{\gamma f_1}(\mathbf{x} - \gamma \nabla f_2(\mathbf{x})) \tag{3.16}$$

If the formula (3.16) iterates for values of the step-size parameter γ_n , the forward-backward splitting algorithm converges to the exact solutions (3.17), where

prox refers to the backward (implicit) algorithm's step with function f_1 and $\mathbf{x}_n - \gamma_n \nabla f_2(\mathbf{x}_n)$ to the forward (explicit) gradient step with function f_2 of the algorithm. [26]

$$\mathbf{x}_{n+1} = \underbrace{\operatorname{prox}_{\gamma f_1}}_{\text{backward step}} \underbrace{(\mathbf{x}_n - \gamma_n \nabla f_2(\mathbf{x}_n))}_{\text{forward step}} \tag{3.17}$$

In the Douglas algorithm, the proximal operator compensates for the disadvantage of having one of these functions differentiable. Its solution for $\gamma \in (0, +\infty)$ is described by conditions: [26] [9]

$$\begin{aligned} &\mathbf{x} = \operatorname{prox}_{\gamma f_2} \mathbf{y} \\ &\operatorname{prox}_{\gamma f_2} \mathbf{y} = \operatorname{prox}_{\gamma f_1} \left(2\operatorname{prox}_{\gamma f_2} \mathbf{y} - \mathbf{y}\right) \end{aligned} \tag{3.18}$$

3.2.5 Solution using the Douglas-Rachford algorithm

If the equation 3.10 is rewritten as a sum of two convex functions, the unconstrained form is as follows (3.19) and can be solved by a proximal, iterative algorithm (DR). It produces a sequence of vectors that converges to the minimizer of the sum of these two functions.

$$\arg\min_{\mathbf{c}} \|\mathbf{c}\|_1 + \iota_C(\mathbf{c}) \tag{3.19}$$

To solve this problem accordingly, the projection onto a set C should be performed, where set C gathers all the coefficients from $M_\mathrm{R}, M_\mathrm{H}$, and M_L into a single set (3.9). For practical application, bounding vectors $\mathbf{b}_\mathrm{L}$ and $\mathbf{b}_\mathrm{H} \in \mathbb{R}^M$ of the set C need to be defined, where i denotes the sample index number. (3.20) [9].

$$(\mathbf{b}_\mathrm{H})_i = \begin{cases} \mathbf{y}_i & for\ i \in M_\mathrm{R} \\ \theta_\mathrm{H} & for\ i \in M_\mathrm{H} \\ -\infty & for\ i \in M_\mathrm{L} \end{cases} \quad (\mathbf{b}_\mathrm{L})_i = \begin{cases} \mathbf{y}_i & for\ i \in M_\mathrm{R} \\ \infty & for\ i \in M_\mathrm{H} \\ \theta_\mathrm{L} & for\ i \in M_\mathrm{L} \end{cases} \tag{3.20}$$

If, e.g., the j-positioned sample is reliable (does not lie within the saturated masks), it will appear in vectors $\mathbf{b}_\mathrm{L}$ and $\mathbf{b}_\mathrm{H}$ in j-position, so $(\mathbf{b}_\mathrm{L})_j = (\mathbf{b}_\mathrm{H})_j = \mathbf{y}_j$ [9]. Using those vectors is feasible for the determining C set as $C = \{\mathbf{c} | \mathbf{b}_\mathrm{L} \leq G\mathbf{c} \leq \mathbf{b}_\mathrm{H}\}$ which matches with possible solutions C [3] [9]. Finally, projection onto set C is described in (3.21):

$$\text{proj}_C(\mathbf{c}) = \mathbf{c} + G^{+}(\text{proj}_{[\mathbf{b}_\text{L},\mathbf{b}_\text{H}]}(G\mathbf{c}) - G\mathbf{c}) \tag{3.21}$$

where $\text{proj}[\mathbf{b}_\text{L}, \mathbf{b}_\text{H}]$ denotes:

$$\text{proj}[\mathbf{b}_\text{L}, \mathbf{b}_\text{H}](\mathbf{y}) = \min(\max(\mathbf{b}_\text{L}, \mathbf{y}), \mathbf{b}_\text{H}) \tag{3.22}$$

After applying all of the statements mentioned above, the final Douglas-Rachford algorithm is performed in (Alg. 1), where the convergence speed is specified by parameter γ [3].

Algorithm 1 Douglas-Rachford algorithm

Input: set starting point $\mathbf{c}^{(0)}$, Set parameters $\lambda = 1, \gamma > 0$.

1: **for** $i = 0, 1, ...$ **do**
2: $\quad \tilde{\mathbf{c}}^{(i)} = \text{proj}_C \mathbf{c}^{(i)}$
3: $\quad \mathbf{c}^{(i+1)} = \mathbf{c}^{(i)} + \lambda(\text{soft}_\gamma(2\tilde{\mathbf{c}}^{(i)} - \mathbf{c}^{(i)}) - \tilde{\mathbf{c}}^{(i)})$
4: **end for**
5: **return** $\mathbf{c}^{(i+1)}$

3.3 Soft Declipping

Studies regarding soft-declipping are not as common as hard declipping approaches. There are various reasons: hard-clipping is way more destructive than soft-clip and may occur on more occasions; because it is relatively easy to revert the signal through the compensation curve since the characteristic of the nonlinear distortion is known. Furthermore, that soft clip is often required as an artistic tool in audio processing, so de-clipping here is not necessary.

Why is soft-declip needed then? It may compensate for real-world devices (such as amplifiers or magnetic recorders) that worsen the audio signal quality and original intent during recording, storing, or reproducing due to operating outside its linear range. [17] There are already studies and patents which focus on a compensation of the audio reproduction system in terms of time, frequency, phase, and transient response based on impulse responses and dealt with by modified DSP filters as well as additional analog electronic circuits. However, only a few of them mention this task as a soft-declip.

Previous attempts are studied in the work of F.R. Ávila, and collective [13] where authors propose a fully blind soft declipper for audio and speech signals where the characteristic curve is unknown. Their primary focus was on the signal's spectrum domain instead of the near sparsity of the signal. The weights are calculated solely from the distorted signal based on the weighted ℓ_1 norm, so it is suitable for various real-world usages and device simulations. The signal is separated into blocks, processed by DCT, and weighted and normalized by its energy. Weighting makes the nonlinear distortion more salient. Later the gradient-based optimization method is employed. [13]

Their other article [17] deals with the same task using the constrained weighted Least Squares for computing the exponential weighting function. Assumptions behind this approach were also spectrum profile based on observations as follows; salient frequency components of the original signal also remain in the degraded signal spectrum, the proposed weightening function behaves similarly to the inverse of the original signal, distortion increases the magnitudes of low-energy regions of the original spectrum, and weightening DCT components roughly measure the amount of nonlinear distortion in the original signal. [17] Results of the proposed weightening cost function show that ℓ_2-norm of distorted and original audio signal detects nonlinear distortion

3.3.1 Problem formulation

As was previously explained in the first chapters, the soft clipping transfer function does not break at the upper and lower threshold points as in hard clipping. A transition part around these points is relatively smoother. Since it is not this thesis purpose to be able to declip any soft-clipped signal, instead, it focuses on a signal restoration where the characteristic curve of the clipping function is known.

