



# 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

## NÁVRH SIGNÁLŮ PRO MODELOVÁNÍ NELINEÁRNÍCH SYSTÉMŮ

DESIGN OF SIGNALS FOR NONLINEAR SYSTEM MODELING

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Michael Kuba

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Štěpán Miklánek

BRNO 2024

# Diplomová práce

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

**Student:** Bc. Michael Kuba

**ID:** 186438

**Ročník:** 2

**Akademický rok:** 2023/24

## NÁZEV TÉMATU:

### Návrh signálů pro modelování nelineárních systémů

#### POKYNY PRO VYPRACOVÁNÍ:

Nastudujte metody modelování nelineárních systémů pomocí hlubokého učení a navrhnete sadu umělých trénovacích signálů, které budou sloužit k vytvoření modelů zkreslovacích kytarových efektů nebo elektronkových kytarových zesilovačů. Popište dosavadní poznatky v oblasti identifikace a modelování nelineárních systémů s hlavním zaměřením na současné metody, které využívají neuronové sítě. Navrhnete vhodnou kombinaci umělých trénovacích signálů, které zpracujete pomocí libovolných kytarových efektů nebo zesilovačů. Dále sestavíte modely neuronových sítí a použijete vytvořené sady vstupních a výstupních signálů k natrénování těchto modelů. Zhodnoťte, zda jsou navržené umělé signály vhodné k natrénování modelů kytarových efektů a proveďte porovnání s modely neuronových sítí, které byly natrénovány za pomoci zvukové sady složené ze záznamů hry na elektrickou kytaru. Na závěr proveďte objektivní a subjektivní zhodnocení všech natrénovaných modelů.

#### DOPORUČENÁ LITERATURA:

- [1] WRIGHT, Alec, Eero-Pekka DAMSKÄGG, Lauri JUVELA a Vesa VÄLIMÄKI. Real-Time Guitar Amplifier Emulation with Deep Learning. Applied Sciences [online]. 2020, 10(3) [cit. 2020-09-14]. ISSN 2076-3417. Dostupné z: doi:10.3390/app10030766
- [2] NOVÁK, Antonín, Laurent SIMON, František KADLEC a Pierrick LOTTON. Nonlinear System Identification Using Exponential Swept-Sine Signal. IEEE Transactions on Instrumentation and Measurement [online]. 2010, 59(8), 2220-2229 [cit. 2022-05-31]. ISSN 0018-9456. Dostupné z: doi:10.1109/TIM.2009.2031836

**Termín zadání:** 5.2.2024

**Termín odevzdání:** 21.5.2024

**Vedoucí práce:** Ing. Štěpán Miklánek

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

#### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Diplomová práce se zabývá trénovacími signály pro modelování nelineárních systémů pomocí hlubokého učení. Je zde popsán teoretický úvod k této problematice včetně úvodního popisu signálů a nelineárních zvukových zkreslovacích efektů. Cílem práce je navržení sady umělých trénovacích signálů, které mají sloužit k vytvoření modelů zkreslovacích kytarových efektů nebo elektronkových kytarových zesilovačů. Navržená sada umělých trénovacích signálů je poté zpracována kytarovými a baskytarovými efekty a pomocí rekurentní neuronové sítě jsou natrénovány jejich modely. Kvalita výsledných modelů je následně srovnána s kvalitou modelů vytvořených pomocí referenčního trénovacího signálu, složeného ze záznamů hry na elektrickou kytaru, a signálů z komerčně dostupných zařízení. Srovnání je provedeno dle objektivních metrik a subjektivního hodnocení pomocí poslechového testu MUSHRA.

## **KLÍČOVÁ SLOVA**

strojové učení, neuronová síť, hluboké učení, zkreslovací efekt, nelineární systém, trénovací signál, modelování, mushra

## **ABSTRACT**

This Master's thesis is focused on training signals for nonlinear system modeling using deep learning. A theoretical introduction to the problem is described, including an initial description of signals and nonlinear distortion audio effects. The goal of the thesis is to design a set of artificial training signals for creating models of distortion guitar effects or tube guitar amplifiers. The designed set of artificial training signals is then processed, utilizing guitar and bass guitar effects, and using a recurrent neural network their models are trained. The quality of the resulting models is afterwards compared with the quality of the models trained with the help of a reference training signal, composed of electric guitar recordings, and signals from commercially available devices. The comparison is carried out in accordance to objective metrics and subjective evaluation by the MUSHRA listening test.

## **KEYWORDS**

machine learning, neural network, deep learning, distortion effect, nonlinear system, training signal, modeling, mushra

KUBA, Michael. *Návrh signálů pro modelování nelineárních systémů*. Diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2024. Vedoucí práce: Ing. Štěpán Miklánek

## Prohlášení autora o původnosti díla

<b>Jméno a příjmení autora:</b>	Bc. Michael Kuba
<b>VUT ID autora:</b>	186438
<b>Typ práce:</b>	Diplomová práce
<b>Akademický rok:</b>	2023/24
<b>Téma závěrečné práce:</b>	Návrh signálů pro modelování nelineárních systémů

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora\*

---

\*Autor podepisuje pouze v tištěné verzi.

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Štěpánu Miklánkovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci, panu Ing. Tomáši Suchánkovi za podporu, trpělivost a cenné rady a v neposlední řadě své rodině a přátelům za shovívavost a podporu.

# Obsah

<b>Úvod</b>	<b>12</b>
<b>1 Signály a systémy</b>	<b>14</b>
1.1 Reprezentace signálů . . . . .	14
<b>2 Nelineární zvukové efekty</b>	<b>17</b>
2.1 Overdrive . . . . .	18
2.2 Distortion . . . . .	19
2.3 Fuzz . . . . .	20
2.4 Elektronkový kytarový zesilovač . . . . .	20
<b>3 Metody pro číslicovou simulaci nelineárních efektů</b>	<b>23</b>
3.1 Vnější popis systému . . . . .	23
3.2 Vnitřní popis systému . . . . .	23
3.3 Greybox . . . . .	25
3.4 Návaznost stavového popisu na rekurentní neuronové sítě . . . . .	25
<b>4 Umělé neuronové sítě</b>	<b>27</b>
4.1 Umělý neuron . . . . .	27
4.2 Aktivační funkce . . . . .	28
4.3 Dopředné sítě . . . . .	30
4.4 Konvoluční sítě . . . . .	31
4.5 Rekurentní sítě . . . . .	32
4.6 Optimalizace sítě . . . . .	36
4.7 Ztrátové funkce . . . . .	37
<b>5 Identifikace nelineárních systémů</b>	<b>39</b>
5.1 Trénovací signály v komerčních zařízeních . . . . .	40
5.2 Návrh trénovacích signálů . . . . .	46
<b>6 Výsledky</b>	<b>52</b>
6.1 Objektivní hodnocení výsledků . . . . .	53
6.2 Subjektivní hodnocení výsledků . . . . .	62
6.3 Shrnutí výsledků . . . . .	64
<b>Závěr</b>	<b>66</b>
<b>Literatura</b>	<b>68</b>

Seznam zkratek	75
Seznam symbolů a veličin	77
A Obsah elektronické přílohy	78

# Seznam obrázků

1.1	Závislost signálu na čase . . . . .	15
1.2	Příklad spektrogramu . . . . .	16
2.1	Číslicová realizace zkreslovacích efektů . . . . .	17
2.2	Překrývání spekter vlivem nevhodného vzorkovacího kmitočtu . . . . .	17
2.3	Pedálové efekty a elektronkový zesilovač . . . . .	18
2.4	Časové průběhy a kmitočtové spektrum pro efekt Overdrive při $f = 300$ Hz . . . . .	19
2.5	Časové průběhy a kmitočtové spektrum pro efekt Distortion při $f = 300$ Hz . . . . .	19
2.6	Časové průběhy a kmitočtové spektrum pro efekt Fuzz při $f = 300$ Hz . . . . .	20
2.7	Blokové schéma elektronkového kytarového zesilovače . . . . .	21
2.8	Časové průběhy a kmitočtové spektrum pro elektronkový zesilovač při $f = 300$ Hz . . . . .	21
2.9	Příklad intermodulačního zkreslení . . . . .	22
3.1	Blokový diagram stavového popisu LTI diskrétního systému . . . . .	24
3.2	Blokové schéma navrhované sítě během trénování . . . . .	26
4.1	Obecné schéma neuronové sítě . . . . .	27
4.2	Aktivační funkce . . . . .	28
4.3	Příklad MLP . . . . .	30
4.4	Základní schéma konvoluční sítě . . . . .	31
4.5	Model rekurentní neuronové sítě . . . . .	32
4.6	Elmanova síť . . . . .	33
4.7	LSTM buňka . . . . .	34
4.8	GRU buňka . . . . .	35
5.1	Zobecněný polynomický Hammersteinův model . . . . .	40
5.2	Neural DSP Quad Cortex . . . . .	40
5.3	Schéma zapojení pro modelovací proceduru . . . . .	41
5.4	Závislost trénovací sady signálů na čase – Quad Cortex . . . . .	42
5.5	Kmitočtové spektrum trénovacích signálů – Quad Cortex . . . . .	42
5.6	Kemper Profiler Stage . . . . .	43
5.7	Wiener-Hammersteinův model . . . . .	43
5.8	Závislost testovací sady signálů na čase – Kemper . . . . .	44
5.9	Kmitočtové spektrum trénovacích signálů – Kemper . . . . .	44
5.10	Závislost trénovací sady signálů na čase – NAM . . . . .	45
5.11	Kmitočtové spektrum sady signálů – NAM . . . . .	46
5.12	Závislost navržené sady signálů na čase . . . . .	50
5.13	Kmitočtové spektrum navržené sady signálů . . . . .	51

6.1	Schéma zapojení při nahrávání . . . . .	53
6.2	Efekty Big Muff a Big Stuff . . . . .	54
6.3	Časový průběh výstupních signálů a absolutní chyby pro Big Stuff . .	54
6.4	Diferenční spektrogram pro Big Stuff . . . . .	55
6.5	Orange TT15 a Cypress TT-15 . . . . .	55
6.6	Časový průběh výstupních signálů a absolutní chyby pro Cypress TT-15	56
6.7	Diferenční spektrogram pro Cypress TT-15 . . . . .	56
6.8	Efekt Boss ODB-3 . . . . .	57
6.9	Časový průběh výstupních signálů a absolutní chyby pro Boss ODB-3	58
6.10	Diferenční spektrogram pro Boss ODB-3 . . . . .	58
6.11	Efekt Darkglass Vintage Ultra v2 . . . . .	59
6.12	Časový průběh výstupních signálů a absolutní chyby pro Darkglass Vintage Ultra v2 . . . . .	60
6.13	Diferenční spektrogram pro Darkglass Vintage Ultra v2 . . . . .	60
6.14	Efekt Marshall JCM900 4100 . . . . .	61
6.15	Časový průběh výstupních signálů a absolutní chyby pro Marshall JCM900 4100 . . . . .	61
6.16	Diferenční spektrogram pro Big Stuff . . . . .	62
6.17	Náhled na prostředí webMUSHRA . . . . .	63
6.18	Výsledky poslechového testu MUSHRA . . . . .	64

## Seznam tabulek

6.1	Hodnoty ztrátových funkcí pro Big Stuff . . . . .	54
6.2	Hodnoty ztrátových funkcí pro Cypress TT-15 . . . . .	56
6.3	Hodnoty ztrátových funkcí pro Boss ODB-3 . . . . .	57
6.4	Hodnoty ztrátových funkcí pro Darkglass Vintage Ultra v2 . . . . .	59
6.5	Hodnoty ztrátových funkcí pro Marshall JCM900 4100 . . . . .	61
6.6	Srovnání výsledných hodnot pro GTR a T9 . . . . .	65

# Úvod

V oblasti zvukového inženýrství se modelování nelineárních zkreslovacích efektů stává stále populárnějším, stejně jako využívání nástrojů pro toto modelování, a to jak v softwarové, tak v hardwarové formě. Kytarové číslicové multiefekty jsou u muzikantů v dnešní době stále vyhledávanější, stejně jako simulace zkreslovacích efektů nebo zesilovačů v podobě zásuvných modulů (*plug-in*). Jedním z velmi silných nástrojů pro realizaci takového modelování jsou neuronové sítě, jejichž využití vykazuje v této oblasti velmi dobré výsledky.

Tato práce je vytvořena za účelem navržení umělého trénovacího signálu pro neuronovou síť, kterým by byla nahrazena trénovací sada složená ze záznamů hry na elektrickou kytaru a baskytaru. Konkrétně se jedná o rekurentní neuronovou síť, která slouží k vytvoření modelů nelineárních zkreslovacích zvukových efektů. První část práce je zaměřena na teoretický úvod, který zahrnuje také popis navrženého trénovacího signálu. Druhá část se potom věnuje popisu modelovaných efektů, průběhu učení neuronové sítě a objektivních a subjektivních výsledků.

Úvod práce je věnován popisu signálů a systémů, jejich forem a možnostem zobrazování jejich časových a kmitočtových závislostí. Druhá kapitola je věnována nelineárním zvukovým efektům. Kapitola je zaměřena především na efekty zkreslovací, z nichž je zde popsán efekt *overdrive*, *distortion*, *fuzz* a zkreslení pomocí elektronického zesilovače. Ve třetí kapitole je pojednáváno o metodách pro číslicovou simulaci nelineárních efektů. Je zde uveden teoretický úvod k vnějšímu a vnitřnímu popisu systému a jejich kombinaci. Na konci kapitoly je uveden popis návaznosti stavového popisu na rekurentní neuronové síť. Další kapitola je věnována umělým neuronovým sítím. Nejprve je zde popsán historický vývoj umělých neuronových sítí a poté umělý neuron a aktivační funkce. Je zde uveden popis různých typů sítí a to konkrétně sítí dopředných, konvolučních, temporálních konvolučních a rekurentních, u kterých jsou blíže popsány také Elmanova rekurentní síť a výpočetní buňky GRU a LSTM. Kapitola pokračuje popisem optimalizace sítě včetně popisu algoritmu zpětného šíření chyby a optimalizačních algoritmů jako gradientní sestup, RMSProp a Adam. Na závěr je zde uveden popis ztrátových funkcí MSE, MAE, ESR a STFT.

Pátá kapitola pojednává o signálech pro identifikaci nelineárních systémů. Jsou zde popsána komerčně využívaná zařízení pro modelování zvukových efektů a signály, které pro toto modelování využívají. Jedná se o zařízení Quad Cortex od firmy Neural DSP a Profiler Stage od firmy Kemper. Dále je zde uvedena knihovna jazyka Python s názvem Neural Amp Modeler jakožto volně dostupná verze nástroje pro modelování efektů pomocí neuronových sítí. V druhé části kapitoly již začíná popis praktické části této práce. Jsou zde popsány nástroje, které byly využity pro praktickou část a poté je zde detailně popsána sada trénovacích signálů, jejíž vytvoření

bylo cílem této práce.

Poslední kapitola práce je zaměřena na popis modelovaných efektů a srovnání získaných výsledků modelů neuronové sítě trénované pomocí sad signálů z dříve zmíněných zařízení a knihovny a sady signálů, která byla vytvořena v rámci této práce. Tyto výsledky jsou srovnávány s výsledky modelu trénovaného pomocí referenční sady nahrávek hry na elektrickou kytaru a baskytaru. Výsledky jsou rozděleny na objektivní, kde jsou hodnoceny dle výsledných hodnot ztrátových funkcí, časových průběhů výstupních signálů a jejich kmitočtových spekter, a na subjektivní, kde byl vyhodnocen poslechový test MUSHRA.

# 1 Signály a systémy

Signál může být definován jako hmotný nositel informace. Jako signál může být považována některá z fyzikálních veličin jako např. elektrický proud nebo napětí. Pro přenos informace bývají využity např. signály elektrické nebo optické [1]. Rozdělení signálů může být provedeno mnoha způsoby. V kontextu této práce jsou zde popsány signály spojité v čase, diskrétní v čase a signály číslicové.

- Signály spojité v čase jsou takové signály, u kterých je definičním oborem spojitá oblast. Jsou značeny např.  $s(t)$ , kde  $t$  značí čas.
- Definičním oborem signálů s diskrétním časem je potom konečná množina bodů a jsou značeny např.  $s[n]$ , kde  $n$  značí vzorek, který je získán navzorkováním spojitého signálu, kdy jsou ze spojitého signálu odebírány vzorky v pravidelných časových okamžicích, které jsou od sebe vzdáleny o vzorkovací periodu  $T_{vz}$ . Podrobnější popis vzorkování je uveden v [2]. Hodnoty signálu jsou tedy známé pouze v určitých časových okamžicích.
- Číslicové signály jsou potom např. binární reprezentací signálů diskrétních a jsou získány převedením spojitého signálu na číslicový. Tato procedura se nazývá analogově číslicový (A/D) převod a je podrobněji popsána v [2]. Důležitý pojem u A/D převodu signálů je kvantování, které spočívá v rozdělení signálu na konečný počet úrovní a zaokrouhlení každé hodnoty na nejbližší úroveň kvantovacího kroku, kterým je určeno rozlišení číslicového signálu a přesnost A/D převodu. Platí tedy, že čím je kvantovací krok menší, tím věrněji je číslicový signál podobný převáděnému spojitému signálu. Při procesu A/D převodu může vznikat kvantovací šum, který může být také omezen velikostí kvantovacího kroku. Kvantovací krok je vyjádřen vzorcem

$$q = \frac{U_{\max}}{2^n}, \quad (1.1)$$

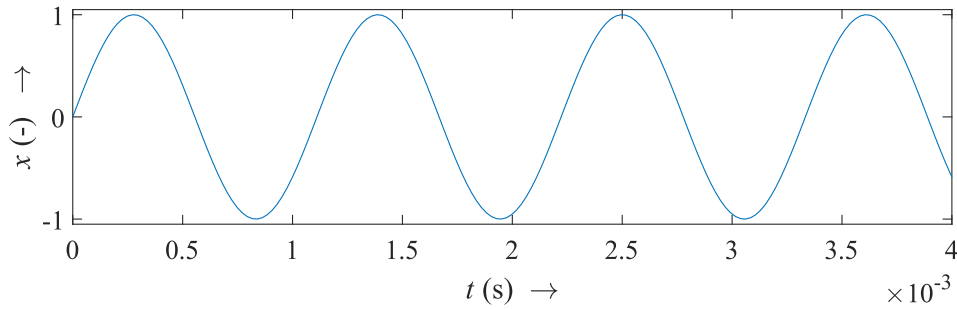
kde  $U_{\max}$  značí maximální hodnotu napětí spojitého signálu, která je ale omezena maximálním napětím, které dokáže zpracovat analogově číslicový převodník a  $n$  značí počet kvantovacích hladin.

Dále je potřeba definovat diskrétní a číslicový systém. Diskrétním systémem je zpracováván signál s diskrétním časem. Provádí tedy transformaci vstupního diskrétního signálu  $x[n]$  na výstupní diskrétní signál  $y[n]$  [1]. Číslicový systém je potom typ diskrétního systému, kterým jsou zpracovávány číslicové signály.

## 1.1 Reprezentace signálů

V této práci je využito několik způsobů reprezentace signálů. První z nich je zobrazení signálů v časové oblasti, tedy jeho závislost na čase. Příklad takového zobrazení

znázorňuje Obr. 1.1.