In the ideal case, the easiest way to de-clip is approximating by the inverse function of the saturation function $f^{-1}(\mathbf{y})$, where $\mathbf{x}$ is the original signal and $\mathbf{y}$ is the distorted signal. Since the nonlinear function applied on signal $\mathbf{y}$ is known, then $f^{-1}(\mathbf{y}) = \mathbf{x}$.

Applying the inverse soft-clip function is possible. However, due to the quantization, it leads to inaccurate results. Under normal circumstances, the uniform constant-step quantization, as was mentioned in chapter 2.1, is used in professional audio devices. After employing a soft-clip function that multiplies and pushes a higher number of samples to the limit values, more samples end up in the clipping

section. Any obtained sample after quantization is rounded to the closest quantization level. If any quantization level is denoted as l_i, where i is the number of quantization level and Δ is the quantization step, then the original signal magnitude $\mathbf{x}$ at a given time t belonged within the interval of $\mathbf{x}(t) \in \langle l_i - \frac{\Delta}{2}, l_i + \frac{\Delta}{2} \rangle$. Since the *knee* (amplitude curve of a soft clip) belongs under just a few quantization levels on y-axis, the equal interval on the input x-axis (number of affected samples) also increases depending on the shape of a nonlinear curve.

As implied, the soft-declipping task combines the inversion of the nonlinear characteristic curve with signal dequantization. Formally the problem is formulated:

$$\arg\min_{\mathbf{c}} ||\mathbf{c}||_1 \text{ s.t. } f^{-1}(l_i(n)) \leq [G\mathbf{c}]_n \leq f^{-1}(l_{i+n}(n)) \tag{3.23}$$

This task is a convex problem where sparse coefficients are sought and meet the condition, where the n-th sample (product of synthesis) is between the interval $f^{-1}(l_i(n))$ and $f^{-1}(l_{i+1}(n))$. The more the nonlinear curve reaches its maximum values and the flatter it is, the more the intervals widen. Towards zero, they are almost identical; the intervals will hardly change and be the same size as the original signal. So the n-th synthesis sample is conditioned by the quantization levels below and above.

3.3.2 Projection on feasible solutions

The condition 3.23 is utilized in the projection to satisfy those synthesis samples and apply the DR algorithm iteratively. Once the quantization levels with according *decision* levels $\mathbf{b}_\mathrm{H}$ $\mathbf{b}_\mathrm{L}$ are established and recomputed by the inverse saturation function f^{-1}, they determine limits for each n-th sample of $\mathbf{y}$. 3.24

$$(\mathbf{b}_\mathrm{H})_n = f^{-1}\left(\mathbf{y}(n) - \frac{\Delta}{2}\right), (\mathbf{b}_\mathrm{L})_n = f^{-1}\left(\mathbf{y}(n) + \frac{\Delta}{2}\right) \tag{3.24}$$

4 Implementation

This thesis aims to create a database of differently saturated signals for testing the implementation of soft clip restoration. Saturation simulations and signal restoration scripts were created in the computing environment MATLAB R2021a. For the successful run of the scripts, downloading The Large Time-Frequency Analysis Toolbox (LTFAT) is necessary. [27]

4.1 Program Solution

The primary part of this program is the file `soft_declip.m`, in which the restoration algorithm takes place. Other function files complement the main script. The script folder is detected and added with all subfolders to the path at the very start of the soft de-clipping script. After adding the LTFAT library to the same folder or changing the MATLAB directory, the command `ltfatstart` loads the LTFAT toolbox. The choice of either plotting the results or not is given right before the sample in .wav format is loaded using function `audioread`, with the sampling frequency $f_s =$ 44.1 kHz and bit depth of 16 bit. Samples were further normalized, and their length was specified in seconds.

4.1.1 Generated Signal

A simple signal was tested for the first test run of the restoration script soft_declip.m. It consists of a sum of three sinusoidal signals with frequencies: 600 Hz, 1 kHz, and 1.1 kHz and Gaussian noise. This reconstruction is shown in Fig.:4.3.

4.1.2 Audio Samples

The tested database consists of various audio signals (which are found in an attachment to this document). Furthermore, the usage and variability of the proposed restoring approach are demonstrated on them.

The audio database contains various types of short samples, consisting of speech, recordings of solo instruments, as well as more complex sounds, specifically: male

and female voice (1_male_speech.wav, 2_female_speech.wav), a recording of different kinds of solo instruments (in relation to the exciter), acoustic guitar playing chords (3_guitar.wav), a flute solo (4_flute.wav), a saxophone solo (5_saxophone.wav), a violin melody (6_violin.wav), orchestral part (7_orchestra.wav), a verse of a pop song (8_pop_band.wav) and electronic-music pattern (9_electronic.wav). This diverse choice mainly aimed to test different types of audio that could be subjected to soft-clip in real-life situations. Even though the restoration algorithms usually provide better results on less complex (sparse) audio signals (i.e., solo instruments or 4.1.1) as shown in the 4.3.

4.1.3 Signal clipping

The soft-clip function switching occurs after sample loading. These functions are saturation effects of *overdrive* [1], *distortion*, and simulations of *tube saturation* [1] and *tape saturation* which were introduced in the theoretical part (chapters 1.3.1, 1.3.2, 1.3.5, and 1.3.6). It is possible to adjust function parameters, such as the amount of the gain, threshold, or working point (in the Tube simulation's case). These parameters play a significant role in the level of distortion. It is advisable to determine this level so that the distribution of the clipped samples is uniform for testing purposes. The clipping parameters were computed from the SDR of testing samples, which was established between 3-5 dB.

The exception is the overdrive function threshold which, according to [1], is computed by Schetzen formula for symmetrical soft clipping and delivers the highest achieved SDR. Any lower or higher threshold number does not change the SDR due to the overdrive function structure, the diverse nature of the tested samples, and the shape of the transfer function does not resemble the soft clip. Illustrations of the impact of input parameters on SDR are shown in Fig. 4.1. After clipping the signal, the inverse de-clipping function is defined according to the clip type and later called by the function handle.

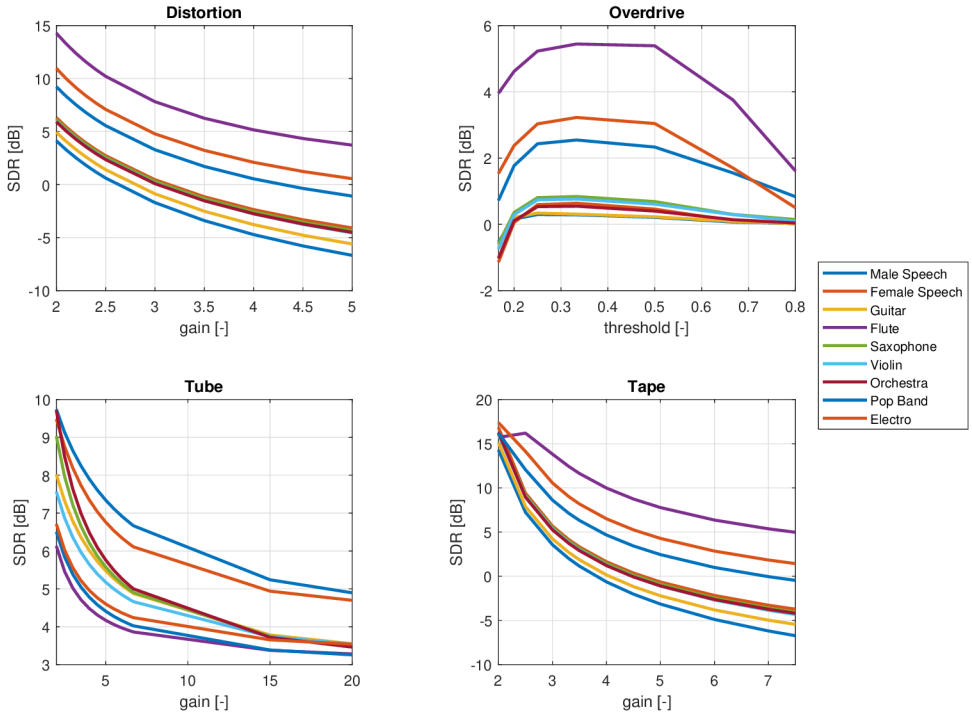

Fig. 4.1: SDR of clipped samples in relation with function parameters

4.1.4 Signal Restoration

Before the de-clipping, the clipped signal undergoes further quantization, set to 8 bit. The decision levels $\mathbf{b}_H$ $\mathbf{b}_L$ are computed from the quantized signal and the inverse clipping function.

The DGTreal frame is constructed with the help of LTFAT function `frame`, with the Hanning window of length $n = 1025$ and 50% overlap, which is employed in frame analysis operator - `frana`. The parameter *ite* determines the number of DR algorithm's iterations, which is set by default to 1000, primarily due to the time efficiency. The Fig 4.2 shows that even with a higher number of iterations, the algorithm's convergence is mostly unchanged.

Every iteration includes in the first step the `projection.m`, where the decision levels are compared with the product of frame synthesis operator `frsyn`, and affected samples are detected and mapped. The second step uses the proximal operator of soft thresholding. In the final iteration step, the objective function of ℓ_1 norm of coefficients c is computed, a counter adds +1, and moves to the next iteration.

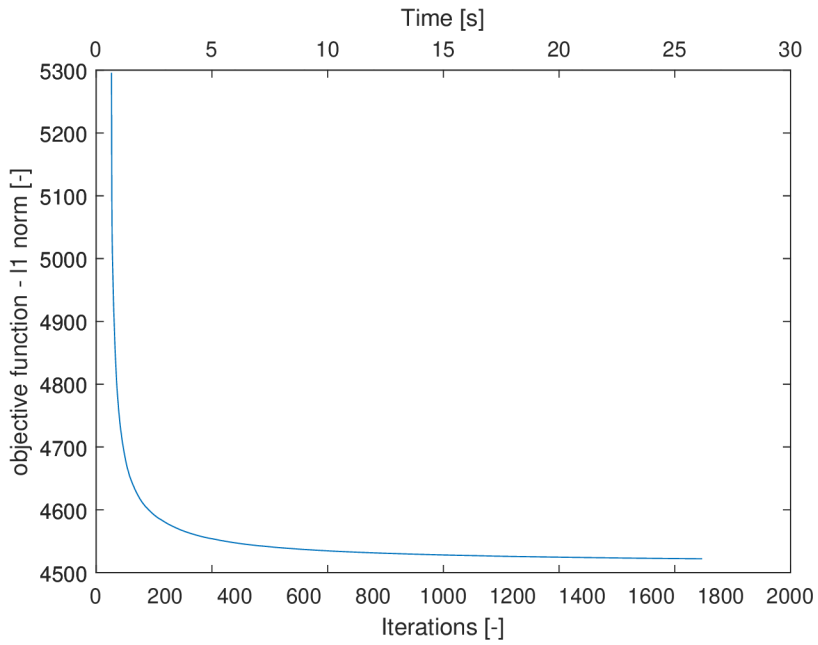

Fig. 4.2: ℓ_1 norm in time/iterations

The algorithm output coefficients are transformed back to the time domain by `frsyn` function. After a successful run of the algorithm, the reconstructed signal is received, and its waveform is plotted in the time and frequency domain using the function `spectrogram`.

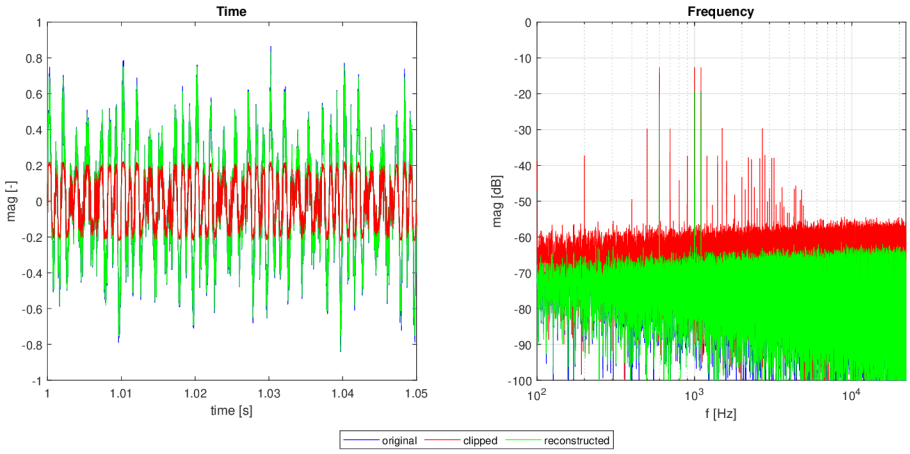

Fig. 4.3: Artificial signal restoration in time and frequency domain

The picture Fig. 4.3 shows the artificial signal consisting of 3 sinusoidal signals and Gaussian noise where distortion clipping with the parameter gain = 5 was applied. The green signal represents the result of the restoration algorithm that

is almost identical to the original signal in both the time and frequency domain. Time-domain displays clipped waveform without the amplification factor.