Obr. 1.1: Závislost signálu na čase.

Další použitá reprezentace je zobrazení kmitočtové oblasti signálů, tedy jejich kmitočtové spektrum, což je závislost amplitud nebo modulů na kmitočtu. Analýza spektra se provádí pomocí Fourierovy transformace a jejích variant. Pomocí ní je kmitočtové spektrum rozloženo na jednotlivé harmonické složky [1]. Pro diskrétní posloupnost vzorků je definována vztahem [3]

$$X[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-\frac{2\pi jkn}{N}}, \quad k = 0, 1, \dots, N-1. \quad (1.2)$$

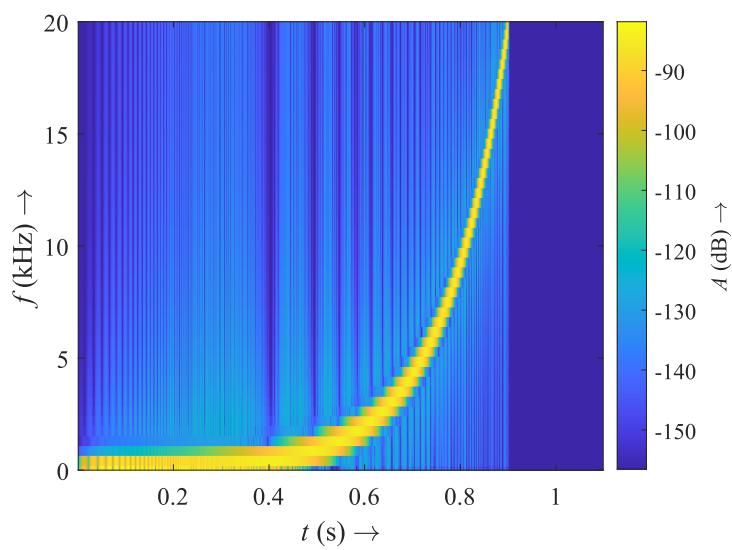
Transformace je aplikována po krátkých usecích, které jsou vyhraničeny analyzačním oknem  $w[n]$ . Okno představuje posloupnost vzorků, jimiž je analyzovaný úsek násoben [3].

Nejjednodušší zavedení časově-kmitočtového popisu signálu je krátkodobá Fourierova transformace (STFT). Pro tento typ transformace platí vztah [3, 4]

$$X_{\text{STFT}}[m, k] = \frac{1}{N} \sum_{n=0}^{N-1} x[hm + n] w[n] e^{-\frac{2\pi jkn}{N}}, \quad k = 0, 1, \dots, N-1, \quad (1.3)$$

kde  $N$  značí délku okna,  $m$  pořadové číslo segmentu a  $h$  značí jeho posun. Transformace je počítána pro  $N$  vzorků, proto je výsledkem také  $N$  složek spektra pro každý transformovaný segment. Parametr  $N$  tedy představuje rozlišení výsledného zobrazení, které se nazývá spektrogram. Zpravidla jde o zobrazení absolutní hodnoty jednostranného reálného spektra v logaritmickém měřítku [3]. U STFT jsou také využity váhovací okna, které mají mnoho různých typů jako např. Obdélníkové, Hammingovo, Hannovo nebo Blackmanovo. Každý z typů má jiné vlastnosti a je tedy vhodný pro různé aplikace. Okna jsou použita pro redukci nežádoucích efektů, které mohou být způsobeny konečnou délkou okna a diskretizací signálu. Bez použití váhovacího okna se předpokládá, že je signál periodický, což může vést k chybám v analýze signálu. Volbou správného váhovacího okna může být tedy ovlivněno rozlišení v kmitočtové a časové doméně signálu. Na Obr. 1.2 je zobrazen

příklad spektrogramu přeladovaného sinusového signálu při vzorkovacím kmitočtu  $f_{\text{vz}} = 44,1 \text{ kHz}$ , délce okna Hammingova typu  $N = 512$  a posunu  $h = 256$ .



Obr. 1.2: Příklad spektrogramu.

## 2 Nelineární zvukové efekty

Kategorii nelineárních zvukových efektů lze rozdělit na efekty ovlivňující dynamiku signálu a efekty zkreslovací. Mezi efekty pro úpravu dynamiky patří např. kompresor, limiter nebo expander. Jako zkreslovací efekty jsou označovány např. efekty imitující přebuzení zesilovače, elektronkové zesilovače či jejich simulátory [5, 6, 7]. Obecně lze tyto efekty popsat vztahem

$$y[n] = f(x[n]), \quad (2.1)$$

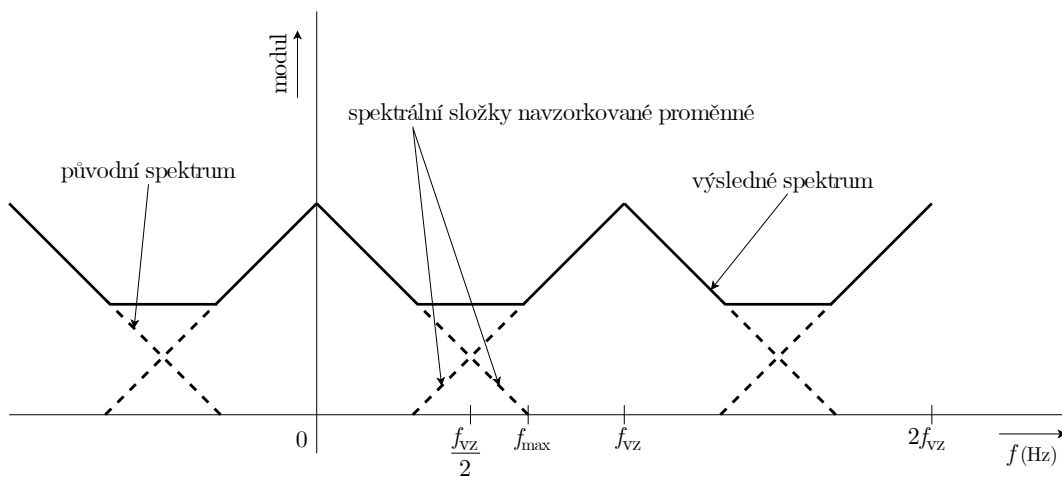
kde pravá strana rovnice značí libovolnou nelineární funkci vstupního signálu  $x[n]$  [6, 8]. Proměnnou  $n$  je v tomto případě značen vzorek signálu, který je získán navzorkováním spojitého signálu, tedy jeho převedením na signál číslicový.

Na obrázku 2.1 je zobrazeno blokové schéma číslicové realizace efektů níže zmíněných typů. Jednotlivé efekty se liší typem nelineární tvarovací funkce, která musí být pro různé efekty vhodně zvolena [13].



Obr. 2.1: Číslicová realizace zkreslovacích efektů [13].

V důsledku nelinearit jsou pomocí těchto efektů vytvářeny nové vyšší harmonické složky v kmitočtovém spektru výstupního signálu, což může vést ke vzniku aliasingu z důvodu rozšíření šířky pásma signálu.



Obr. 2.2: Překrývání spekter vlivem nevhodného vzorkovacího kmitočtu [2].

Aliasing může být také způsoben nevhodnou volbou vzorkovacího kmitočtu. Při převodu z číslicové formy zpět na spojitý signál nebo také při číslicovém zpracování signálů systémem, který generuje harmonické složky, které jsou větší než  $f_{vz}/2$ , může docházet k překrývání sousedních spekter signálu (Obr. 2.2). Tomu lze zamezit vhodnou volbou vzorkovacího kmitočtu v souladu s vzorkovací poučkou

$$f_{vz} = 2f_{max}. \quad (2.2)$$

Rovnice značí, že vzorkovací kmitočet  $f_{vz}$  musí být větší než dvojnásobek maximálního kmitočtu signálu  $f_{max}$ , který je analyzován. Tato poučka bývá také označována jako Shannonův-Kotelnikův nebo Nyquistův teorém [2]. Aliasing je také možné potlačit použitím např. antialiasingového filtru [5, 6, 8]. Vzniku aliasingu při číslicovém zpracování signálu lze také zamezit pomocí převzorkování signálu, kdy je signál nejprve nadvzorkován a poté podvzorkován. Proces převzorkování je popsán v [9].

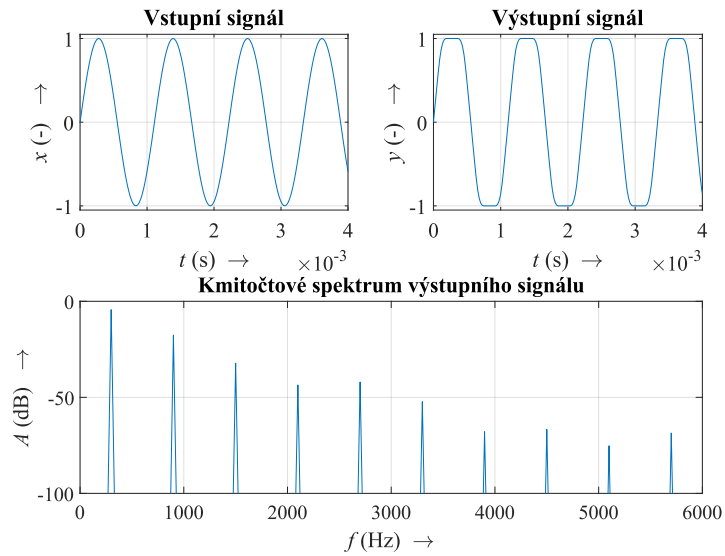
Zkreslovací efekty jsou nedílnou součástí rockové hudby, kde jsou použity pro typické zkreslené kytarové zvuky. Často bývají použity nohou ovládané pedálové procesory jako jsou overdrive, distortion a fuzz nebo elektronkové zesilovače (Obr. 2.3).



Obr. 2.3: Pedálové efekty a elektronkový zesilovač [48].

## 2.1 Overdrive

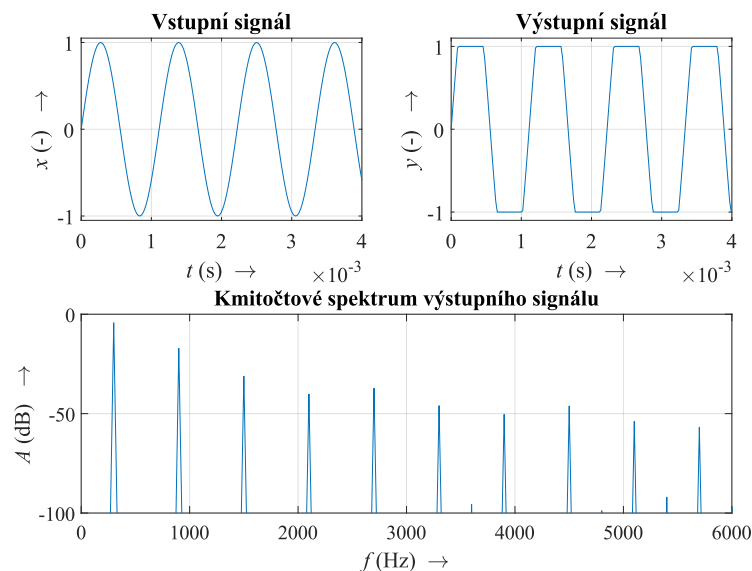
U efektu typu overdrive se jedná o tzv. soft clipping neboli měkký ořez signálu. Efektem je simulováno přebuzení elektronkového nebo tranzistorového zesilovače. Z důvodu omezení vstupního rozkmitu signálu napájecím napětím zesilovače je způsobena limitace výstupního signálu [5, 6, 11].



Obr. 2.4: Časové průběhy a kmitočtové spektrum pro efekt Overdrive při  $f = 300$  Hz [8].

## 2.2 Distortion

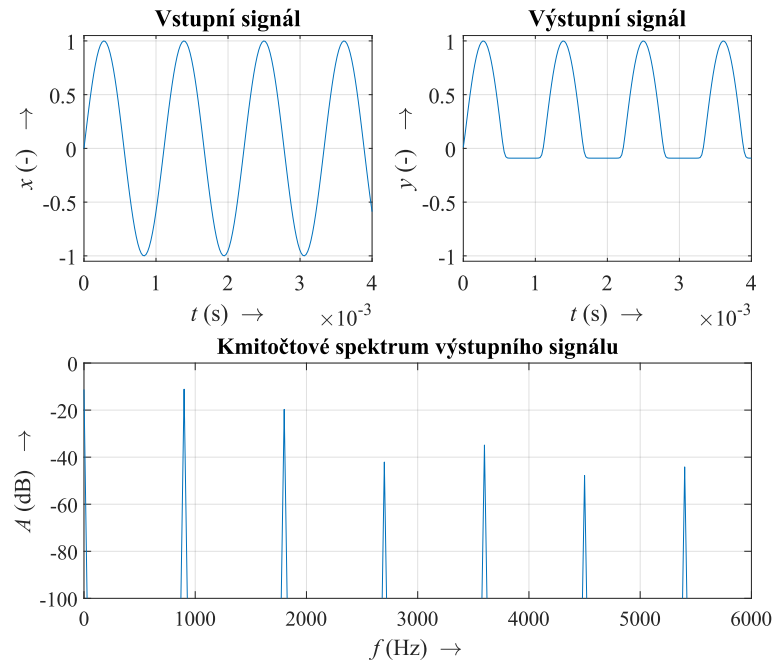
Zde se jedná o tzv. hard clipping nebo také tvrdý ořez signálu. Efekt je tvořen silným zesílením vstupního signálu následovaným jeho limitací. Pomocí tohoto efektu je kmitočtové spektrum výstupního signálu výrazně obohaceno o vyšší harmonické složky [6, 11].



Obr. 2.5: Časové průběhy a kmitočtové spektrum pro efekt Distortion při  $f = 300$  Hz [8].

## 2.3 Fuzz

Tento efekt je charakteristický asymetrickým zkreslením výstupního signálu, kdy je úroveň záporného ořezu nižší než hodnota kladného ořezu nebo obráceně. V efektu bývají využity dva a více tranzistorů k zesílení signálu až do bodu silného zkreslení. Např. u efektu Fuzz Face [10], který využíval legendární kytarista Jimi Hendrix, je nejdříve dosaženo limitace kladných částí signálu. Limitace záporných částí signálu je dosaženo až při dalším zesílení vstupního signálu. Pomocí tohoto efektu jsou zesíleny jak sudé, tak liché harmonické složky. Výstupní signál obsahuje tedy bohaté kmitočtové spektrum [5, 11].



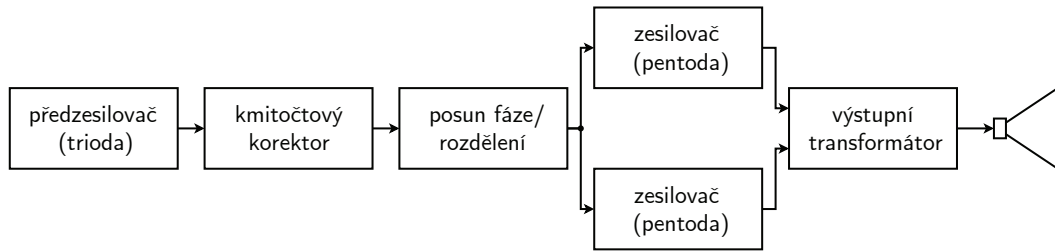
Obr. 2.6: Časové průběhy a kmitočtové spektrum pro efekt Fuzz při  $f = 300$  Hz [8].

## 2.4 Elektronkový kytarový zesilovač

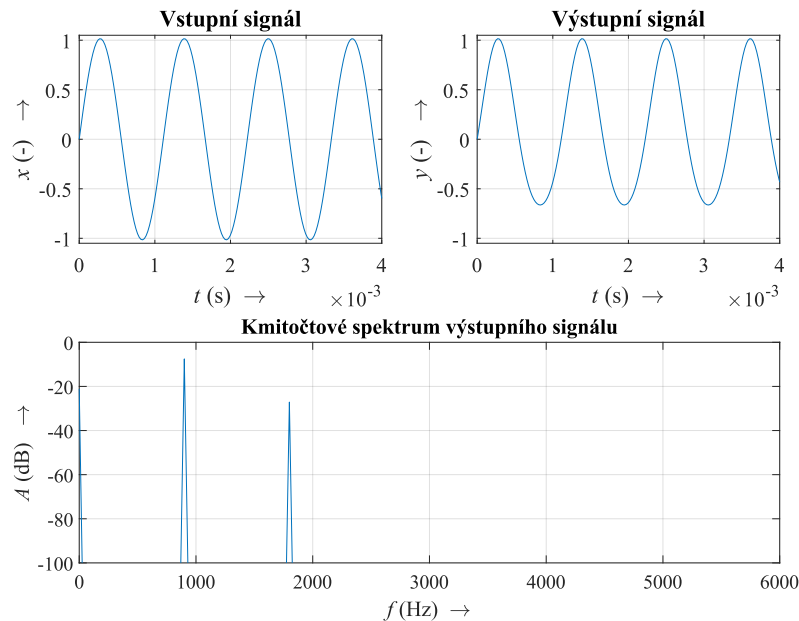
Při dostatečném zesílení vstupního signálu může být dosaženo maximálního výstupního výkonu elektronky a tím k jejímu nasycení, což vede ke zkreslení vstupního signálu. Vzhledem k nelineární povaze elektronky je toto zkreslení také nelineární. Na Obr. 2.7 je zobrazeno blokové schéma elektronkového kytarového zesilovače. Tyto zesilovače mohou být simulovány pomocí vhodné nelineární funkce jako např.

$$y[n] = ax + bx^2, \quad (2.3)$$

kde  $a = 2$ ,  $b = 1$  a  $x < 0$ . Touto křivkou je přibližně aproximována volt-ampérová charakteristika zesilovacího stupně. Ve výstupním signálu je nižší výskyt vyšších harmonických složek než u předešlých efektů a převažují sudé násobky první harmonické nad ostatními. V kytarovém aparátu je však dosaženo vyšší saturace (viz [12]) a kmitočtové spektrum bude plnější [6, 8, 11]. Tento typ zesilovače bývá často využíván právě pro jeho typické a jedinečné tónové charakteristiky a jeho charakteristický bohatý zvuk.



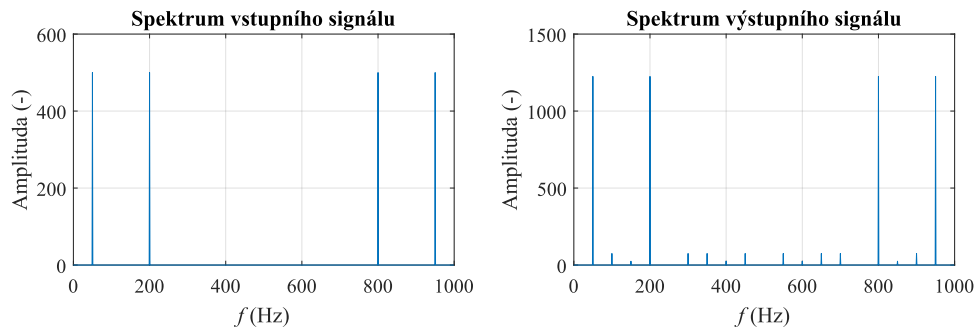
Obr. 2.7: Blokové schéma elektronkového kytarového zesilovače – převzato z [8].



Obr. 2.8: Časové průběhy a kmitočtové spektrum pro elektronkový zesilovač při  $f = 300$  Hz.

Je zde také důležité zmínit intermodulační zkreslení (Obr. 2.9), což je důležitý parametr pro charakteristiku kvality zesilovače. Toto zkreslení může v elektronkovém zesilovači a v dalších nelineárních zařízeních vznikat při zpracování dvou nebo více signálů o různých kmitočtech. Příčinou tohoto zkreslení může být např. nelinearita elektronek nebo jiných komponent obvodu. Výsledkem je potom vznik nových

kmitočtů, které nejsou přítomny v původních signálech a je jimi tedy ovlivněna kvalita zvuku. Tyto nové kmitočty se objevují na součtu a rozdílu původních kmitočtů a nejsou s původními kmitočty harmonické. Pokud jsou tedy přítomny kmitočty  $f_1$  a  $f_2$ , mohou být v důledku intermodulačního zkreslení vytvořeny kmitočty  $nf_1 \pm mf_2$ , kde  $n$  a  $m$  jsou celá čísla [14].



Obr. 2.9: Příklad intermodulačního zkreslení.

## 3 Metody pro číslicovou simulaci nelineárních efektů

Na diskretní systém může být obecně pohlíženo jako na abstraktní model, kdy je znám pouze jeho vstupní a výstupní signál. Tento popis je nazýván jako vnější. Další možností je popis systému pomocí diferenciálních rovnic, kdy je popisován vztah mezi vstupními, stavovými a výstupními veličinami. Tento popis je nazýván jako vnitřní. V některých případech může být použita kombinace obou těchto způsobů popisu diskretních systémů [15, 16, 17].

### 3.1 Vnější popis systému

Systém je v tomto případě popsán pomocí reakce jeho výstupu na vstupní podněty. Analýzou vstupních a výstupních dat je potom možné získat matematický model daného systému. Modelování systémů pomocí tohoto popisu bývá označováno jako *blackbox*. Spočívá ve schopnosti napodobit chování popisovaného systému [15, 18, 27].

Mezi často používané formy modelování jsou zařazeny např. Volterrovy řady, dynamická konvoluce nebo Wiener-Hammersteinovy modely [26]. V současné době však nejvíce převládá použití neuronových sítí, které jsou podrobněji popsány v následující kapitole.

### 3.2 Vnitřní popis systému

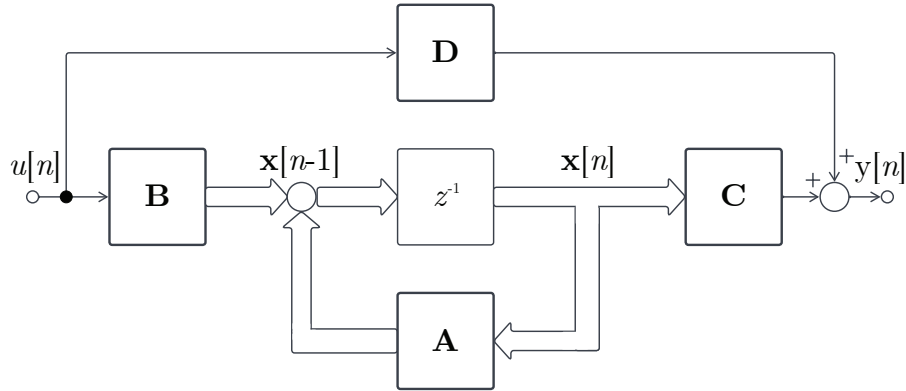
Modelování systémů pomocí vnitřního nebo také stavového popisu bývá označováno jako *whitebox*. Tento způsob modelování je založen na skutečnosti, že je známa kompletní studie vnitřního obvodu. Potom může být toto modelování velmi přesné. K získání výstupu systému je však potřeba velkého množství výpočetních operací, a proto nemusí být tento popis vhodný pro modelování s minimálními výpočetními nároky. Často bývá využíván pro modelování dynamických procesů, jako jsou např. kompresory [8, 15, 18].

Popis systému je proveden pomocí stavové rovnice dynamiky a stavové rovnice výstupu systému. Pro lineární časově invariantní MIMO systém je maticový zápis stavového popisu ve tvaru

$$\begin{aligned}\mathbf{x}(n+1) &= \mathbf{A}\mathbf{x}(n) + \mathbf{B}\mathbf{u}(n) \\ \mathbf{y}(n) &= \mathbf{C}\mathbf{x}(n) + \mathbf{D}\mathbf{u}(n),\end{aligned}\tag{3.1}$$

kde  $\mathbf{A}$  je matice systému, pomocí které jsou popsány vztahy mezi stavy systému a jejich změnami,  $\mathbf{B}$  je matice buzení, pomocí které jsou popsány vztahy mezi vstupy do

systému a změnami stavů,  $\mathbf{C}$  je výstupní matice, pomocí které jsou popsány vztahy mezi stavy systému a výstupy systému a  $\mathbf{D}$  je matice převodu, pomocí které jsou popsány vztahy mezi vstupy a výstupy systému. Matice  $\mathbf{D}$  bývá v mnoha případech nulová. Narozdíl od ostatních způsobů popisů systému je zde vstupní signál značen  $\mathbf{u}$  a  $\mathbf{x}$  zde značí vektor stavů systému. Tyto rovnice jsou znázorněny na obrázku 3.1 jako blokový diagram [8, 19].



Obr. 3.1: Blokový diagram stavového popisu LTI diskrétního systému [19].

Tento přístup se zaměřuje na odvození základních diferenciálních rovnic vnitřních procesů systémů a jejich následnou diskretizaci za účelem vytvoření numerického řešení. Obecně jsou preferovány automatizované nástroje pro všeobecné použití než provádění celého procesu pomocí ručních výpočtů. Příklady takových systémů jsou např. vlnové číslicové filtry WDF (Wave Digital Filter) [25] nebo Port-Hamiltonovy systémy [26]. Další metodou je např. DK-metoda [21], což je odvozená diskrétní varianta K-metody [20], která využívá Kirchhoffových zákonů pro řešení stavové reprezentace nelineárních obvodů ve spojitém čase. DK-metoda využívá stavový popis, pomocí kterého může být popsán jakýkoliv systém, který lze charakterizovat systémem diferenciálních rovnic. Jedná se o metodu, kterou je nejprve prováděna diskretizace stavových prvků, což jsou prvky popisující vnitřní parametry systému jako např. kondenzátory nebo cívky. Po diskretizaci stavových prvků je řešen výsledný systém pro stavový popis systému. Nejprve je sestaven maticový systém rovnic, pomocí kterého jsou popsány koeficienty přechodů mezi stavy a nelineární vztahy mezi proměnnými systému [20, 21]. Nelinearita obvodu je potom řešena pomocí numerických algoritmů jako např. Newtonova metoda. Z důvodu proměnlivé výpočetní složitosti je vhodné využití výpočetního výkonu číslicového signálového procesoru DSP. Pro aplikace v reálném čase je také vhodné využití aproximací funkcí popisujících nelinearitu systému za pomocí např. vyhledávacích tabulek. Celý proces řešení nelinearity probíhá iterativně až do dosažení dostatečně přesného výsledku stanoveného předem definovanou hodnotou chyby [22].

### 3.3 Greybox

Přístupy k modelování systému, které jsou na pomezí dvou předešlých jsou označovány jako *greybox*. Zde jsou využívány jak znalosti obvodu, tak reakce výstupu systému na vstupní data [7, 15, 18]. Jednou z hlavních nevýhod této techniky je, že musí být zvláště vytvořen model pro každou možnou kombinaci ovládacích prvků a poté jsou prováděny interpolace mezi parametry modelu podle aktuálního uživatelského nastavení [24].

V článku [24] je např. popsán proces modelování, kdy jediným předpokladem o referenčním systému je jeho základní struktura. Proces modelování je zcela automatický a využívá pouze vstupně-výstupní měření a iterativní optimalizaci pro přizpůsobení číslicového modelu referenčnímu zařízení. Nejsou zde vyžadovány žádné znalosti o obvodu.

### 3.4 Návaznost stavového popisu na rekurentní neuronové síť

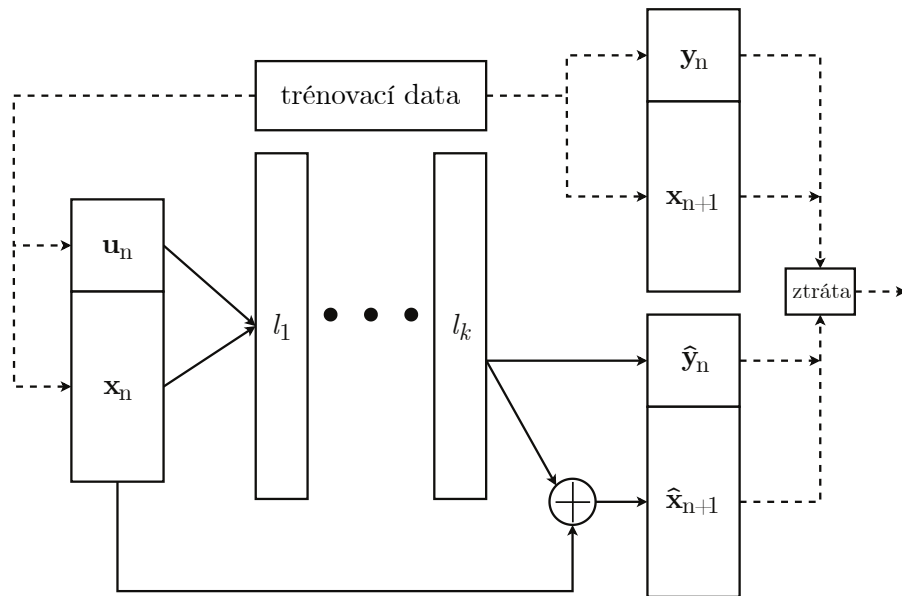
Jak již bylo řečeno, pomocí stavového popisu je vytvořen matematický nebo fyzikální model, kterým je popsán vývoj systému v čase pomocí sady diferenciálních rovnic. Rekurentní neuronové sítě, které jsou popsány v Kapitole 4.5, mají schopnost uchovávat informace z předchozích vstupů, které pak mohou ovlivnit výstup v aktuálním kroku. Tato vlastnost je velmi podobná konceptu stavu ve stavovém popisu, čímž jsou tedy rekurentní síť a stavový popis systému propojeny.

V kontextu modelování nelineárních systémů je při použití stavového popisu nelinearita řešena iterativně a pomocí aproximace funkcí popisující nelinearity systému např. pomocí vyhledávací tabulky. V článku [26] je popsáno, jak lze takové řešení nelinearit systému nahradit hlubokou neuronovou sítí. Je zde představena struktura nazývaná State Trajectory Network (STN), což je hluboká neuronová síť, která je zabudována do diskrétního stavového systému. Neuronové sítě jsou známé jako univerzální pro aproximace jakékoliv spojité funkce  $N$  proměnných. Systém pak lze také simulovat pomocí iterativně se opakujících operací, ale s tím rozdílem, že je použita aproximace funkce  $\hat{f}_d$  a ne funkce  $f_d$  odvozená analyticky pomocí diskretizace. Rovnice takového systému může potom mít tvar

$$\begin{pmatrix} \mathbf{x}[n+1] \\ \mathbf{y}[n] \end{pmatrix} = \hat{f}_d \begin{pmatrix} \mathbf{u}[n] \\ \mathbf{x}[n] \end{pmatrix}, \quad (3.2)$$

kde  $\mathbf{x}$  je vektor stavů systému,  $\mathbf{u}$  je vektor vstupů do systému a  $\mathbf{y}$  je vektor výstupů.

Pro správné učení dynamiky modelovaného systému je třeba mít k dispozici dostatečný počet naměřených veličin v porovnání s počtem stavů systému. Stavy systému představují veličiny, ve kterých je uložena energie nebo informace jako např. kondenzátory nebo cívky. Formální horní hranici počtu nezávislých měření lze určit pomocí Whitneyova teorému  $2m + 1$ , kde  $m$  je počet stavů. Takenův teorém potom umožňuje nahrazení jednoho či více měření časově zpožděnou verzí jiného již provedeného měření. V praxi však může být dynamika systému složitější, a proto je doporučeno využívat měření co nejbližší se blížící skutečným stavům systému. Na Obr. 3.2 je znázorněno blokové schéma navrhované sítě během trénování [26].

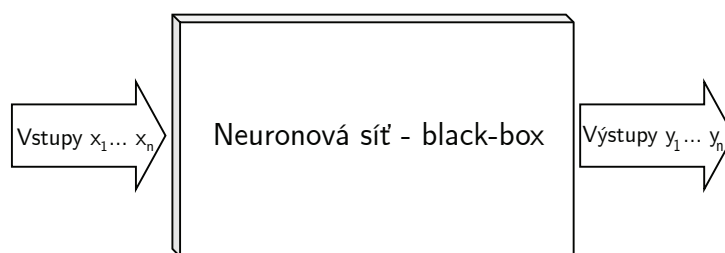


Obr. 3.2: Blokové schéma navrhované sítě během trénování [26].

## 4 Umělé neuronové sítě

Umělé neuronové sítě (ANN) jsou podoblastí oboru strojového učení a umělé inteligence. Jejich algoritmy jsou navrženy podle principu chování biologických neuronů [3, 29, 30]. Jejich počátek je datován přibližně do poloviny 20. století, kdy byla na základě zkoumání lidského mozku vytvořena první jednoduchá neuronová síť. Později byl představen umělý lineární model tzv. Perceptronu, nejjednodušší neuronové sítě, která byla sestavena z jednoho neuronu [7, 3, 29]. Tato síť byla schopna provádět pouze binární operace, a proto bylo později navrženo použití optimalizované nelinearity zvané aktivační funkce [7]. Většího rozmachu je pak dosaženo až s příchodem výkonnější výpočetní techniky [3, 29].

ANN jsou složeny z umělých neuronů, které jsou navzájem propojeny tak, že výstup jednoho neuronu je zároveň vstupem jednoho nebo více neuronů. Pomocí neuronů je zpracováván vstupní signál a generován signál výstupní. Každá síť je mimo jiné charakterizována typem neuronů, jejich uspořádáním a strategií adaptace při trénování sítě. Dráha pro šíření signálu je představována spoji mezi neurony, které jsou násobeny vahou, čímž je potom modifikována intenzita procházejícího signálu. Pokud spojení neexistuje, lze jej modelovat spojem násobeným vahou rovnou nule [29, 31].



Obr. 4.1: Obecné schéma neuronové sítě.

### 4.1 Umělý neuron

Jak již bylo řečeno, ANN jsou složeny z umělých neuronů. Mezi umělými neurony jsou předávány informace pomocí tzv. aktivací. Princip fungování umělého neuronu je analogický k fungování biologického, ale biologické funkce jsou změněny na matematické [29]. Neuron může být rozdělen na synapse, kterými jsou do neuronu přiváděny vstupy  $\mathbf{x} = (x_1, x_2, \dots, x_n)$ , násobeny vahami  $\mathbf{w} = (w_1, w_2, \dots, w_n)$ , kterými je ovlivněn příspěvek daného vstupu ve vnitřním potenciálu, tělo neuronu, ve kterém je získáván vnitřní potenciál  $\xi$  jako suma příspěvků jednotlivých vstupů,

blok aktivační funkce a výstup neuronu  $y$ . Pro vnitřní potenciál neuronu  $\xi$  platí

$$\xi = b + \sum_{i=1}^n x_i w_i, \quad (4.1)$$

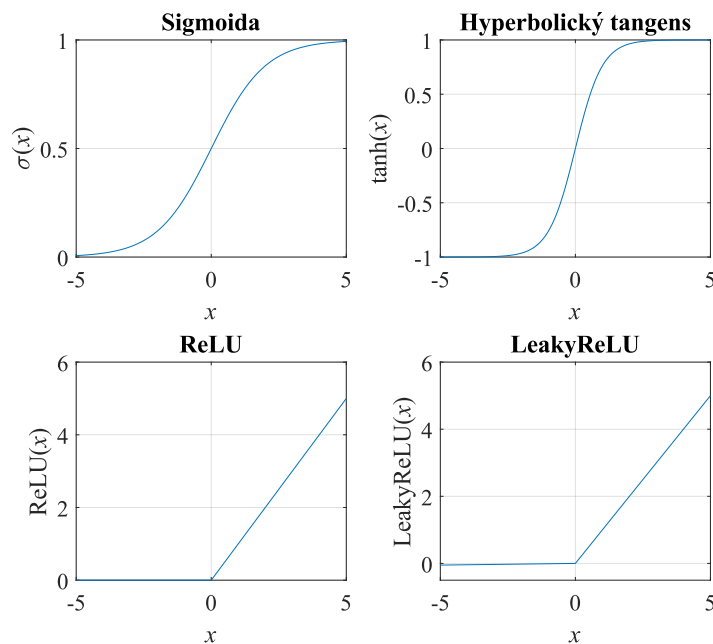
kde  $b$  je prahová hodnota pro vnitřní potenciál zvaná bias,  $x_i$  je vstup a  $w_i$  váha. Výstup sumační funkce bývá přiveden na vstup nelineární aktivační funkce. Touto funkcí je ovlivněn způsob převodu vnitřního potenciálu na výstup. Výstup neuronu je vyjádřen vztahem

$$y = f(\xi), \quad (4.2)$$

kde  $f$  je aktivační funkce a  $\xi$  je vnitřní potenciál neuronu [3, 30, 31].

## 4.2 Aktivační funkce

Aktivační funkci lze obecně definovat jako nelineární funkci, pomocí které je transformován vnitřní potenciál neuronu na výstupní signál. Nelinearita, kterou tyto funkce přináší do neuronové sítě je důležitá pro řešení komplexnějších problémů. Obvykle jsou používány funkce: sigmoida, hyperbolický tangens, ReLU (*Rectified Linear Unit*) nebo Leaky ReLU. Na Obr. 4.2 jsou zobrazeny jejich tvary [7, 3, 31].



Obr. 4.2: Aktivační funkce.

Sigmoida je nelineární funkce, kterou lze popsat vztahem

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \quad (4.3)$$

Funkce přijímá reálné hodnoty, které následně omezí do rozsahu  $\langle 0, 1 \rangle$ . V praxi to znamená, že vysokým záporným hodnotám je přiřazena hodnota 0 a vysokým kladným hodnotám je přiřazena hodnota 1. V klasifikačních sítích není jako aktivační funkce příliš využívána a to ze dvou hlavních důvodů. První z nich je skutečnost, že pokud dojde k saturaci funkce v bodech 0 nebo 1, gradient (viz [32]) v je těchto místech téměř nulový. Pokud je lokální gradient velmi malý, dojde k ovlivnění globálního gradientu, díky čemuž neproudí přes neuron do jeho vah ani dat téměř žádný signál. Je tedy důležité dbát na to, aby váhy nebyly příliš velké a nedošlo tak k saturaci neuronů, což by v síti vedlo k neschopnosti učit se. Druhý z důvodů je skutečnost, že je to nulově centrovaná funkce. Neurony v následných vrstvách by v tomto případě byla přijímána data, která nejsou centrována okolo nuly, čímž by byla ovlivněna dynamika systému během gradientního sestupu a mohla by zde být zavedena nežádoucí cik-cak dynamika v aktualizacích gradientu pro váhy. Gradient na vahách  $w$  je v tomto případě vždy kladný nebo vždy záporný v závislosti na gradientu celé funkce [33]. Sigmoida je však zcela běžně využívána v Long Short-Term Memory a Gated Recurrent Unit buňkách, které jsou popsány později v Kapitole 4.5.