4.2 Evaluation Methods

There are several ways of evaluating audio signal quality. The objective methods deliver exact unbiased numerical results of the audio signal. Meanwhile, the subjective methods directed to the group of the instructed listeners show the end-user's relative preferences. The SDR and PEMO-Q evaluation (objective methods) and MUSHRA (subjective method) were selected to measure the signal's parameters and audible differences between the original, clipped, and restored signal.

4.2.1 SDR

Signal to distortion ratio (SDR) is the most commonly used metric for testing and comparing the similarity of given signals. SDR is equivalent to the signal-to-noise ratio (SNR), which describes the noise and signal level ratio. The SDR is computed using (4.1) and denotes the ratio of the original signal $\mathbf{u}$ and the clipped or restored signal $\mathbf{v}$. The unit is expressed in dB, and the bigger the value, the more similar signals are. [7]

$$\mathrm{SDR}(\mathbf{u},\mathbf{v}) = 10\log_{10}\frac{\|\mathbf{u}\|_2^2}{\|\mathbf{u}-\mathbf{v}\|_2^2}[\mathrm{dB}] \tag{4.1}$$

By computing the difference ΔSDR of the SDRs of the clipped signal and reconstructed signal (4.2), where $\mathbf{y}$ is the original signal, $\mathbf{y}_c$ denotes clipped signal, and $\mathbf{y}_R$ indicates restored signal, the approximate level of reconstruction is achieved.

$$\Delta\mathrm{SDR} = \mathrm{SDR}(\mathbf{y},\mathbf{y}_c) - \mathrm{SDR}(\mathbf{y},\mathbf{y}_R), \tag{4.2}$$

In figure 4.4 the SDR improvement is compared across all of the testing samples in relation to a different clip type.

According to the evaluation by SDRΔ, the most significant improvement is received from distortion and tape restoration, Fig. 4.4. On the other hand, tube saturation and overdrive show a minor improvement. Complex sounds such as orchestra, electronic part, or the pop band generally show better results than speech

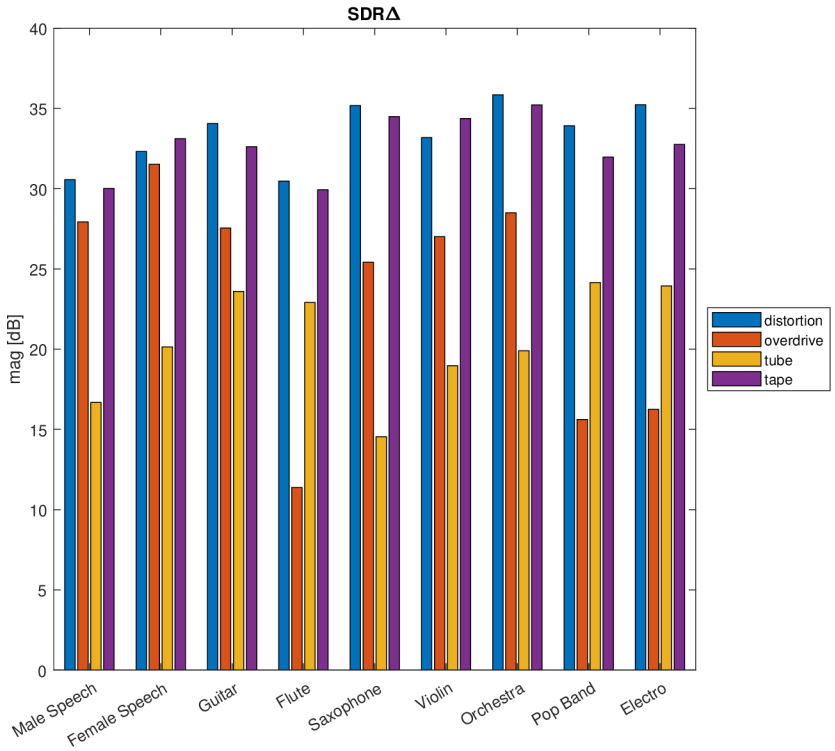

Fig. 4.4: SDRΔ improvement of reconstructed samples

and solo instruments, but it does not necessarily mean better sound quality and will be subjected to the listening test.

4.2.2 PEMO-Q

Since 1998 a tool for an objective method used for the measurement of the signal quality called *Perceptual evaluation of audio quality* (PEAQ) became ITU-R recommendation BS.1387. Its purpose was mainly focused on subjective quality ratings of low-bit-rate coded audio signals. Around 2006 the new method PEMO-Q (full name *Perception Model - Quality Assessment*) was built on a modified and expanded version of PEAQ. The method focuses on predicting the perceived quality degradation of wide-band audio signals to that of a reference signal and uses a psycho-acoustically validated auditory processing model. [7] [11]

The Matlab implementation of *PEMO-Q*[1] includes the function `audioqual` [32] that is called within the code for the testing purpose. Output arguments are PSM

[1]available for academic use and research until 2021 from the website: `https://www.hoertech.de/de/f-e-produkte/pemo-q.html`

(Perceptual Similarity Measure) - overall correlation between internal representations, PSMt (2^{nd} overall objective quality measure), ODG (Objective Difference Grade), and PSM_inst (vector of instantaneous objective quality). The ODG scale (tab. 4.1) has its origin in ITU-R and indicates a worsening of the objective signal quality.

0	imperceptible
-1	perceptible but not annoying
-2	slightly annoying
-3	annoying
-4	very annoying

Tab. 4.1: ODG Scale Evaluation

The following graph 4.5 very much complies with what was shown before. The values are upside down while applying the negative ODG scale, opposite to the SDRΔ comparison. According to the psychoacoustic model in general, the results of overdrive and tube are the least pleasant, and samples with the worst evaluation in every clipping type were guitar, saxophone, and violin samples. Surprisingly pop, female speech, and orchestra samples are considered perceptible but not annoying.