Funkcí hyperbolický tangens jsou narozdíl od sigmoidy hodnoty omezeny do rozsahu  $\langle -1, 1 \rangle$ . Také dochází k saturaci, ale funkce je nulově centrovaná, takže jsou odstraněny nevýhody předchozí funkce a je tedy hyperbolický tangens preferován před sigmoidou. Matematicky lze funkci vyjádřit [33]

$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1}. \quad (4.4)$$

Funkci Rectified Linear Unit (ReLU) lze popsat vztahem

$$\text{ReLU}(x) = \max(0, x). \quad (4.5)$$

Výstupy funkce jsou nastaveny na nulu, pokud jsou vstupní hodnoty záporné. Kladné vstupní hodnoty jsou vráceny jako nesaturované lineární výstupy. Hlavní výhodou této funkce je poskytnutí vysoké rychlosti konvergence stochastického gradientního sestupu. Hlavní nevýhodou je potom tzv. *umírající ReLU*. V praxi to znamená, že by např. velký gradient procházející neuronem ReLU mohl způsobit takovou aktualizaci vah, že by neuron nemusel být už nikdy aktivován, čímž je potom způsobeno, že procházející gradient od tohoto bodu bude navždy nulový. Jinými slovy, pokud je během učení sítě váha neuronu nastavena tak že výstup ReLU je vždy menší než 0 pro většinu vstupů, neuron se stává neaktivním a neprodukuje žádný výstup. Výsledkem je tedy „mrtvý“ neuron, který nemůže být nadále využíván v procesu učení. K zamezení tohoto problému byla představena tzv. Leaky ReLU, která místo nastavení nulové výstupní hodnoty pro záporné vstupní hodnoty nastavuje tyto

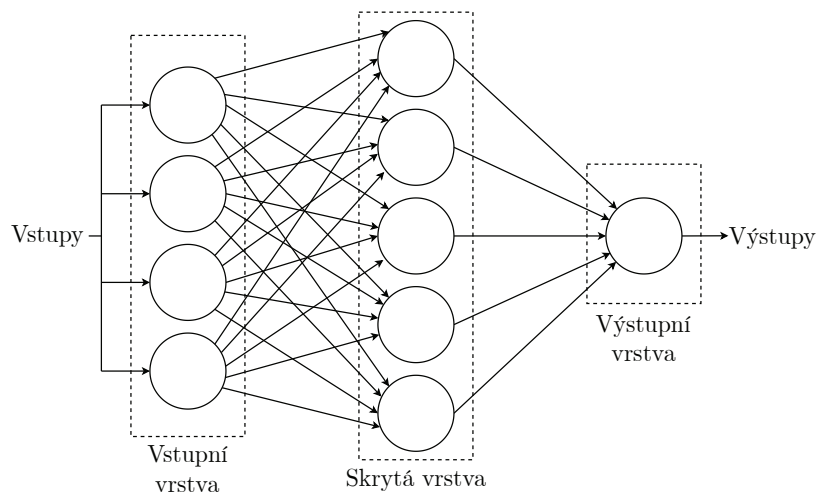
hodnoty na hodnoty blížíící se nule. Matematicky lze popsat jako

$$f(x) = \begin{cases} x & \text{pro } x > 0 \\ \alpha x & \text{pro } x \leq 0 \end{cases}, \quad (4.6)$$

kde  $\alpha$  je konstanta, která bývá označována jako malá konstanta nebo také sklon. Pomocí ní je určen sklon funkce pro záporné hodnoty. Tato forma aktivační funkce vykazuje úspěchy, ale výsledky nejsou vždy konzistentní. V článku [34] je uvedeno, že ze sklonu v negativní oblasti lze také vytvořit parametr každého neuronu. Zde se jedná o neurony PReLU [33].

### 4.3 Dopředné sítě

Tento typ sítě je základním konceptem vícevrstevných neuronových sítí. Nejběžnější typ takové sítě je potom vícevrstvý perceptron (MLP). Příklad MLP je zobrazen na Obr. 4.3. Oproti jednovrstvému perceptronu je síť rozšířena o minimálně jednu tzv. skrytou vrstvu, která je umístěna mezi vrstvou vstupní a výstupní. Počet vrstev odpovídá termínu hloubka. Čím více vrstev neuronová síť obsahuje, tím je hlubší (např. hluboká neuronová síť) [7]. Neurony jsou tedy v případě vícevrstvé sítě rozděleny dle vrstev na vstupní, skryté a výstupní. Počet neuronů ve skrytých vrstvách je volen dle charakteru úlohy pro kterou je neuronová síť využita [3, 7, 31].



Obr. 4.3: Příklad MLP.

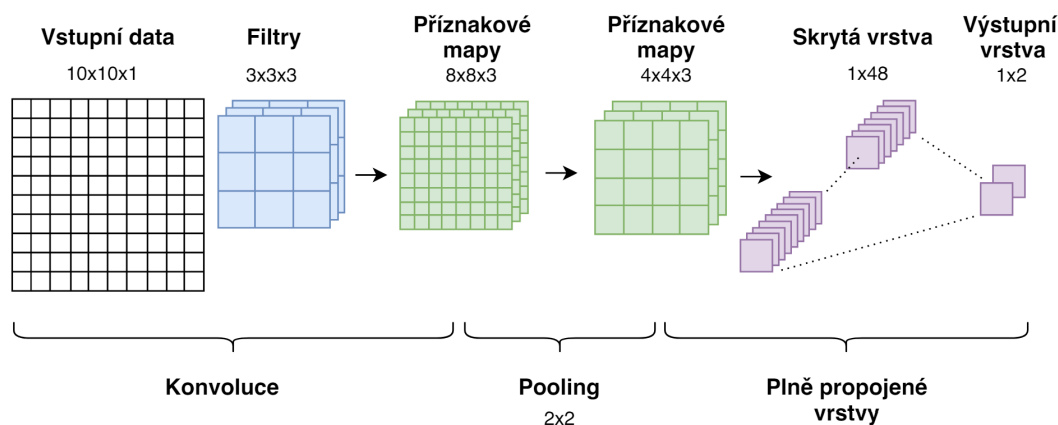
Spojení jednotlivých neuronů je realizováno pouze ve směru od vstupu k výstupu, a to takovým způsobem, že neurony sousedních vrstev jsou plně propojeny, ale neurony v rámci jedné vrstvy propojeny nejsou. Informace v takové síti jsou potom šířeny a zpracovávány pomocí změn stavů neuronů ve skrytých vrstvách [7, 3, 31].

Aktivační funkce skryté a výstupní vrstvy jsou stejné pro celou vrstvu. Síť lze tedy považovat za kauzální, neboť její výstup závisí pouze na aktuálních vstupních datech [3].

Tento typ sítě byl zprvu velmi využíván, ale dnes je již překonán jak z hlediska přesnosti, tak z hlediska výpočetní náročnosti. S vyšším počtem neuronů ve vrstvách, z důsledku plně propojených vrstev, exponenciálně narůstá počet vah a výpočtů [3].

## 4.4 Konvoluční síť

Konvoluční síť (CNN) jsou rozšířením dopředných neuronových sítí, které je využíváno především ke zpracování obrazu. Základní schéma konvoluční sítě pro práci s 2D daty je znázorněno na Obr. 4.4. Jejich princip je také založen na výpočtu a transformaci vnitřního potenciálu, ale neurony ve skrytých vrstvách jsou v případě této sítě uspořádány do tzv. *příznakových map* (Feature Maps), které mají určitou šířku, výšku a hloubku [3, 7, 30].



Obr. 4.4: Základní schéma konvoluční sítě – převzato z [3].

Pro sekvenční učení jsou CNN využívány v temporálních konvolučních sítích (TCN). Tyto sítě mají flexibilní a jednoduchou strukturu. Jedná se také o alternativu rekurentní neuronové sítě, která je zde blíže popsána později. Na rozdíl od rekurentních sítí, kde je pro časový kontext vstupních dat využito zpětných vazeb, je zde pro tuto vlastnost využito filtrů s rostoucí šířkou napříč vrstvami. Výhodou oproti rekurentním sítím je vyšší rychlost a přesnost při učení časových sekvencí. Rychlost při učení je zvýšena právě díky absenci zpětných vazeb [3, 35].

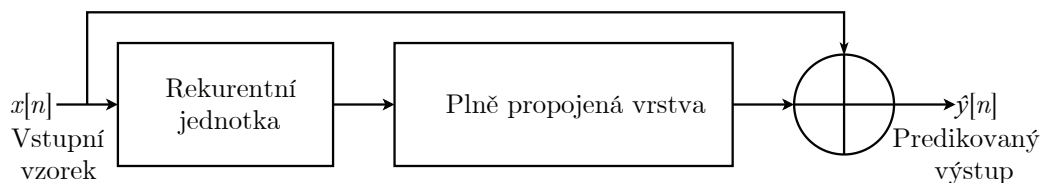
Stejně jako u CNN jsou v síti obsaženy sdílené filtry pro každou z vrstev, ale v případě TCN mají pouze 2D rozměr, tedy šířku a výšku. Šířkou je znázorněno časové rozpětí pro výpočetní operace filtru a hloubka pak může odpovídat počtu zpracovávaných vzorků v jednom časovém kroku. Slouží tedy ke skenování vstupních dat v čase, díky čemuž se model může učit z různých časových okamžiků v datech.

Důležité je, že stejný filtr se stejnými váhami je používán na různých místech v časové sekvenci. To znamená, že se model učí detekovat stejný vzorec na různých místech v čase [3].

Klíčovými pojmy z této oblasti jsou dilatační faktor  $d = 2^D$ , kde  $D$  je počet vrstev sítě a pole vjemu (*Receptive Field*)  $R = (k - 1)2^{D-1}$ , kde  $k$  je základní šířka filtru. Dilatačním faktorem je určeno, o kolik časových kroků je rozšířen filtr v dané vrstvě. Pokud bude tedy např.  $d = 2$ , bude konvoluční filtr aplikován na každý druhý vzorek vstupní sekvence. Čím více vrstev neuronová síť obsahuje, tím více je rozšířeno pole vjemu a výstup aktuálního kroku učení je ovlivněn vzdálenějšími prvky ve vstupních datech. Pro zachování časového rozlišení vstupních dat je nutné, aby byly všechny skryté vrstvy stejně dlouhé, což je provedeno doplněním nul (*zero padding*) pro obě strany vrstvy o délce pro danou vrstvu odpovídající  $(k - 1)d \cdot 2^{-1}$  [3].

## 4.5 Rekurentní síť

Jak zde bylo dříve uvedeno, struktura rekurentních sítí RNN (*Recurrent Neural Networks*) je odlišná od předchozích především obsažením zpětných vazeb, pomocí kterých je výstup neuronů přiveden zpět na jejich vstup. Těmito zpětnými vazbami je vytvořena forma paměti, díky které je síť schopna zpracovávat časově proměnné libovolně dlouhé posloupnosti, ve kterých bývá potřeba hledání souvislostí mezi vstupními daty u delších časových úseků [3, 30]. Na tuto paměť může být nahlíženo jako na stavový vektor, ve kterém jsou obsaženy informace o historii minulých prvků sekvence [18, 36]. Tyto stavy jsou využity a aktualizovány v každém časovém kroku, díky čemuž je RNN schopna pracovat v každém časovém kroku pouze s jedním vzorkem jako vstupem, přičemž jsou stále využity informace z předešlých časových kroků [18]. Model RNN je zobrazen na Obr. 4.5.

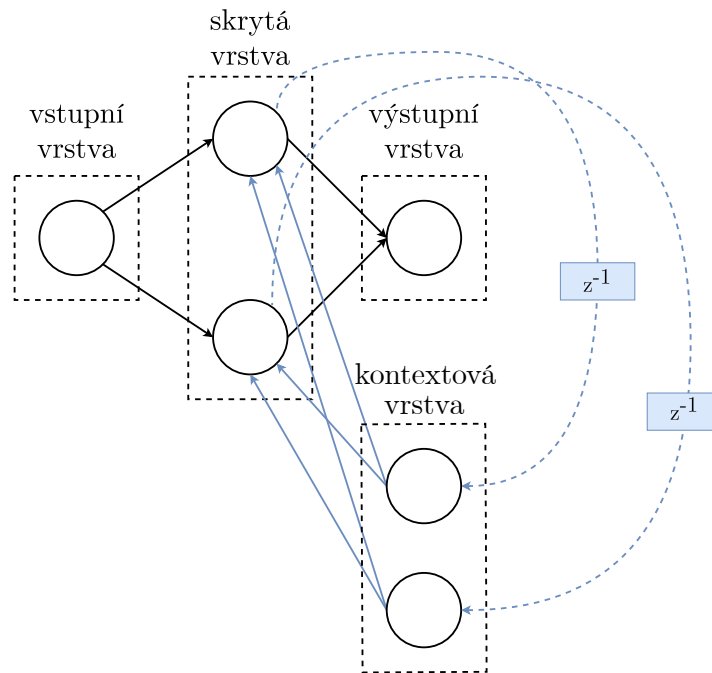


Obr. 4.5: Model rekurentní neuronové sítě [18].

Učení RNN je možné provádět s učitelem (viz [31]) analogicky k MLP, tedy pomocí vstupních a cílových vektorů, které jsou předány síti. Tento způsob je však pro tento typ neefektivní. Efektivnější způsob učení je takový, že vstupní a cílové data jsou měněny v diskretních časových krocích  $n$  v rámci jedné epochy. Síť je potom trénována na vstupních a cílových sekvencích. Tímto typem učení je využíváno zpětné

šíření chyb v čase BPTT (*Backpropagation Through Time*), které je zde popsáno později. Síť je rozvinuta na vzorky diskrétní v čase, čímž jsou odstraněny zpětné vazby. Nevýhodou je zvýšení počtu vrstev sítě a to přímo uměrně počtu časových kroků. Je potom ale možné použití principu gradientního sestupu [37]. Gradientní sestup je zde později popsán podrobněji v Kapitole 4.6. Další nevýhodou je snížení efektivity sítě a zvýšení výpočetní náročnosti [38]. Dalším problémem při učení sítě může být např. mizení gradientů, které je zapříčiněno jejich růstem nebo zmenšením v každém kroku při zpětném šíření [36].

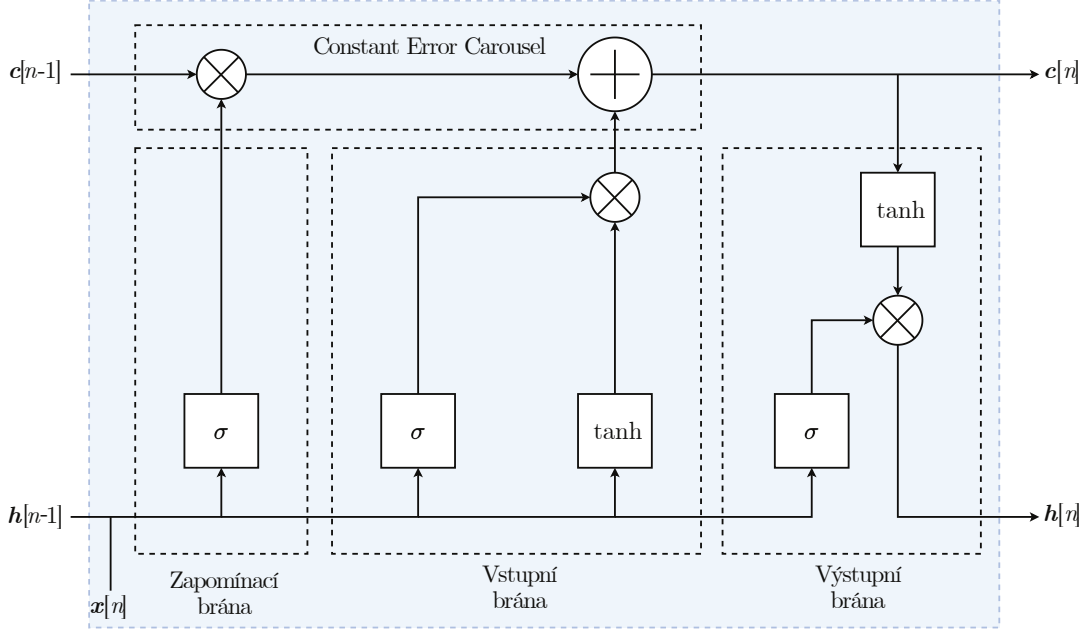
Elmanova rekurentní neuronová síť byla představena v roce 1990 americkým psycholingvistou Jeffrey Elmanem. Její struktura je zobrazena na Obr. 4.6. Jedná se o neuronovou síť, kde jsou neurony ve skryté vrstvě rekurentně spojeny a přijímají tedy signály samy od sebe a také z neuronů ve vstupní vrstvě. V síti je obsažena kontextová vrstva, kterou jsou reprezentována tato rekurentní spojení uložená pomocí zpožděných signálů [39].



Obr. 4.6: Elmanova síť.

Long Short-Term Memory (LSTM) výpočetní buňka byla poprvé představena Hochreiterem a Schmidhuberem v publikaci [40]. Jedná se o metodu, která je navržena např. pro minimalizaci problému mizejících gradientů při sledování delších časových závislostí v RNN. Díky LSTM je také navýšena efektivní paměť sítě z řádů desítek až na tisíce časových kroků [3]. Schéma buňky je zobrazeno na Obr. 4.7.

Stav buňky je tvořen vektorem vnitřních stavů  $\mathbf{c}[n]$ , kterým je umožněno uchování dlouhých časových závislostí, a vektorem skrytých stavů  $\mathbf{h}[n]$ , který představuje paměť krátkodobou. Celá buňka je potom tvořena několika branami. Jedná se



Obr. 4.7: LSTM buňka.

o bránu zapomínací, kde je rozhodováno o odstranění informací z vnitřního stavu buňky, bránu vstupní, kde je rozhodováno o zařazení nových informací do vnitřního stavu buňky, a bránu výstupní, ve které je určeno, jakým způsobem je aktualizován skrytý stav buňky [3, 18]. Jádrem buňky je však jednotka Constant Error Carousel (CEC). Jedná se o jednotku s vnitřní zpětnou vazbou a jednotkovým přenosem, která je základem při řešení problému mizejících gradientů [40].

V každém časovém kroku je využit vstupní vzorek buňky  $\mathbf{x}[n]$  a počáteční stavy  $\mathbf{c}[n - 1]$  a  $\mathbf{h}[n - 1]$ , čímž jsou vytvořeny výstupy LSTM, tedy aktualizované stavy  $\mathbf{c}[n]$  a  $\mathbf{h}[n]$ . Způsob využití vstupních a počátečních stavů pro aktualizace stavů lze popsat následujícími vztahy

$$\mathbf{i}[n] = \sigma(\mathbf{w}_{ii}\mathbf{x}[n] + b_{ii} + \mathbf{w}_{hi}\mathbf{h}[n - 1] + b_{hi}), \quad (4.7)$$

$$\mathbf{f}[n] = \sigma(\mathbf{w}_{if}\mathbf{x}[n] + b_{if} + \mathbf{w}_{hf}\mathbf{h}[n - 1] + b_{hf}), \quad (4.8)$$

$$\tilde{\mathbf{c}}[n] = \tanh(\mathbf{w}_{ic}\mathbf{x}[n] + b_{ic} + \mathbf{w}_{hc}\mathbf{h}[n - 1] + b_{hc}), \quad (4.9)$$

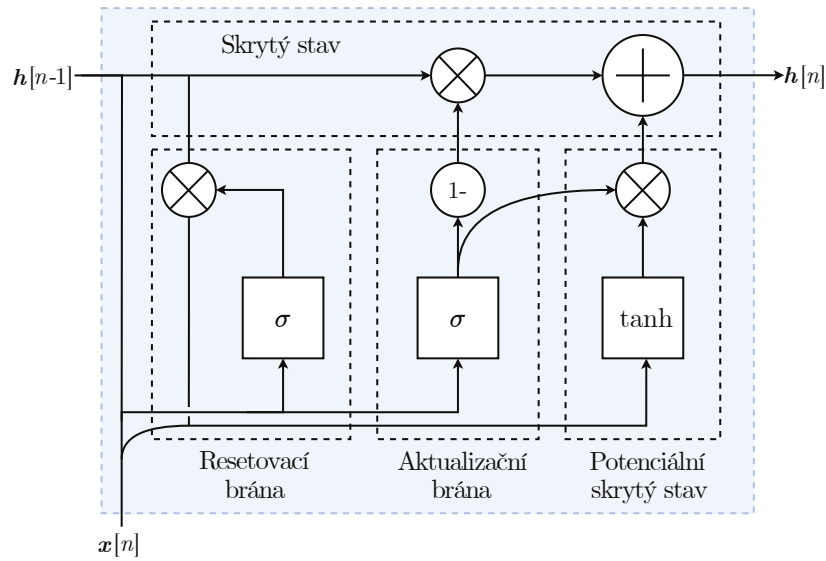
$$\mathbf{o}[n] = \sigma(\mathbf{w}_{io}\mathbf{x}[n] + b_{io} + \mathbf{w}_{ho}\mathbf{h}[n - 1] + b_{ho}), \quad (4.10)$$

$$\mathbf{c}[n] = \mathbf{f}[n]\mathbf{c}[n - 1] + \mathbf{i}[n]\tilde{\mathbf{c}}[n], \quad (4.11)$$

$$\mathbf{h}[n] = \mathbf{o}[n]\tanh(\mathbf{c}[n]), \quad (4.12)$$

kde  $\mathbf{i}[n]$  je vstupní brána,  $\mathbf{f}[n]$  je zapomínací brána,  $\tilde{\mathbf{c}}[n]$  je potenciální stav buňky a  $\mathbf{o}[n]$  je výstupní brána.  $w$  potom značí příslušné váhy a  $b$  příslušné prahy [18].

Gated Recurrent Unit (GRU) je alternativní výpočetní buňka, která je využívána v RNN podobně jako na Obr. 4.5. Zde je narozdíl od LSTM stav buňky tvořen pouze z jednoho skrytého stavového vektoru  $\mathbf{h}$ . Celá buňka je pak tvořena ze dvou bran a dvou stavů. Jedná se o skrytý a potenciální skrytý stav a brány resetovací a aktualizací, kterými je kontrolován vnitřní stav buňky, přičemž restartovací branou je určeno množství minulých dat, kterým je ovlivněn skrytý stav. Funguje tedy obdobně jako brána zapomínací u LSTM. Aktualizační branou je u tohoto typu buňky nahrazena funkce zapomínací a vstupní brány. Pomocí její odezvy je transformován potenciální skrytý stav na nový skrytý stav [3, 38]. Celá GRU buňka je zobrazena na Obr. 4.8.



Obr. 4.8: GRU buňka.

V každém časovém kroku je na vstupu využit aktuální vstup  $\mathbf{x}[n]$  a předchozí skrytý stav  $\mathbf{h}[n - 1]$ . Výstupem buňky je potom jediný aktualizovaný skrytý stav, který je vypočítán dle následujících vztahů

$$\mathbf{r}[n] = \sigma(\mathbf{w}_{ir}\mathbf{x}[n] + b_{ir} + \mathbf{w}_{hr}\mathbf{h}[n - 1] + b_{hr}), \quad (4.13)$$

$$\mathbf{z}[n] = \sigma(\mathbf{w}_{iz}\mathbf{x}[n] + b_{iz} + \mathbf{w}_{hz}\mathbf{h}[n - 1] + b_{hz}), \quad (4.14)$$

$$\tilde{\mathbf{h}}[n] = \tanh(\mathbf{w}_{ih}\mathbf{x}[n] + b_{ih} + \mathbf{w}_{hh}\mathbf{h}[n - 1] + b_{hh}), \quad (4.15)$$

$$\mathbf{h}[n] = (1 - \mathbf{z}[n])\tilde{\mathbf{h}}[n] + \mathbf{z}[n]\mathbf{h}[n - 1], \quad (4.16)$$

kde  $\mathbf{r}[n]$  je resetovací brána,  $\mathbf{z}[n]$  je aktualizací brána a  $\tilde{\mathbf{h}}[n]$  je potenciální skrytý stav.  $w$  a  $b$  zde opět značí příslušné váhy a prahy [3, 38].

## 4.6 Optimalizace sítě

Hlavní úlohou optimalizace neuronové sítě je minimalizace ztrátových funkcí, které jsou podrobněji popsány v následující podkapitole. Této minimalizace lze docílit pomocí různých optimalizačních algoritmů, které jsou popsány níže. Konkrétně se zde jedná o gradientní sestup, stochastický gradientní sestup (SGD), RMSProp a Adam.

Nejdříve je zde nutný popis algoritmu zpětného šíření chyby, který je úzce spojen s později popsaným optimalizačním algoritmem gradientního sestupu. Jedná se o klíčový algoritmus pro učení neuronové sítě. Po průchodu vstupních dat od vstupu k výstupu neuronové sítě, jsou na jejím výstupu vypočteny rozdíly mezi predikovanými a cílovými hodnotami a tím tedy dojde k určení hodnoty ztrátové funkce. Po výpočtu chyby jsou vypočteny parciální derivace této funkce vzhledem ke všem parametřům neuronové sítě a tím tedy k výpočtu gradientu (viz [32]) ztrátové funkce, který znázorňuje, jak moc každý parametr sítě ovlivnil výslednou chybu. Chyba z výstupní vrstvy je šířena strukturou sítě zpět k předcházející vrstvě, kde je chyba na výstupu modifikována derivací přenosové funkce neuronu výstupní vrstvy a poté vahou, která vede z této předcházející vrstvy k neuronu vrstvy výstupní. Každému neuronu skryté vrstvy je tedy přivedena chyba, která odpovídá příspěvku tohoto neuronu k celkové chybě na výstupní vrstvě. Pomocí optimalizačních algoritmů jsou potom modifikovány parametry sítě tak, aby došlo k minimalizaci ztrátové funkce.

Gradientní sestup je nejběžněji využívaným algoritmem pro optimalizaci neuronové sítě. Hlavní úlohou optimalizace je v tomto případě minimalizace ztrátové funkce  $J(\boldsymbol{\theta})$ . Symbolem  $\boldsymbol{\theta}$  jsou souhrnně označeny váhy  $\mathbf{w}$  a prahy  $\mathbf{b}$ . Cílem algoritmu je tedy nalezení optimálních parametrů  $\boldsymbol{\theta}^* = \operatorname{argmin} J(\boldsymbol{\theta})$ . Tento typ optimalizace je založen na předpokladech, že ztrátová funkce má jedno či více lokálních minim a že výstupy jednotlivých vrstev jsou diferencovatelné [3].

Algoritmem gradientního sestupu jsou na základě gradientu aktualizovány parametry  $\boldsymbol{\theta}$  pomocí již zmíněné metody zpětného šíření chyby. Toto lze popsat vztahem [3]

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}), \quad (4.17)$$

kde  $\eta$  je rychlost učení a  $\nabla_{\boldsymbol{\theta}}$  je gradient vzhledem k parametřům sítě  $\boldsymbol{\theta}$ . Gradient ztrátové funkce je potom spočten pomocí jejích parciálních derivací vzhledem k parametřům příslušných vrstev sítě. Tímto gradientem je značen růst ztrátové funkce. Parametry jsou potom adaptovány opačným směrem, což je docíleno pomocí záporného znaménka ve vztahu (4.17). Velikost změny je přímo úměrná tomuto gradientu, hodnotě chyby a parametru rychlosti učení. Problémem je, že pokud je nastavena příliš nízká hodnota tohoto parametru, konvergence chyby k možnému minimu probíhá příliš pomalu. V opačném případě pak nemusí ke konvergenci zcela dojít [3].

Další nevýhodou může být také velmi vysoká výpočetní náročnost zejména při více-dimenzionálních problémech, protože jsou prováděny výpočty pro celou trénovací datovou sadu. Řešením této nevýhody je metoda stochastického gradientního sestupu SGD. Jedná se taktéž o iterativní metodu, ale výpočet gradientu a změna parametrů je prováděna pouze pro náhodné trénovací dílčí datové sady. Díky tomu je tedy snížena výpočetní náročnost a zrychleny iterace algoritmu [60].

Algoritmus učení využívající zpětného šíření chyby spolu s algoritmem gradientního sestupu lze shrnout do následujících kroků [31]

1. Inicializace vah sítě.
2. Výběr vstupních a cílových dat.
3. Určení odezvy sítě.
4. Určení chyby na výstupu sítě.
5. Určení chyby na předcházejících vrstvách pomocí zpětného šíření chyby.
6. Lokální úpravy vah.
7. Opakování od bodu 2 do dostatečné přesnosti funkce.

Dalším optimalizačním algoritmem je např. RMSProp, což je optimalizační metoda úzce spojená s metodou Adam, která je zde popsána později. Na rozdíl od metody Adam jsou touto metodou generovány aktualizace svých parametrů pomocí hybnosti na změněném gradientu. Tato metoda také postrádá korekci zkreslení, což může v určitých případech vést k velmi velkým velikostem kroků a často k divergenci [59].

Další efektivní stochastickou optimalizační metodou je metoda zvaná Adam. Metoda vyžaduje pouze gradienty prvního řádu s malými nároky na paměť. Pomocí této metody jsou z odhadů prvního a druhého momentu gradientu vypočítávány adaptivní rychlosti učení pro různé parametry. Metoda je navržena tak, aby byly spojeny výhody populárních metod AdaGrad a RMSProp. Její výhody jsou například invariantnost aktualizace parametrů na změnu měřítka gradientu, ohraničení velikosti kroků hyperparametrem velikosti kroků nebo práce s řídkými gradienty [59].

## 4.7 Ztrátové funkce

Jak již bylo řečeno, neuronové sítě jsou trénovány k minimalizaci ztrátové funkce. V této sekci je uveden popis některých z nich. Konkrétně se jedná o střední absolutní chybu (MAE), střední kvadratickou chybu (MSE), poměr chyby k signálu (ESR), a ztrátu krátkodobé Fourierovy transformace (STFT).

Střední absolutní chyba je definována jako průměr absolutního rozdílu vstupní a predikované hodnoty [43]. Bývá využita v případě, kdy je třeba robustní metriky,

kteřá není příliš citlivá na odlehlé hodnoty (*outliers*). Je také využívána jako alternativa kategorické křížové entropické ztráty (CCE) pro zamezení snižování výkonu hlubokých neuronových sítí při jejich trénování [42]. Lze vyjádřit jako

$$\mathcal{L}_{\text{MAE}} = \frac{\sum_{n=1}^N |\hat{y}[n] - y[n]|}{N}. \quad (4.18)$$

Střední kvadratická chyba je definována jako průměr kvadratických rozdílů vstupní a predikované hodnoty. Bývá využita, např. když je třeba znát kvadratickou odchylku. Lze ji vyjádřit vztahem

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{n=1}^N |\hat{y}[n] - y[n]|^2. \quad (4.19)$$

Poměr chyby k signálu je ekvivalent kvadratické chyby mezi hodnotami cílového signálu  $y$  a hodnotami výstupního signálu neuronové sítě  $\hat{y}$  o délce  $N$  vzorků [41]. V článku [38] je při učení sítě použita ztrátová funkce založená na ESR s předfázovou filtrací, která je aplikována na výstupní a cílový signál před výpočtem ESR. Díky této předfázové filtraci je zvýšena přesnost modelování informací ve vysokých kmitočtech. ESR je dána vztahem

$$\mathcal{L}_{\text{ESR}} = \frac{\sum_{n=0}^{N-1} |y[n] - \hat{y}[n]|^2}{\sum_{n=0}^{N-1} |y[n]|^2}, \quad (4.20)$$

kde  $y$  je cílový signál po předfázové filtraci a  $\hat{y}$  je výstupní signál neuronové sítě po předfázové filtraci.

Při testování výstupního modelu byl zjištěn nežádoucí stejnosměrný ofset. Pro jeho minimalizaci byl zaveden stejnosměrný člen

$$\mathcal{L}_{\text{DC}} = \frac{|\frac{1}{N} \sum_{n=0}^{N-1} (y[n] - \hat{y}[n])|^2}{\frac{1}{N} \sum_{n=0}^{N-1} |y[n]|^2}. \quad (4.21)$$

Celková chyba je potom dána vztahem

$$\mathcal{L} = \mathcal{L}_{\text{ESR}} + \mathcal{L}_{\text{DC}}. \quad (4.22)$$

Ztráta krátkodobé Fourierovy transformace je složena ze spektrální konvergence 4.23 a spektrální logaritmické hladiny 4.24. Celková ztráta STFT je tedy definována jako suma těchto dvou částí. Pro jednotlivé části platí vztahy

$$\mathcal{L}_{\text{SC}} = \frac{\| |\text{STFT}(y)| - |\text{STFT}(\hat{y})| \|_{\text{F}}}{\| |\text{STFT}(y)| \|_{\text{F}}}, \quad (4.23)$$

$$\mathcal{L}_{\text{SM}} = \frac{1}{N} \|\log(|\text{STFT}(y)|) - \log(|\text{STFT}(\hat{y})|)\|_1, \quad (4.24)$$

kde  $\|\cdot\|_{\text{F}}$  je Frobeniova norma,  $\|\cdot\|_1$  je  $L_1$  norma a  $N$  je délka okna STFT [41].

## 5 Identifikace nelineárních systémů

Pro analýzu nelineárních zvukových systémů bývá velmi často využit exponenciální přeladovaný sinusový signál. V článku [44] je zkoumána teorie měření nelineárních systémů za pomoci tohoto signálu a postup synchronizace, který je nezbytný pro analýzu vyšších harmonických, a který vede k vylepšení vzorce pro generování sinusového přeladovaného signálu. Tento typ signálu je velmi relevantní pro tuto práci, neboť bývá použit také jako trénovací signál pro neuronové sítě při modelování nelineárních zvukových efektů. Je také použit jako trénovací signál v později zmíněných modelovacích zařízeních využívajících neuronové sítě.

Technika analýzy nelineárních systémů pomocí přeladovaného sinusového signálu byla nejprve představena Farinou v roce 2000 [45]. Touto technikou je umožněno oddělení impulsních odezev pro každý řád harmonického zkreslení. Později byla tato technika Farinou nazvána jako nelineární konvoluce. Matematicky je tento signál popsán jako

$$x(t) = \sin \left\{ \frac{2\pi f_1 T}{\ln(f_2/f_1)} \left[ \exp \left( \frac{t}{T} \ln \left( \frac{f_2}{f_1} \right) \right) - 1 \right] \right\}, \quad (5.1)$$

kde  $f_1$  je počáteční kmitočet,  $f_2$  je konečný kmitočet a  $T$  je doba trvání signálu [45].

Pro správný odhad amplitudy a fáze kmitočtové odezvy vyšších harmonických je však potřeba přeladovaný sinusový signál generovat jiným způsobem, a to

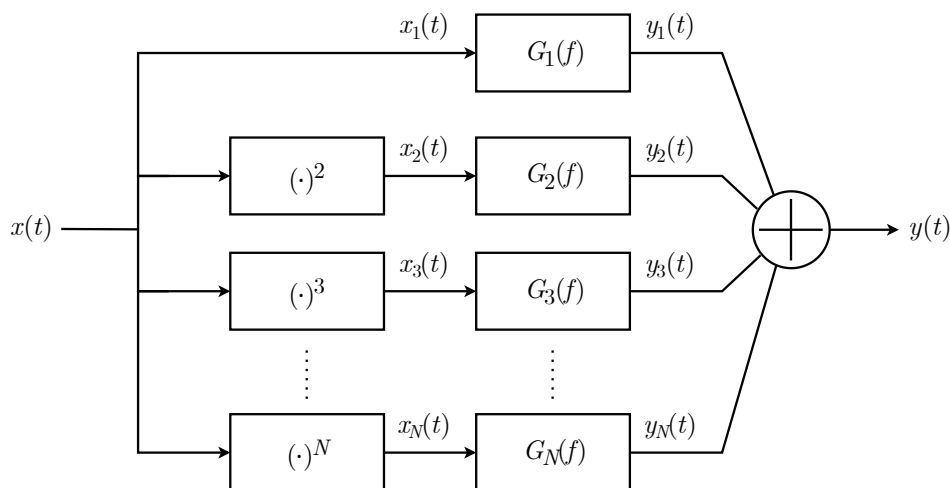
$$x(t) = \sin \left\{ 2\pi f_1 L \left[ \exp \left( \frac{t}{L} \right) - 1 \right] \right\}, \quad (5.2)$$

kde  $L$  je rychlost nárůstu kmitočtu, která je definována jako

$$L = \frac{1}{f_1} \text{round} \left[ \frac{\hat{T} f_1}{\ln \left( \frac{f_2}{f_1} \right)} \right], \quad (5.3)$$

kde  $\hat{T}$  je přibližná časová délka signálu  $x(t)$  a **round** představuje zaokrouhlení k nejbližšímu celému číslu. Toto zaokrouhlení je nezbytné pro správný odhad fáze kmitočtové odezvy vyšších harmonických [46].

V článku [46] je potom detailněji uvedena metoda využití přeladovaného sinusového signálu pro identifikaci nelineárních systémů založená na nelineární konvoluci. Metoda využívá neparametrický zobecněný polynomický Hammersteinův model (Obr. 5.1), který je vytvořen z mocninných řad spojených s lineárními filtry. Metoda vykazuje dobré výsledky a je schopna identifikace testovaného nelineárního systému a odhadu lineárních filtrů neznámého systému. V článku jsou mimo jiné uvedeny i další vhodné signály pro různé metody odhadu. Jedná se např. o buzení sinusovým signálem, vícetónové buzení, buzení pomocí náhodných šumů nebo pseudonáhodné signály.



Obr. 5.1: Zobecněný polynomický Hammersteinův model.