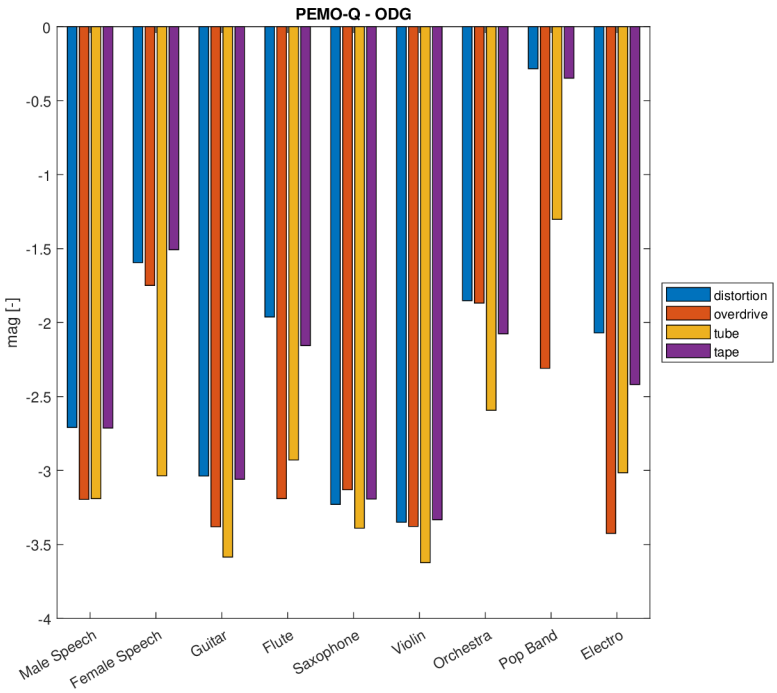

Fig. 4.5: PEMO-Q ODG evaluation

4.2.3 MUSHRA

A commonly used method for subjective comparisons of the signal quality is a listening test MUSHRA (*Multi Stimulus test with Hidden Reference and Anchor*). The MUSHRA method shows the respondent maximum of 15 stimuli, where the known reference, hidden reference (original signal), hidden anchor (clipped signal), and tested samples lie. The listener chooses the numbers from the interval 0 (bad) – 100 (excellent), and absolute results are then averaged. [10] The advantage of this method is that respondents can assess even very subtle differences between test samples. A total of 15 respondents aged 23-33 years with equal representation of females and males were selected to perform the test on one day in a quiet room with closed headphones (headphones: Ultrasone 840, soundcard: Steinberg UR242, PC: HP ProBook 450 G2). The test was created using the webMUSHRA package [30] and the following Fig.4.6 with the help of `multiple_boxplot.m` function [31]. Respondents mostly successfully detected the hidden anchor and the reference. Overall, there were minor differences between the four reconstructed clipping types. Distortion clipping was chosen for reconstruction using Izotope RX and was rated worse than reconstruction using the DR algorithm.

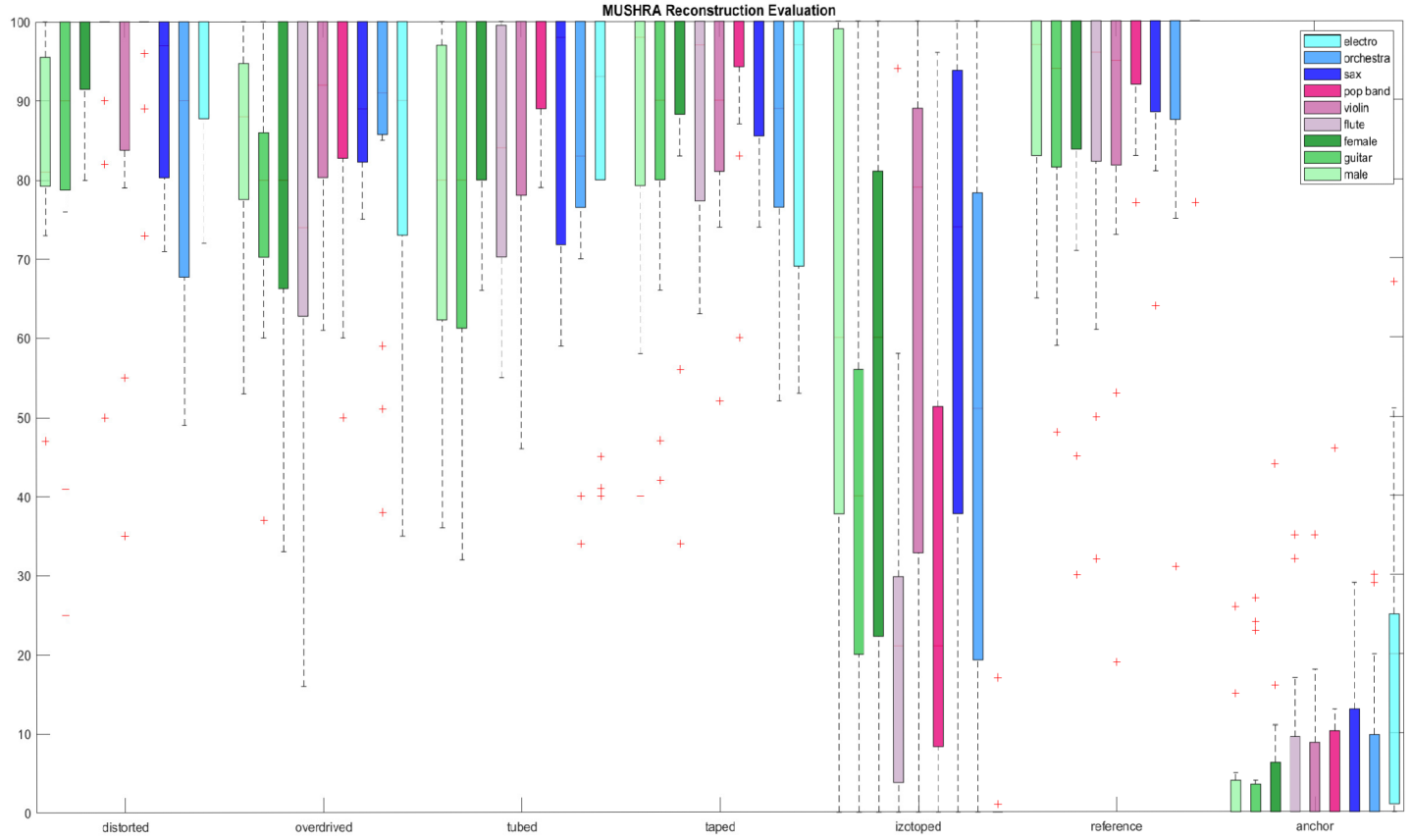

Fig. 4.6: *MUSHRA* evaluation results

4.3 Comparison with Commercial Software (Izotope RX)

This work aims to compare the reconstruction results with commercially available and widely used de-clipping software, Izotope RX. The same test samples as in the previous section were used in SDR, PEMO-Q and MUSHRA comparisons.