## 5.1 Trénovací signály v komerčních zařízeních

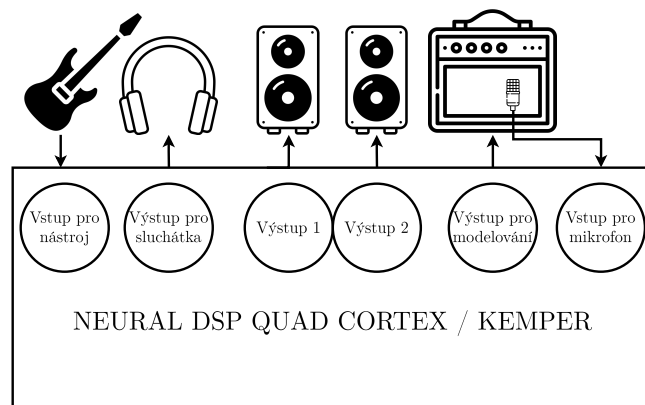
Tato podkapitola je zaměřena na popis komerčně využívaných produktů pro modelování kytarových zesilovačů či efektů a na srovnání trénovacích či testovacích signálů, které jsou pro tato modelování využity. Konkrétně se jedná o zařízení Quad Cortex od firmy Neural DSP a Profiler Stage od firmy Kemper. Je zde také uveden popis knihovny Neural Amp Modeler jakožto volně dostupné alternativy předchozích dvou zařízení. Jedná se o knihovnu programovacího jazyka Python pro modelování kytarových efektů nebo zesilovačů pomocí neuronových sítí.

Neural DSP Quad Cortex je dle uživatelského návodu [47] nejvýkonnější pedálové modelovací zařízení na světě, který díky svému výkonu dokáže zpracovávat až čtyři nástroje současně. Náhled na zařízení je zobrazen na Obr. 5.2.



Obr. 5.2: Neural DSP Quad Cortex [47].

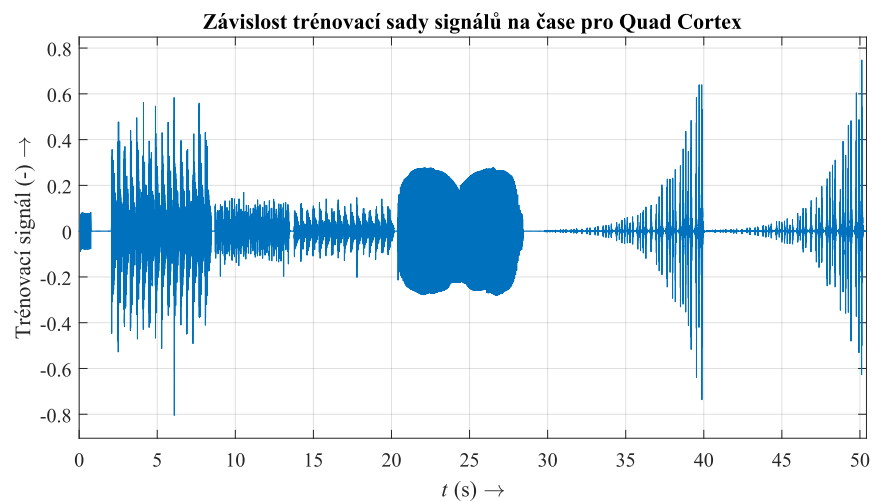
Zařízení nabízí proceduru zvanou Neural Capture, která pomocí biomimetické umělé inteligence umožňuje replikovat zvukové charakteristiky jakéhokoli fyzického zesilovače či efektu a to s velkou přesností. Do zařízení je nejprve zapojen hudební nástroj a poté je jeho výstup připojen ke vstupu replikovaného zařízení. Pokud se jedná o efekt nebo zesilovač, který nemá zabudovaný DI výstup (viz [50]) a musí být zapojen do reproboxu, je nutné připojení mikrofonu. Pokud se jedná o zařízení s DI výstupem, lze výstup zapojit přímo do vstupu Quad Cortexu. Schéma zapojení je zobrazeno na Obr. 5.3. Dále je před zahájením trénovacího procesu potřeba nastavení optimálních úrovní pro zamezení limitace signálů. Poté je zahájen modelovací proces. Nejprve je provedena kalibrace, kdy je měřena latence připojeného zařízení, poté jsou do replikovaného zařízení přivedeny trénovací signály, které jsou nahrány a použity v dalším kroku, kdy probíhá samotné trénování neuronové sítě pro vytvoření emulace požadovaného zvukového zařízení. Nakonec může být prováděno testování výsledného modelu.



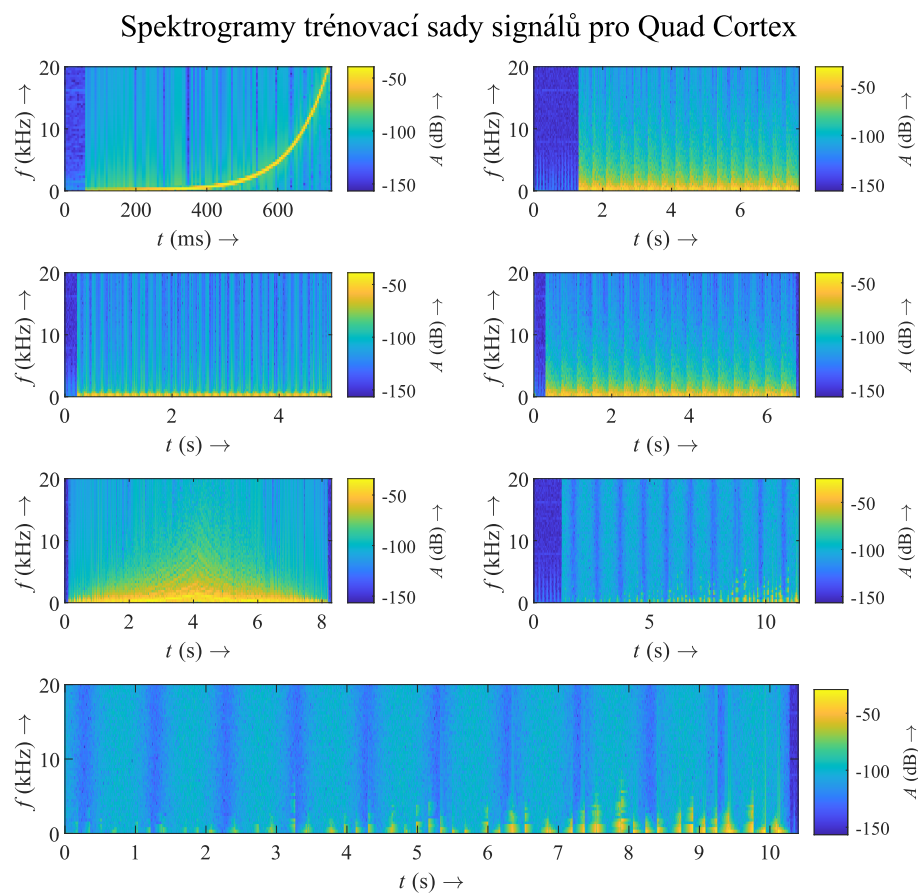
Obr. 5.3: Schéma zapojení pro modelovací proceduru [49].

Trénovací sada je složena z pěti částí a její délka je přibližně 50 sekund. Na Obr. 5.4 je zobrazena závislost trénovací sady signálů na čase a na Obr. 5.5 jsou zobrazeny jejich spektrogramy.

Vzhledem ke komplexnosti použitých signálů a jejich různým kombinacím je lze ve většině případů pojmenovat pouze podle subjektivních domněnek. V případě prvního signálu sady se však s nejvyšší pravděpodobností jedná o přeladovaný sinusový signál v kmitočtovém rozsahu od 0 Hz do 20 kHz o délce přibližně 0,7 s. Z poslechové analýzy a z vykresleného časového průběhu lze odvodit, že trénovací sada pokrývá různé transienční složky a různě dlouhé doby náběhu a doběhu. Mnoho informací lze také odvodit z vykreslených spektrogramů. Trénovací sadou je pokryto celé slyšitelné kmitočtové spektrum od 0 Hz do 20 kHz. Nejvíce energie je však zaměřeno na efektivní rozsah kmitočtového spektra pro elektrickou kytaru a baskytaru, tedy přibližně od 40 Hz do 10 kHz.



Obr. 5.4: Závislost trénovací sady signálů na čase – Quad Cortex.



Obr. 5.5: Kmitočtové spektrum trénovacích signálů – Quad Cortex.

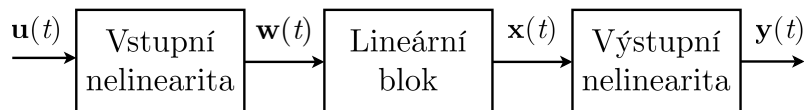
Firma Kemper nabízí několik produktů. V této práci byl využit konkrétně model Kemper Profiler Stage, tedy digitální kytarový zesilovač kombinovaný s nožním

přepínačem. Náhled je zobrazen na Obr. 5.6.



Obr. 5.6: Kemper Profiler Stage [48].

Zařízení nabízí technologii Kemper profiling, pomocí které je umožněno zachycení a modelování jakéhokoliv kytarového zesilovače [51]. Proces modelování probíhá obdobným způsobem jako u přechozího zařízení a schéma zapojení se shoduje (Obr 5.3). Také je využita série testovacích signálů, která je vysílána do vstupu modelovaného zařízení a opět přijímána modelovacím zařízením. Je však nutné podotknout, že se zde nejedná o proces využívající neuronovou síť, ale patentovanou technologii firmy Kemper, která zůstává obchodním tajemstvím. Dle patentu [52] se však jedná o model založený na Wiener-Hammersteinově struktuře [53], která je zobrazena na Obr. 5.7. Na tomto obrázku je vstupní signál značen jako  $\mathbf{u}(t)$ , výstupní signál vstupní nelinearity  $\mathbf{w}(t)$ , výstupní signál lineárního bloku  $\mathbf{x}(t)$  a výstupní signál systému  $\mathbf{y}(t)$ . Oproti předchozímu zařízení je zde také rozdíl v závěru modelování. Kemper zde využívá finální proces, kdy po hraní na kytaru zhruba dvacet sekund dojde k další analýze odezvy modelovaného zařízení a tím k dalšímu zvýšení autentičnosti výsledného modelu.

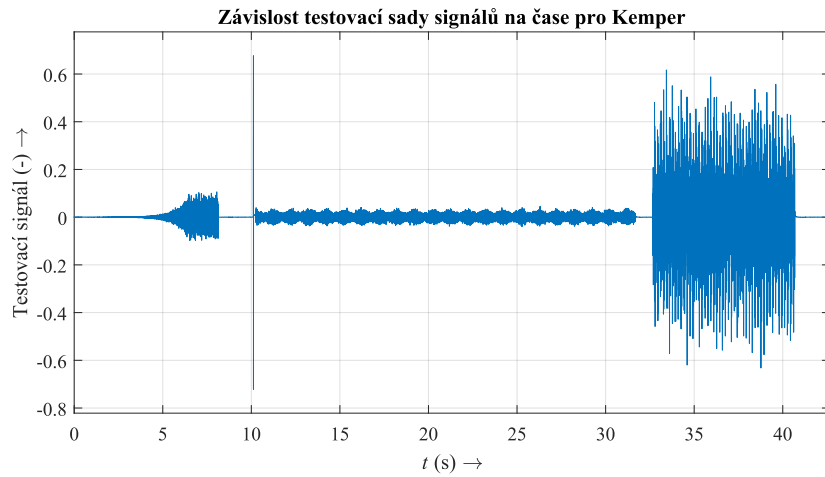


Obr. 5.7: Wiener-Hammersteinův model.

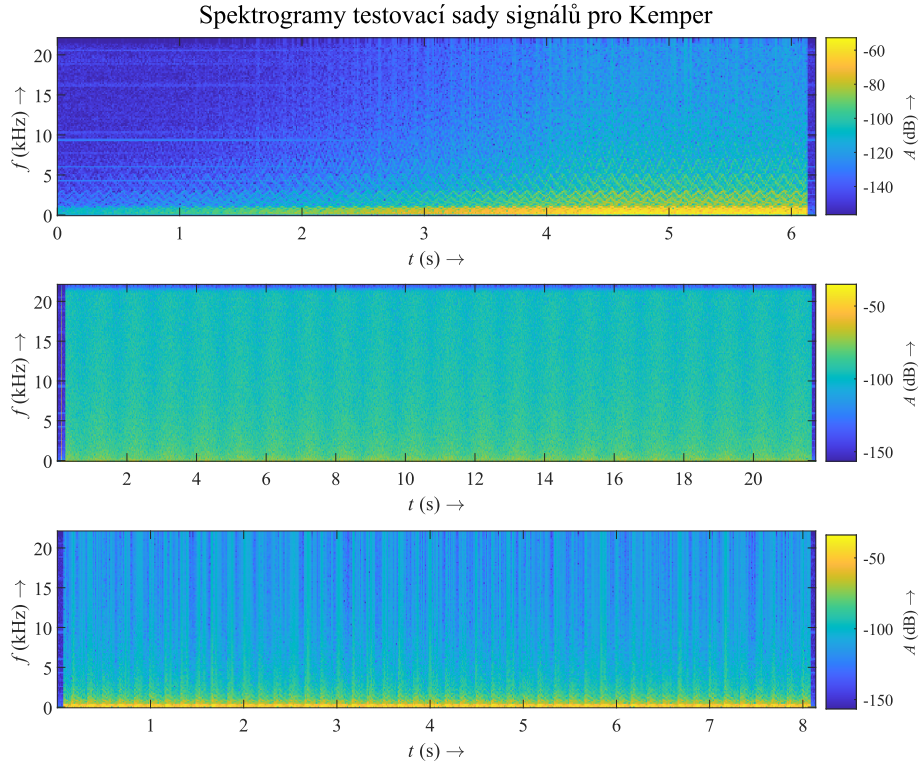
V případě tohoto zařízení se jedná o testovací signály, protože zařízení nevyužívá modelování pomocí neuronové sítě. Sada signálů je zde rozdělena do tří částí s celkovou délkou přibližně 41 sekund. Na Obr 5.8 je zobrazena závislost testovací sady signálů na čase a na Obr. 5.9 jsou zobrazeny spektrogramy jednotlivých signálů.

Stejně jako v předchozím případě lze jednotlivé signály pojmenovat pouze subjektivně. Rozdílem je však jejich menší počet, tedy i jejich menší pestrost. Jsou zde

využity dva podobně dlouhé signály jako z předchozí sady, ale prostřední signál je více než dvakrát delší a je v kmitočtové oblasti více zaměřen na celé slyšitelné kmitočtové spektrum než dva zbývající a také než signály použité u předchozího zařízení. Nejvíce energie v kmitočtovém spektru je však opět soustředěno na kmitočtech, kde je předáváno nejvíce informací o barvě zvuku elektrické kytary a baskytary.



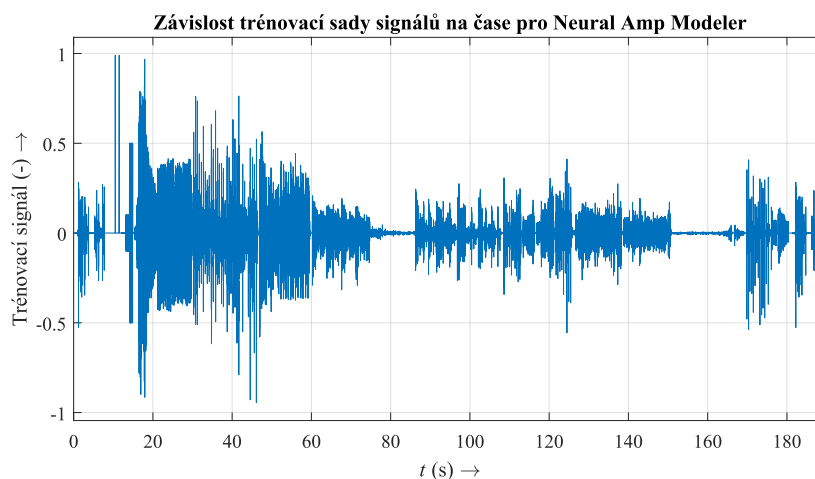
Obr. 5.8: Závislost testovací sady signálů na čase – Kemper.



Obr. 5.9: Kmitočtové spektrum trénovacích signálů – Kemper.

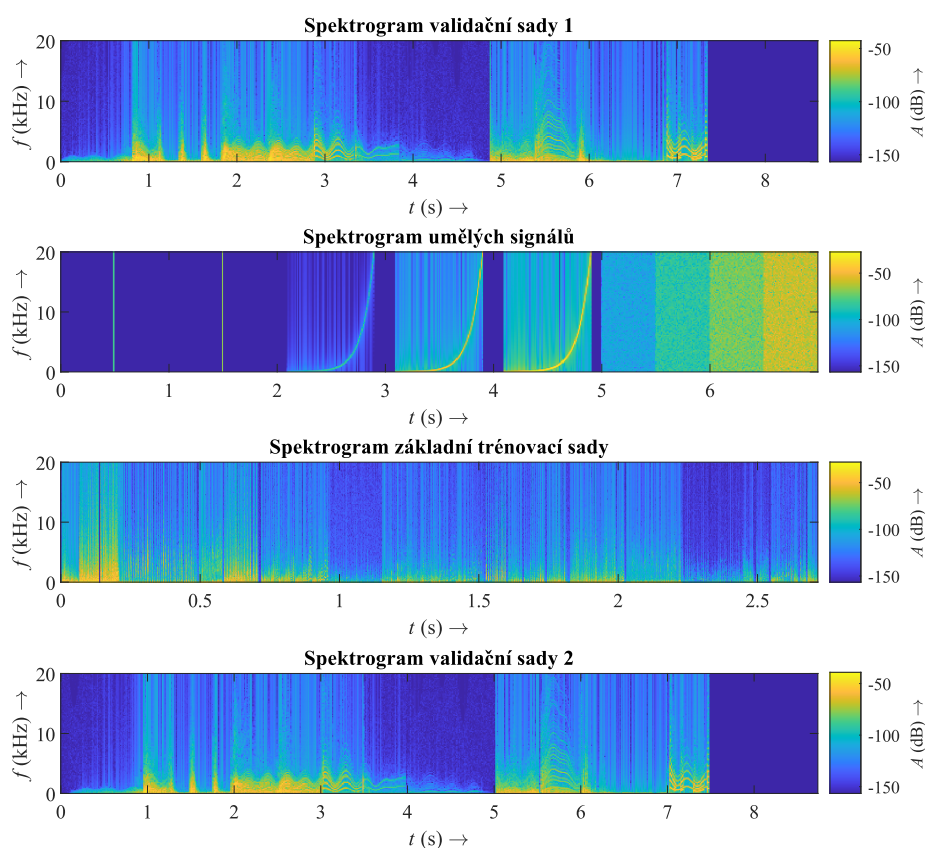
Neural Amp Modeler (NAM) je volně dostupnou verzí modelovacího nástroje založeného na modelování pomocí neuronových sítí. Jedná se o knihovnu programovacího jazyka Python, která nabízí vlastní navržený model neuronové sítě včetně trénovacích a validačních sad signálů, z nichž je pro využití doporučena nejnovější sada s názvem NAM\_v3. Na Obr 5.10 je zobrazena závislost trénovací a validační sady signálů na čase a na Obr. 5.11 jsou potom zobrazeny jejich spektrogramy. Celá sada je dlouhá přibližně 190 sekund, což je až čtyřikrát delší než sady signálů u předchozích dvou zařízení. Dle dokumentace této knihovny je rozdělena takto

- (0:00 – 0:09) Validační sada 1.
- (0:09 – 0:10) Ticho.
- (0:10 – 0:12) Jednotkové impulsy v časech 0:10.5 a 0:11.5.
- (0:12 – 0:15) Přeladované sinusové signály.
- (0:15 – 0:17) Šum.
- (0:17 – 3:00.5) Základní trénovací signály.
- (3:00.5 – 3:01) Ticho.
- (3:01 – 3:10) Validační sada 2.