The latest available Izotope RX 9 is the repair bundle for restoring audio. It consists of various tools for de-reverbing, de-noising, de-humming, and de-clipping. *De-Clip* repairs, according to the online documentation [28], the digital and analog clipping artifacts, either if its hard-clipping from A/D converters or over-saturated tape. The program structure requires automatically or manually setting the threshold over a waveform displayed in a histogram meter where the threshold can be independently adjusted to positive and negative amplitude. The values above and below the threshold interpolate the waveform to a more "rounded" shape. In cases of severe distortion, where *De-Clip* is not sufficient, the manual recommends using modules *Deconstruct* or *Spectral Repair*. [28]

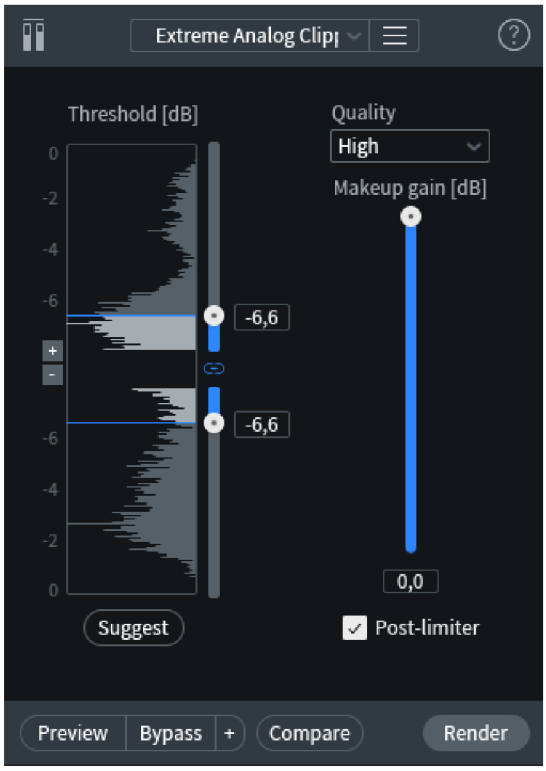

Fig. 4.7: *DeClip* by Izotope RX (printscreen)

Since Izotope exports de-clipped samples with the 16bit depth, the distorted

samples were quantized accordingly beforehand, so the quality results are not dishonested. However, in the MUSHRA comparison were these samples compared to the DR algorithm reconstruction from clipped samples quantized by 8 bit. Even so, the quality of the reconstruction was worse, according to the respondents. The evaluation using SDRΔ and PEMO-Q is shown in Fig. 4.8.

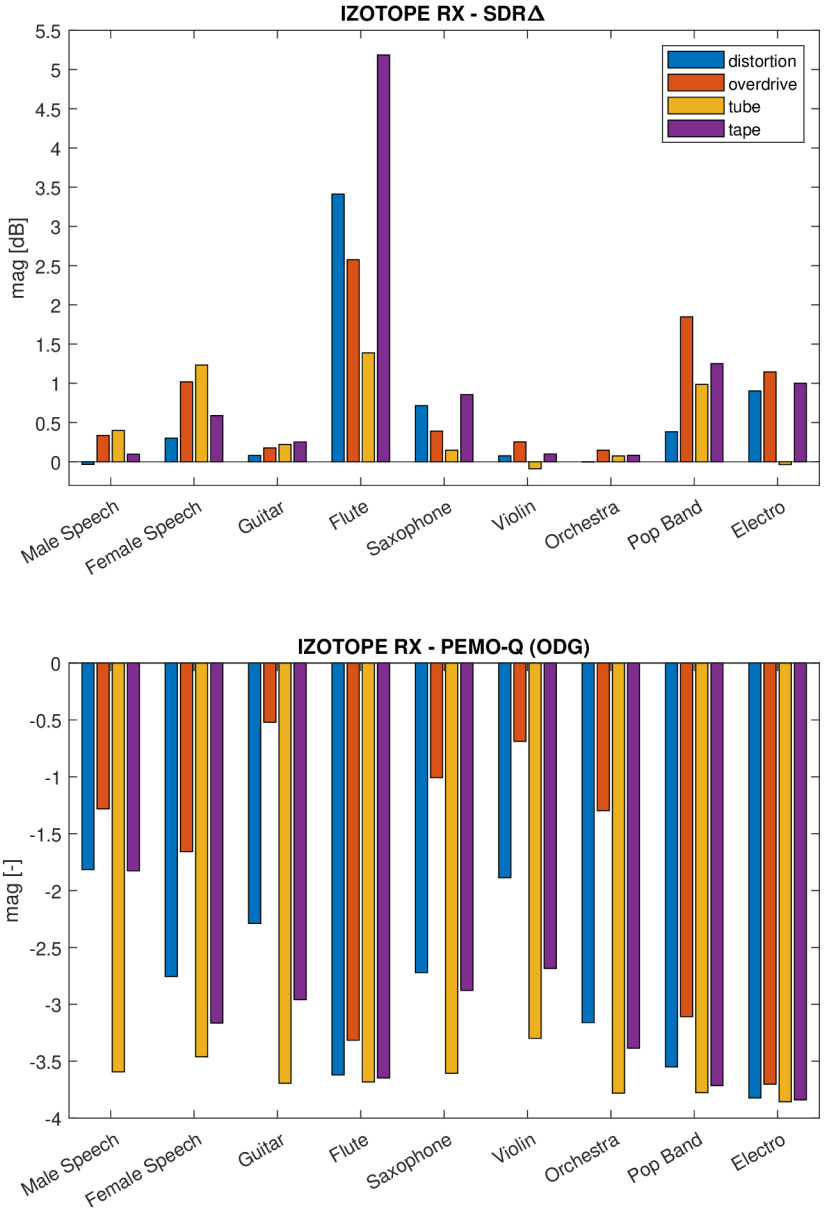

Fig. 4.8: *DeClip* restoration evaluation using SDRΔ and PEMO-Q

In a few cases (male speech and violin), the reconstruction was evaluated by

SDRΔ in negative values, so there was a decrease in signal to distortion ratio compared to the clipped signal rather than an increase. The highest SDRΔ values were achieved by the de-clipped flute, probably due to the distortion parameters that were set at the highest for this sample to achieve the desired SDR. According to the PEMO-Q assessment, the flute sample was, on the contrary, the qualitatively inferior one along with the complex recordings of pop band and electro. Overall, the de-clipping of asymmetrical tube distortion was rated -3.5. The reconstruction of samples clipped with distortion and overdrive was rated the best.