Obr. 5.10: Závislost trénovací sady signálů na čase – NAM.

Sada se skládá se jak z uměle generovaných signálů, tak ze signálů znějících jako hra na elektrickou kytaru a baskytaru. Hra na tyto nástroje ovšem není klasická jako např. akordy, melodie apod. Jedná se spíše o zvuky znějící jako různé modulace a zkreslení zvuků vytvořených silnou excitací strun, použití páky pro vibrato při excitaci strun a další tomuto podobné zvuky. Co se týče umělých signálů, opět se zde vyskytuje přeladovaný sinusový signál, ale tentokrát ve třech různých variantách. Délky těchto signálů jsou 0,9 sekundy a kmitočtový rozsah je od 0 Hz do 20 kHz. Rozdílné jsou jejich amplitudy, kdy první signál má amplitudu 0,002, druhý 0,1 a třetí 0,5. Další z rozeznatelných umělých signálů je bílý šum, který se zde vysky-



Obr. 5.11: Kmitočtové spektrum sady signálů – NAM.

tuje ve čtyřech různých hladinách a to přibližně 0,002; 0,02; 0,1 a 0,2. Jejich délky jsou přibližně 0,5 sekundy. Vyskytují se zde také dva jednotkové impulsy (*blips*), které mohou představovat krátký náhlý skokový jev v signálu procházejícím elektronickým zesilovačem. Opět je trénovací sadou pokryto celé slyšitelné kmitočtové spektrum.

Při použití této sady signálů jako trénovací množiny pro neuronovou síť v praktické části této práce, byly výsledné modely natrénovány s lepšími objektivními výsledky při použití celé sady než při použití pouze její trénovací části. Proto byla tedy v celé praktické části využita celá tato sada signálů.

## 5.2 Návrh trénovacích signálů

V této podkapitole je uveden průběh návrhu sady umělých trénovacích signálů pro neuronovou síť a použitý software včetně s ním spojených dalších prostředků. K testování byla použita neuronová síť vytvořená pomocí programovacího jazyka Python verze 3.9.18 ve vývojovém prostředí PyCharm verze 2023.2.1 s využitím několika

knihoven na laptopu s grafickou kartou NVIDIA GeForce GTX 1650. Dále byl také použit program MATLAB a DAW Studio One 5 vyvíjené společností Presonus. V práci byly využity tyto knihovny programovacího jazyka Python

- **PyTorch** – knihovna pro strojové učení, která přináší efektivitu a podporu hardwarových akcelerátorů jako GPU. V knihovně jsou obsaženy funkce pro neuronové sítě, ztrátové či aktivační funkce a další [54].
- **PyTorch Lightning** – knihovna hlubokého učení pro výzkum umělé inteligence, která narozdíl od PyTorch poskytuje předchystané kódy např. pro trénovací proces a další [55].
- **PyTorch TensorBoard** – sada vizualizačních nástrojů pro experimentování se strojovým učáním, která umožňuje sledování a vizualizaci metrik, jako je ztráta a přesnost, vizualizace grafu modelu, prohlížení histogramů, zobrazování obrázků a mnoho dalšího [56].
- **Soundfile** – zvuková knihovna založená na knihovnách NumPy, libsndfile a CFFI, která umožňuje čtení a zápis zvukových souborů [57].
- **Weights & Biases (WandB)** – knihovna strojového učení pro vizualizaci dat, která mimo jiné umožňuje propojení s webovým prostředím, kde je možné zobrazovat a uchovávat data získané při učení sítě jako např. výstupní zvukové nahrávky, grafy ztrátových funkcí nebo grafy využití GPU a další [58].
- **Pedalboard** – knihovna vytvořená společností Spotify pro práci se zvukovými soubory, jako např. čtení, zápis, přidávání efektů a další. Podporuje širokou škálu zvukových formátů a řady zvukových efektů. Umožňuje také použití formátů VST3 a Audio Unit pro načítání softwarových nástrojů a efektů třetích stran, což je hlavním důvodem využití této knihovny v této práci [61].

Sady signálů popsané v předchozí kapitole jsou v této práci brány jako výchozí bod pro návrh sady umělých signálů, kterou má být nahrazena sada signálů složená z nahrávek hry na elektrickou kytaru a baskytaru jakožto trénovací množina pro neuronovou síť určenou k vytvoření modelů nelineárních zkreslovacích zvukových efektů. Jako referenční výsledek simulace byly tedy potom použity výsledné hodnoty natrénovaných modelů sítě pomocí kytarového signálu resp. referenční trénovací sady a to konkrétně hodnoty ztrátových funkcí ESR a MAE (kapitola 4.7) a cílové a výsledné výstupní signály neuronové sítě v podobě zvukových souborů. Pro samotné učení sítě byla potom použita ztrátová funkce ESR, zatímco MAE a MSE byly použity pouze jako validační metriky bez vlivu na změnu parametrů sítě během učení.

V rámci této práce byla tedy navržena sada signálů s názvem „Train 9“, která je dlouhá stejně jako referenční trénovací sada, tedy 240 sekund. Její časová závislost je zobrazena na Obr.5.12 a její spektrogramy na Obr. 5.13. Signály byly nejdříve generovány pomocí tónového generátoru dostupného v DAW Studio One 5 a poté pomocí

programu MATLAB. Pomocí DAW byly signály také efektovány a popř. stříhány nebo míchány. Generovány jsou potom s vzorkovacím kmitočtem  $f_{vz} = 44100$  Hz a bitovou hloubkou 24 bitů. Časové rozvržení celé sady je uvedeno v následujícím seznamu. Údaj v závorce na konci odrážky značí způsob generování signálu.

- (0:00 – 0:06,9) Přeladované sinusové signály, bílý šum a jednotkové impulsy (NAM).
- (0:06,9 – 0:10,3) Přeladovaný pilový signál (DAW).
- (0:10,3 – 0:16,3) Smíchané sinusové signály nejprve bez šumu a poté s šumem (MATLAB).
- (0:16,3 – 0:38,2) Postupně měněné akordy složené ze sinusových signálů bez šumu a poté s šumem (MATLAB).
- (0:38,2 – 0:48,2) Náhodné časově obrácené sinusové signály (MATLAB).
- (0:48,2 – 1:20,5) Zkreslené postupně měněné akordy složené ze sinusových signálů bez šumu a poté s šumem (MATLAB).
- (1:20,5 – 1:36,62) Modulovaný pilový signál (DAW).
- (1:36,67 – 1:57,52) Pilový signál s náhodně měněným kmitočtem v rozsahu 1 Hz až 22 kHz (DAW).
- (1:57,56 – 2:17,51) Náhodné sinusové signály s pomalým postupným náběhem (*fade in*) (MATLAB).
- (2:17,55 – 2:24,46) Sinusové signály o kmitočtech standardního ladění šestistrunné kytary (MATLAB).
- (2:24,46 – 2:31) Sinusové signály o kmitočtech standardního ladění čtyřstrunné baskytary (MATLAB).
- (2:31 – 2:51,04) Sinusové signály o kmitočtech standardního ladění čtyřstrunné baskytary zkreslené funkcí popisující elektronkový zesilovač a poté měkký ořez signálu (MATLAB).
- (2:51,11 – 2:57,13) Přeladované sinusové signály o kmitočtech standardního ladění šestistrunné kytary o dvě oktávy nahoru, zpět a o oktávu dolů (MATLAB).
- (2:57,14 – 3:01,96) Sinusové signály o kmitočtech standardního ladění čtyřstrunné baskytary zkreslené měkkým ořezávačem o oktávu výše než předchozí (MATLAB).
- (3:02 – 3:12,01) Sinusové signály o kmitočtech standardního ladění šestistrunné kytary zpracované kompresorem s automatizovaným nastavením parametrů attack a release (MATLAB + DAW).
- (3:12,2 – 3:27,2) Osm zkombinovaných přeladovaných sinusových signálů v rozmezí kmitočtů 20 Hz až 6 kHz (MATLAB).
- (3:27,4 – 3:31,98) Zkreslený akord složený ze sinusových signálů zpracovaný efektem tremolo (MATLAB + DAW).
- (3:31,99 – 3:42) Sinusové signály o kmitočtech standardního ladění šestistrunné

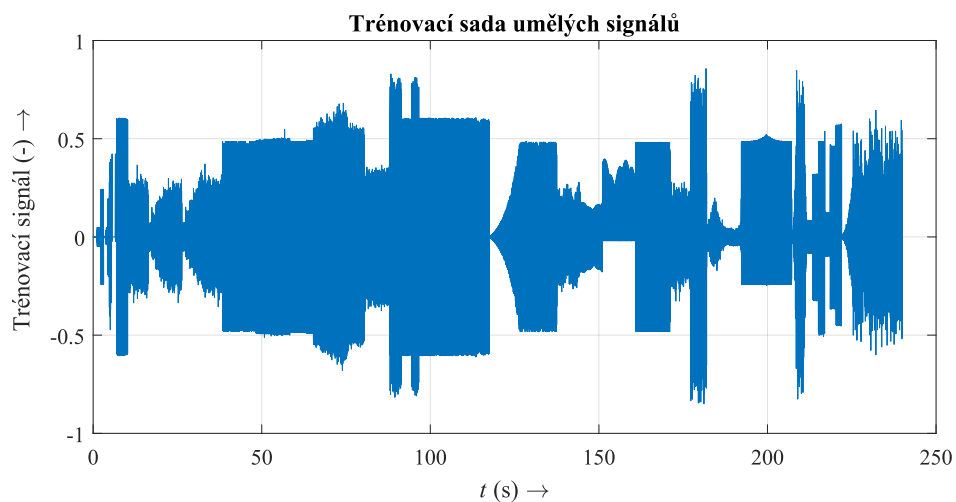
kytary zkreslené nejprve efektem distortion a poté fuzz (MATLAB).

- (3:42,1 – 4:00) Sinusové signály generované s náhodnou délkou, kmitočtem a amplitudou prokládané náhodně dlouhými úseky ticha (MATLAB).

Při prvotním testování byly vykázány nejlepší výsledky při použití sady signálů dostupné ke knihovně NAM, a proto začátek navrženého signálu vychází přímo z této sady. Jedná se konkrétně o tři přeladované sinusové signály, tři šumové signály a dva jednotkové impulsy, které jsou blíže popsány v kapitole 5.1. Po těchto signálech následuje přeladovaný pilový signál v rozsahu kmitočtů od 20 do 720 Hz, který byl vygenerován pomocí tónového generátoru v DAW. Dále následují smíchané sinusové signály o kmitočtech tónů  $e'$  až E,  $c'$  až C a  $g'$  až G, ke kterým je později přimíchán šum. Následující signál je složen z podobných tónů, ale tentokrát zní na počátku současně tóny  $e'$ ,  $c''$  a  $g''$ , ke kterým se postupně přidávají další tóny ve stejném rozpětí jako jsou tóny u předchozího signálu. Opět jsou nejdříve bez šumu a poté s přimíchaným šumem. Následují sinusové signály, které jsou generovány s náhodným kmitočtem, ale stejnou délkou a jsou proloženy pravidelnými segmenty ticha. V DAW byly potom tyto signály časově obráceny. Poté následují pilové signály. Nejdříve modulovaný parametrem *wobble* tónového generátoru, pomocí kterého přechází kmitočet generovaného tónu od nastaveného kmitočtu k cílovému. Následný pilový signál je automatizován tak, aby se náhodně měnil jeho kmitočet v rozsahu 1 Hz až 22 kHz. Po těchto signálech následují opět náhodné sinusové signály s postupným náběhem o délce 8,967 sekund, které jsou zmodulovány pomocí efektu Autofilter v DAW. Dále následují smíchané sinusové signály o kmitočtech tónu standardního ladění šestistrunné kytary a poté o kmitočtech standardního ladění čtyřstrunné baskytary. Tyto signály se zde vyskytují v různých hladinách a s různými typy zkreslení pomocí funkcí popisující měkký ořez signálu, zkreslení pomocí elektronkového zesilovače, distortion nebo fuzz. Během těchto zkreslení nebyl použit antialiasingový filtr, takže je pravděpodobné, že došlo k aliasingu. Později jsou tyto signály také přeladěny o oktávu výše. Sinusové signály o kmitočtech standardního ladění šestistrunné kytary jsou zde také přeladovány o dvě oktávy nahoru, zpět a poté o oktávu dolů. Později jsou také zpracovány efektem kompresor, který má pomocí automatizace měněno nastavení parametrů attack a release. Další z obsažených signálů je osm současně znějících přeladovaných sinusových signálů v rozmezí kmitočtů 20 Hz až 6 kHz, které jsou přeladovány směrem k vyšším kmitočtům a zase zpět. Jejich celková délka je 15 sekund. Po těchto signálech následuje akord, který se zde vyskytuje také v čase 0:10,3 – 0:16,3, ale tentokrát je zkreslený pomocí efektu fuzz, zpracován pomocí efektu tremolo, přeladěn o oktávu výše a má nastavenou dlouhou postupnou dobu náběhu a doběhu. Celá sada je potom zakončena sinusovými signály generovanými s náhodnou délkou, kmitočtem a amplitudou prokládanými různě dlouhými úseky ticha, které byly potom pomocí DAW dvojnásobně zrychleny.

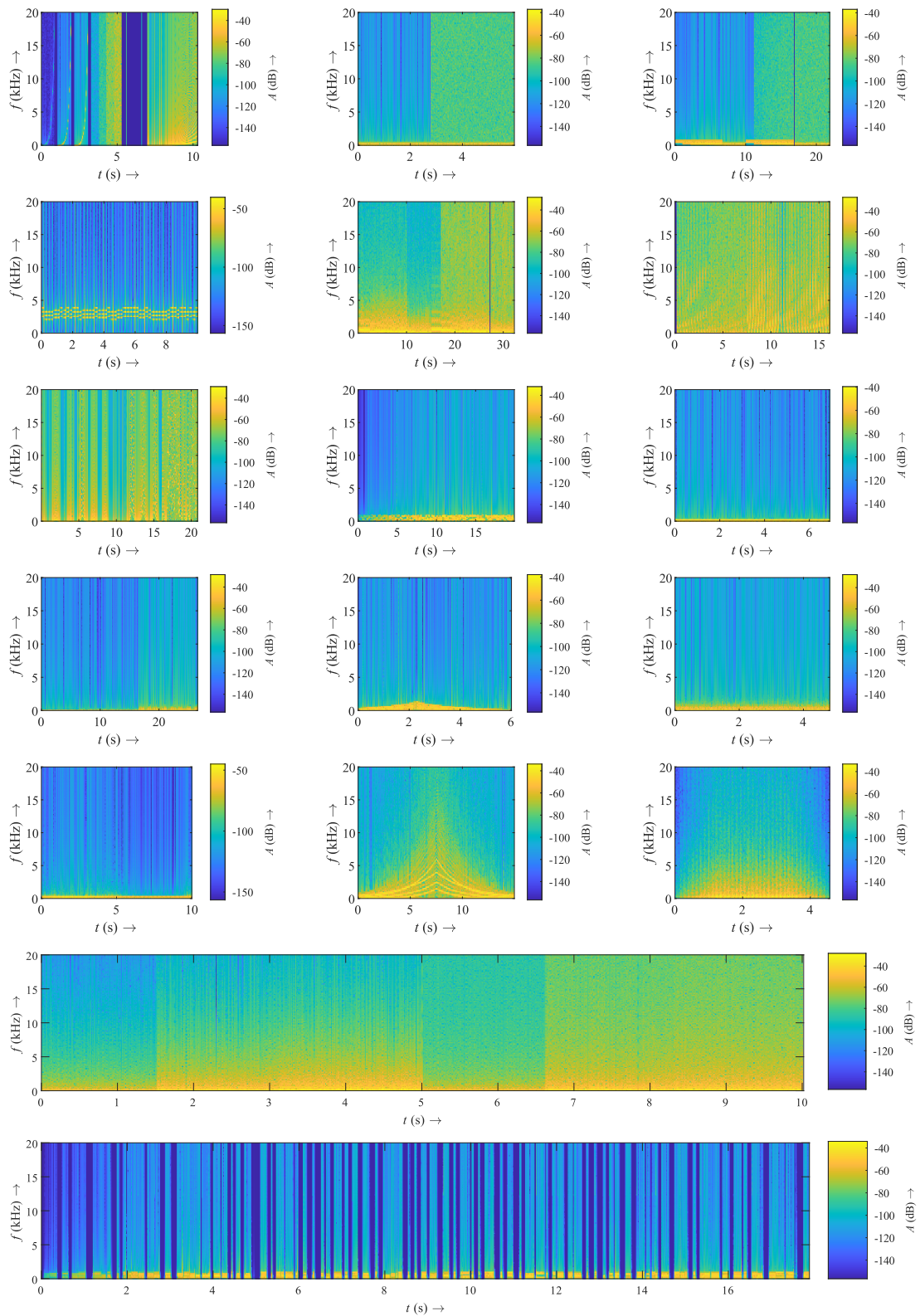
Výpis 5.1: Příklad kódu pro generování náhodných sinusových signálů prokládaných segmenty ticha v prostředí Matlab.

```
1 fs = 44100;
2 numSignals = 100;
3 numf = 10;
4 trainData = [];
5
6 for i = 1:numSignals
7     signalLength = randi([1, fs]);
8     t = (0:signalLength-1)/fs;
9     signal = zeros(1, signalLength);
10
11     for j = 1:numf
12         freq = 100 + (1000-100).*rand(1, 1);
13         amp = rand(1, 1);
14
15         signal = signal + amp * sin(2*pi*freq*t);
16     end
17
18     signal = signal / max(abs(signal));
19     trainData = [trainData; signal'];
20     pauseL = randi([1, fs]);
21     trainData = [trainData; zeros(pauseL,1)];
22 end
23
24 audiowrite('trainData_rand.wav', trainData', fs);
```



Obr. 5.12: Závislost navržené sady signálů na čase.

Spektrogramy sady umělých trénovacích signálů



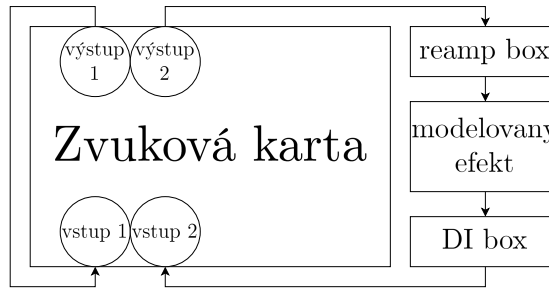
Obr. 5.13: Kmitočtové spektrum navržené sady signálů.