Conclusion

This work describes and observes the effect of reconstructing audio soft-clipped signals. The de-clipping uses the Douglas-Rachford algorithm previously used to restore hard-clipped signals with modified projection depending on the degree of quantization and type of distortion.

The first section describes an introduction to saturation and selected tested clipping functions, specifically simulations of overdrive and distortion effects, tube amplifier and magnetic tape distortion. The second part outlines the quantization issues that affect the soft-declipping task. The third section summarizes the various declipping approaches and describes the sparse signal theory and the de-clipping algorithm used. The fourth section covers the implementation performed in MATLAB using the LTFAT toolbox. The de-clipping of the selected test audio samples was then evaluated by both objective and subjective methods; SDRΔ, the PEMO-Q psychoacoustic model and the MUSHRA listening quality test. A comparison with the commercially available software Izotope RX was also performed.

At 1000 iterations, the algorithm showed a reconstruction speed of less than 15 seconds for a 5-second long audio sample. The comparisons showed reasonably good results mainly in the SDRΔ (improving by approximately 30dB in the case of distortion effect and tape saturation) and MUSHRA (where reconstructed samples were often confused with the reference) and less well in the PEMO-Q evaluation. The soft-declipping algorithm also showed better results in the SDRΔ and MUSHRA evaluations than the DeClipper from Izotope RX.

Bibliography

[1] ZÖLZER, Udo. *DAFX - Digital Audio Effects: Second Edition* [online]. 2nd ed. Hamburg, Germany: Wiley, [2011] [cit. 2020-11-16]. ISBN 978-0-470-66599-2. Dostupné z URL: <https://onlinelibrary.wiley.com/>.

[2] MUSIL Vladislav, J. BOUŠEK, M. HORÁK, O. HÉGR, *Elektronické součástky* Brno, 2017, [cit. 2020-11-17]

[3] RAJMIC P., P. ZÁVIŠKA, V. VESELÝ a O. MOKRÝ, *A New Generalized Projection and Its Application to Acceleration of Audio Declipping.*. Axioms [online]. 2019, 2019(105), 20 [cit. 2020-11-19] Dostupné z URL: <https://www.mdpi.com/2075-1680/8/3/105/>.

[4] ZÁVIŠKA, Pavel a Pavel RAJMIC. *Sparse and Cosparse Audio Dequantization Using Convex Optimization* [online]. 2020, 5 [cit. 2020-11-24]. Dostupné z URL: <https://www.researchgate.net/publication/343590223_Sparse_and_Cosparse_Audio_Dequantization_Using_Convex_Optimization/>.

[5] HRBÁČEK, Radek, Pavel RAJMIC, Vítězslav VESELÝ a Jan ŠPIŘÍK *Introduction to sparse signal representations* Elektro Revue [online]. 2011, 2011(5), 11 [cit. 2020-11-29]. ISSN 1213-1539. Dostupné z URL: <http://elektrorevue.cz/cz/download/ridke-reprezentace-signalu--uvoddo-problematiky/>.

[6] FERRARO, R. *ASSIGNMENT 2 – Analogue Tape Simulation in MATLAB* [online]. Sydney, 2020, [cit. 2020-12-01]. Dostupné z: <https://ses.library.usyd.edu.au/bitstream/handle/2123/22615/DESC9115%20-%20Analogue%20Tape%20Simulation%20in%20MATLAB%20-%20Written%20Report.pdf?sequence=1&isAllowed=y.>.

[7] BEŇO, Tomáš *Restoration of signals with limited instantaneous value using a psychoacoustic model* [online]. Brno, 2019, [cit. 2020-12-01]. Dostupné z: <https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=190525>. Master's Thesis. Brno University of Technology. Vedoucí práce Ing. Pavel Záviška.

[8] KOVAČEVIĆ, Jelena a Amina CHEBIRA *An Introduction to Frames. Foundations and Trends® in Signal Processing* [online]. Brno, 2007, [cit. 2020-12-01]. 2(1), 1-94. ISSN 1932-8346. Dostupné z: doi:10.1561/2000000006

[9] ZÁVIŠKA, Pavel *Restaurace audiosignálů založená na řídkých reprezentacích* [online]. Brno, 2017, [cit. 2020-12-01]. Dostupné z: <https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=147702>. Master's Thesis. Brno University of Technology. Vedoucí práce Doc. Mgr. Pavel Rajmic, Ph.D

[10] SCHIMMEL, Jiří. *Akustika a zvukové systémy.* Brno, 2018, [cit. 2020-12-03]

[11] HUBER, Rainer, Birger KOLLMEIER. *PEMO-Q—A New Method for Objective Audio Quality Assessment Using a Model of Auditory Perception* [online]. 2006, [cit. 2021-04-17]. Dostupné z URL: <https://ieeexplore.ieee.org/document/1709880>.

[12] WEINSTEIN, Alejandro J., Michael B. WAKIN *Recovering a Clipped Signal in Sparseland* [online]. 2011, [cit. 2021-05-16]. Dostupné z URL: <https://arxiv.org/pdf/1110.5063.pdf>.

[13] ÁVILA, Flávio R., Luiz W.P. BISCAINHO. *Audio Soft Declipping Based on Weighted L1-norm* [online]. 2017, IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), 2017, pp. 299-303, doi: 10.1109/WASPAA.2017.8170043 [cit. 2022-03-26]. Dostupné z URL: <https://ieeexplore.ieee.org/document/8170043>.

[14] MIŠUREC, Jiří, Z. SMÉKAL. *Číslicové zpracování signálů* Brno, 2011, [cit. 2022-03-30]

[15] SCHIMMEL, Jiří. *Studiová a hudební elektronika* Brno, 2015, [cit. 2022-03-30]

[16] ABEL, J.S., J.O. SMITH *Restoring a clipped signal* [Proceedings] ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing, 1991, pp. 1745-1748 vol.3, doi: 10.1109/ICASSP.1991.150655., [cit. 2022-04-05]. Dostupné z URL: <http://ieeexplore.ieee.org/document/150655/>.

[17] ÁVILA, Flávio R., M.P. TCHEOU, L. W.P. BISCAINHO. *Audio Soft Declipping Based on Constrained Weighted Least Squares* [online] in IEEE Signal Processing Letters, vol. 24, no. 9, pp. 1348-1352, Sept. 2017, doi: 10.1109/LSP.2017.2727964 [cit. 2022-04-06]. Dostupné z URL: <https://ieeexplore.ieee.org/document/7982663>.

[18] GORLOW S., J. REISS. *Model-Based Inversion of Dynamic Range Compression* [online] in IEEE Transactions on Audio, Speech and Language Processing, vol. 21, no. 7, pp. 1434-1444, July 2013, doi: 10.1109/TASL.2013.2253099. Dostupné z URL: <https://ieeexplore.ieee.org/document/6480792>.

[19] ZÁVIŠKA P., P. RAJMIC, A. OZEROV and L. RENCKER, *A Survey and an Extensive Evaluation of Popular Audio Declipping Methods,* [online] in IEEE Journal of Selected Topics in Signal Processing, vol. 15, no. 1, pp. 5-24, Jan. 2021, doi: 10.1109/JSTSP.2020.3042071. Dostupné z URL: <https://ieeexplore.ieee.org/document/9281027>.

[20] ADLER A., V. EMIYA, M. G. JAFARI, M. ELAD, R. GRIBONVAL and M. D. PLUMBLEY, *A constrained matching pursuit approach to audio declipping,* [online] 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011, pp. 329-332, doi: 10.1109/ICASSP.2011.5946407. [cit. 2022-04-12] Dostupné z URL: <https://ieeexplore.ieee.org/document/5946407>.

[21] DEFRAENE B., N. MANSOUR, S. De HERTOGH, T. van WATERSCHOOT, M. DIEHL and M. MOONEN, *Declipping of Audio Signals Using Perceptual Compressed Sensing,* in *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 12, pp. 2627-2637, Dec. 2013, doi: 10.1109/TASL.2013.2281570. [cit. 2022-04-12] Dostupné z URL: <https://ieeexplore.ieee.org/document/6600777>.

[22] SIEDENBURG K., M. KOWALSKI and M. DÖRFLER, *Audio declipping with social sparsity* in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 2014, pp. 1577-1581, doi: 10.1109/ICASSP.2014.6853863. [cit. 2022-04-12] Dostupné z URL: <https://ieeexplore.ieee.org/document/6600777>.

[23] ZÁVIŠKA P., P. RAJMIC, O. MOKRÝ, *Audio declipping performance enhancement via crossfading* in *Signal Processing*, Volume 192, 2022, 108365, ISSN 0165-1684, https://doi.org/10.1016/j.sigpro.2021.108365. [cit. 2022-04-12] Dostupné z URL: <https://www.sciencedirect.com/science/article/pii/S0165168421004023>.

[24] KITIĆ S., N. BERTIN, R. GRIBONVAL, *Sparsity and cosparsity for audio declipping: a flexible non-convex approach* 2015, arXiv, doi = 10.48550/ARXIV.1506.01830. Dostupné z URL: <https://arxiv.org/abs/1506.01830>.

[25] RENCKER L., BACH F., WANG W., PLUMBLEY M.D., *Consistent Dictionary Learning for Signal Declipping,* 2018, In: Deville, Y., Gannot, S., Mason, R., Plumbley, M., Ward, D. (eds) Latent Variable Analysis and Signal Separation. LVA/ICA 2018. Lecture Notes in Computer Science(), vol 10891. Springer,

Cham. Dostupné z URL: <https://doi.org/10.1007/978-3-319-93764-9_41>

[26] COMBETTES, L.P.; PESQUET, J-C. *Proximal splitting methods in signal processing,* Fixed-Point Algorithms for Inverse Problems in Science and Engineering, 2011: 28 s. Dostupné z URL: <http://arxiv.org/abs/0912.3522v4>.

[27] PRŮŠA, Zdeněk, Peter L. SØNDERGAARD, Nicki HOLIGHAUS, Christopher WIESMEYR a Peter BALASZ. *The Large Time-Frequency Analysis Toolbox 2.4.0. Download LTFAT* [online]. 2018 [cit. 2022-04-23]. Dostupné z URL: <http://ltfat.org/download.html>.

[28] De-Clip. *De-Clip RX Help* [online]. 2021, [cit. 2022-05-05]. Dostupné z URL: <https://s3.amazonaws.com/izotopedownloads/docs/rx8/en/de-clip/index.html#more-information>

[29] Logic Pro Effects. *Logic Pro User Guide* [online]. 2021, [cit. 2022-05-10]. Dostupné z URL: <https://help.apple.com/pdf/logicpro-effects//en_US/logic-pro-effects-user-guide.pdf>

[30] SCHOEFFLER, M. et al., *webMUSHRA — A Comprehensive Framework for Web-based Listening Tests.* (2018) Journal of Open Research Software. 6(1), p.8. Dostupné z URL: <https://github.com/audiolabs/webMUSHRA>

[31] BIGURE A., *multiple_boxplot.m*, 2022, MATLAB Central File Exchange. [cit. 2022-05-15]. Dostupné z URL: <https://www.mathworks.com/matlabcentral/fileexchange/47233-multiple_boxplot-m>

[32] PEMO-Q - Hörtech gGmbH [online] [cit. 2022-05-25] Dostupné z URL: <http://www.hoertech.de/en/f-e-products/pemo-q.html>

Symbols and abbreviations

DSP	digital signal processing
ADC	analog-digital converter
DR	Douglas - Rachford
MUSHRA	Multi Stimulus test with Hidden Reference and Anchor
THD	Total Harmonic Distortion
RMS	Root Mean Square
DAW	Digital Audio Workstation
SDR	Signal to Distortion Ratio
SNR	Signal to Noise Ratio
FFT	Fast Fourier Transform
DFT	Discrete Fourier Transform
LTFAT	The Large Time-Frequency Analysis Toolbox

A Content of the electronic attachment

Tested in MATLAB R2021a. LTFAT toolbox [27] and PEMO-Q [32] `audioqual.m` required.

- `/` ..... root
  - `audio` ..... tested audio samples
    - `1_male_speech.wav`
    - `2_female_speech.wav`
    - `3_guitar.wav`
    - `4_flute.wav`
    - `5_saxophone.wav`
    - `6_violin.wav`
    - `7_orchestra.wav`
    - `8_pop_band.wav`
    - `9_electro.wav`
  - `mfiles` ..... .m scripts
    - `distortion.m`
    - `inverse_overdrive.m`
    - `overdrive.m`
    - `projection.m`
    - `quantization.m`
    - `soft_declip.m` ..... main file
    - `tape_saturation.m`
    - `tube_simulation.m`