## 6 Výsledky

V této kapitole jsou uvedeny výsledky natrénovaných modelů neuronové sítě při použití kytarového signálu, testovacího signálu ze zařízení Kemper, trénovacích signálů ze zařízení Quad Cortex a knihovny Neural Amp Modeler a sady signálů vytvořené v rámci této práce jako trénovacích množin při učení konkrétní rekurentní neuronové sítě poskytnuté vedoucím práce. Kytarové signály, které byly použity jako vzorová trénovací množina a všechny validační a testovací množiny byly taktéž poskytnuty vedoucím práce. Výsledky jsou rozděleny na objektivní a subjektivní. V případě objektivního hodnocení jsou srovnávány hodnoty ztrátových funkcí a grafy získané při učení sítě pomocí všech dříve zmíněných trénovacích množin. V případě subjektivního hodnocení je potom kladen důraz spíše na lidský sluchový vjem a je tedy proveden poslechový test.

Před modelováním skutečných zvukových zařízení byly nejprve použity již vytvořené simulace v podobě zásuvných modulů (*plug-in*). Konkrétně se jedná o simulaci distortion efektu Big Muff od firmy Electro Harmonix, který je primárně používán pro elektrickou kytaru. Zásuvný modul s názvem Big Stuff byl vytvořen firmou Nembirini audio. Pro generování výstupních signálů tohoto efektu byla využita výše zmíněná knihovna Pedalboard. Jako druhý efekt byla využita simulace elektronkového zesilovače Orange TT15 Tiny Terror s názvem Cypress TT-15, která je vyvinuta firmou Black Rooster Audio. Jako reálná zařízení byl potom použit basový overdrive efekt Boss ODB-3, basový předzesilovač Darkglass Vintage Ultra v2 a kytarový elektronkový zesilovač Marshall JCM900 4100.

Při nahrávání datasetů pro modelování reálných zařízení byla použita zvuková karta Steinberg UR22 mkII. Impedanční přizpůsobení výstupů a vstupů bylo provedeno pomocí reamping boxu [63], který byl zapojen před modelované zařízení pro zvýšení impedance a DI boxu [50], který byl zapojen za modelované zařízení pro opětovné snížení impedance. Pro vyřešení problému s latencí a podobnými problémy se sfázováním vstupních a cílových signálů pro neuronovou síť, byly oba tyto signály nahrávány současně do dvou stop podle zapojení uvedeného na Obr. 6.1. Pro nahrávání byl použit nástroj Pipeline XT, který je dostupný v DAW. Nástroj umožňuje, aby byl hardwarový procesor vložen do efektového řetězce zvukové stopy stejně jako zásuvný modul nebo virtuální efekt. Kalibrace hlasitostí vysílaného a přijímaného signálu byla provedena pomocí tónového generátoru, který byl vložen před nástroj Pipeline XT do efektového řetězce. Pomocí generátoru byl vysílán sinusový signál jehož hlasitost byla na vstupu a výstupu nástroje Pipeline XT srovnána na -20 dB. Poté byl zapnut modelovaný efekt a pomocí jeho ovladačů byla opět srovnána hlasitost na výstupu nástroje Pipeline XT. Byla také využita možnost automatického sfázování signálů, kterou tento nástroj poskytuje.



Obr. 6.1: Schéma zapojení při nahrávání.

Neuronová síť byla potom nastavena na počet vzorků dat použitých pro učení v jedné iteraci pro trénovací množinu  $\text{batch size} = 40$ . Pro validační množinu potom  $\text{batch size} = 12$ . Počet skrytých vrstev byl zvolen 32 a počáteční rychlost učení sítě, která ovlivňuje, jak rychle jsou měněny váhy neuronové sítě během učení byla zvolena 0,004. Při prvotním testování byly při učení použity jak buňky GRU, tak LSTM (viz 4.5), ovšem při použití LSTM byly vykázány lepší výsledky a proto byly při všech učení použity právě tyto buňky.

## 6.1 Objektívni hodnocení výsledků

Jako hodnocení kvality výsledných modelů neuronové sítě jsou použity výsledky ztrátových funkcí a to zejména funkce ESR, která bývá využívána pro hodnocení nelineárních systémů (Kapitola 4.7), a proto jsou v tabulkách tučně vyznačeny hodnoty této funkce získané při použití referenční trénovací sady signálů a sady vytvořené v rámci této práce. Dále jsou pro hodnocení využity časové průběhy výstupních signálů použitého efektu v porovnání s výstupními signály neuronové sítě, absolutní chyba výstupních signálů a srovnání jejich spektrogramů. Grafy jsou získané za pomoci programu MATLAB. Nejdříve jsou zobrazeny již zmíněné časové průběhy, absolutní chyba a nakonec diferenční spektrogramy vytvořené pomocí rozdílu spekter těchto výstupních signálů. Grafy časových průběhů jsou zobrazeny v detailním zobrazení pro lepší čitelnost. Zobrazeno je tedy pouze 50 ms signálů.

### Big Stuff

První z modelovaných efektů je již zmíněná simulace distortion efektu Big Muff s názvem Big Stuff. Náhled na efekt i na jeho simulaci je zobrazen na Obr. 6.2. Big Muff je legendární a velmi populární efekt, který byl na trh uveden v 70. letech. Efekt nabízí širokou škálu nastavení a umožňuje zkreslení signálu od jemného overdrive až po fuzz.

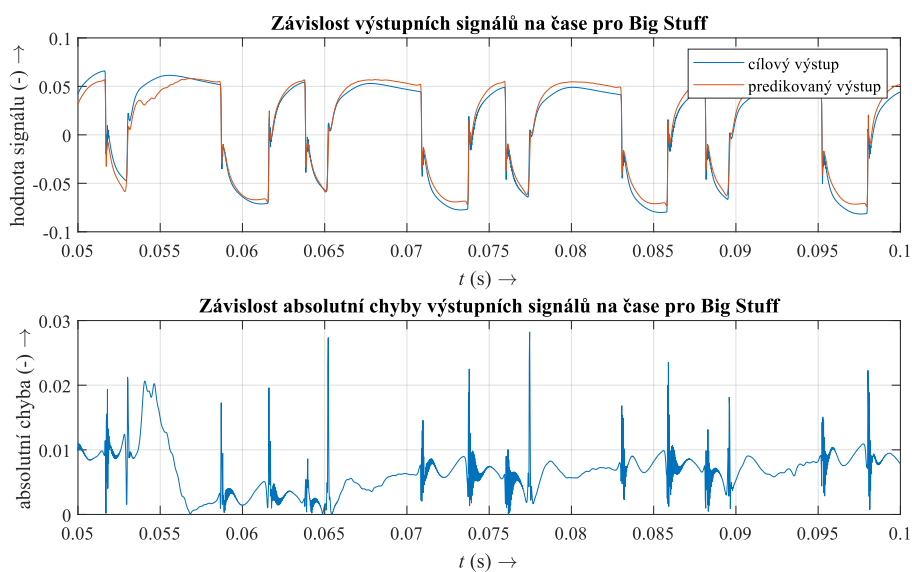


Obr. 6.2: Efekty Big Muff a Big Stuff.

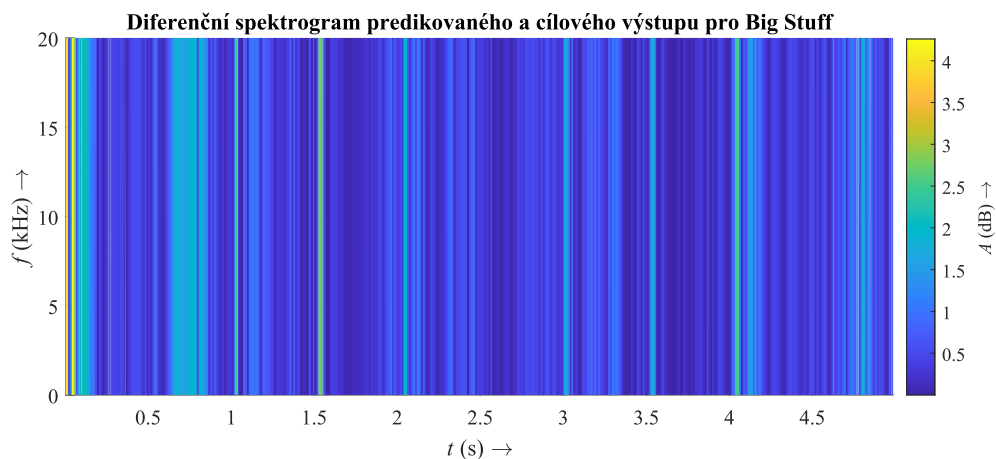
V Tabulce 6.1 jsou uvedeny výsledné hodnoty ztrátových funkcí ESR a MAE pro validační i testovací množinu. Pro hodnocení výsledného modelu je třeba hodnotit především ztrátovou funkcí ESR. Z tabulky vyplývá, že se tato chyba pro navržený trénovací signál liší od ESR pro referenční signál pouze o několik tisícín.

Tab. 6.1: Hodnoty ztrátových funkcí pro Big Stuff.

Big Stuff					
	Kytarový signál	<b>Train 9</b>	Quadcortex	Kemper	NAM
$\mathcal{L}_{\text{ESR}}(\text{val})$	<b>0,00764</b>	<b>0,02016</b>	0,07807	0,07068	0,00924
$\mathcal{L}_{\text{MAE}}(\text{val})$	0,00141	0,00411	0,00824	0,00544	0,00133
$\mathcal{L}_{\text{ESR}}(\text{test})$	<b>0,01330</b>	<b>0,01653</b>	0,08918	0,08999	0,01132
$\mathcal{L}_{\text{ESR}}(\text{test})$	0,00169	0,00372	0,00814	0,00582	0,00168



Obr. 6.3: Časový průběh výstupních signálů a absolutní chyby pro Big Stuff.



Obr. 6.4: Diferenční spektrogram pro Big Stuff.

V grafech časových průběhů výstupních signálů a absolutní chyby (Obr. 6.3) lze potom pozorovat rozdíly v hladinách v časové oblasti signálů a vývoj absolutní chyby v čase. Nakonec je na Obr. 6.4 zobrazen diferenční spektrogram pomocí rozdílu spekter obou signálů. Jsou zde patrné pouze minimální rozdíly, i když jsou pro tento efekt vykázány druhé nejhorší objektivní výsledky ze všech modelovaných efektů.

### Cypress TT-15

Cypress TT-15 je druhý ze simulovaných zásuvných modulů. Jedná se o simulaci elektronkového zesilovače Orange TT15 Tiny Terror. Náhledy na zesilovač a na jeho simulaci jsou zobrazeny na Obr. 6.5. Jedná se o zesilovač s dvěma elektronkami ECC83 (12AX7) v předzesilovači a dvěma elektronkami EL84 pro výkonovou část.

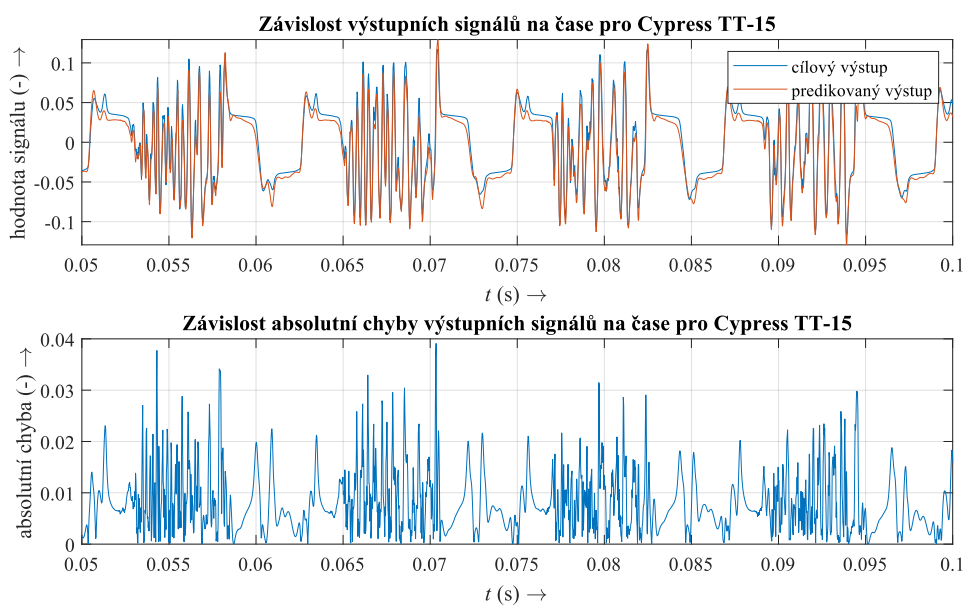


Obr. 6.5: Orange TT15 a Cypress TT-15.

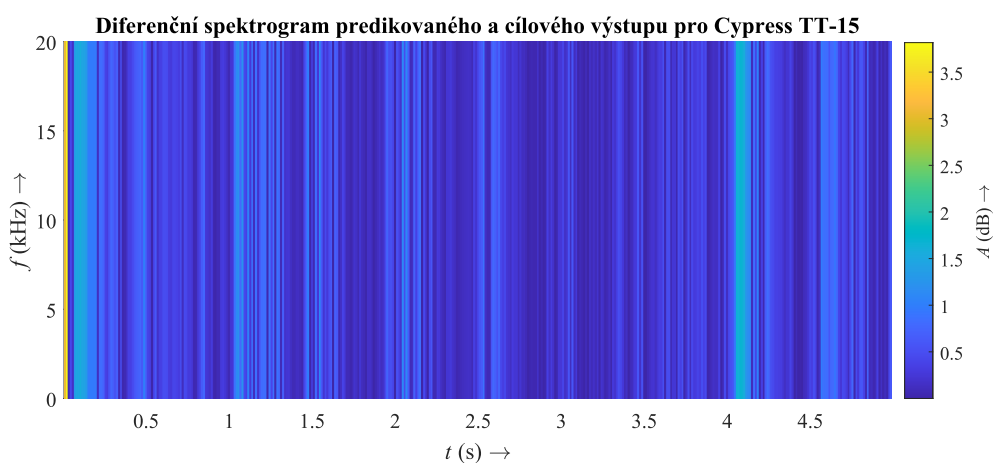
V Tabulce 6.2 lze opět pozorovat rozdíly výsledných hodnot ztrátových funkcí ESR a MAE pro validační i testovací množinu. Z tabulky vyplývá, že se tato chyba pro navržený trénovací signál opět liší od ESR pro referenční signál pouze o několik tisícín, tektokrát je ovšem rozdíl větší než u předchozího efektu.

Tab. 6.2: Hodnoty ztrátových funkcí pro Cypress TT-15.

Cypress TT-15					
	Kytarový signál	<b>Train 9</b>	Quadcortex	Kemper	NAM
$\mathcal{L}_{\text{ESR}}(\text{val})$	<b>0,00474</b>	<b>0,01613</b>	0,05066	0,70160	0,01261
$\mathcal{L}_{\text{MAE}}(\text{val})$	0,00222	0,00355	0,01002	0,04000	0,00433
$\mathcal{L}_{\text{ESR}}(\text{test})$	<b>0,00959</b>	<b>0,01711</b>	0,07052	0,67360	0,01203
$\mathcal{L}_{\text{MAE}}(\text{test})$	0,00314	0,00380	0,01268	0,03997	0,00437



Obr. 6.6: Časový průběh výstupních signálů a absolutní chyby pro Cypress TT-15.



Obr. 6.7: Diferenční spektrogram pro Cypress TT-15.

Na Obr. 6.6 jsou zobrazeny grafech časových průběhů výstupních signálů a absolutní chyby. Opět lze pozorovat rozdíly v hladinách v časové oblasti signálů a vývoj absolutní chyby v čase. Nakonec je na Obr. 6.7 opět zobrazen diferenční spektrogram pomocí rozdílu spekter obou signálů. I u tohoto efektu jsou zde patrné pouze minimální rozdíly. Tentokrát se jedná o nejhůře objektivně hodnocené výsledky ztrátových funkcí ze všech modelovaných efektů. Ve spektrogramu jsou ovšem viditelné menší rozdíly než u předchozího efektu.

### Boss ODB-3

Zde se jedná o pedálový overdrive efekt, který je navržen pro elektrickou baskytaru. Tento efekt je také prvním reálným zařízením modelovaným v rámci této práce. Náhled na efekt je zobrazen na Obr. 6.8.



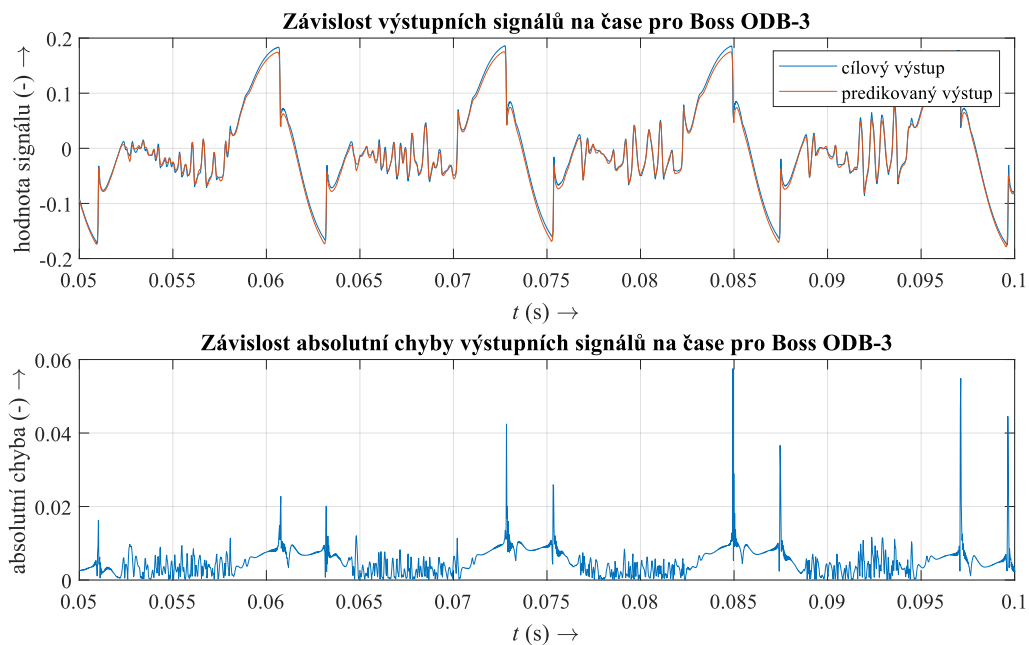
Obr. 6.8: Efekt Boss ODB-3 [62].

Výsledné hodnoty ztrátových funkcí jsou zobrazeny v Tabulce 6.3. Zde jsou výsledné hodnoty ztrátových funkcí podstatně lepší než u přechodících efektů. Chyba ESR při použití trénovacího signálu Train 9, který byl vytvořen v rámci této práce, je pro testovací množinu dokonce nižší než při použití referenční trénovací sady. Pro validační množinu je opět chyba vyšší pouze o několik tisícín. Vzhledem k těmto hodnotám se tento výsledek dá považovat za velmi dobrý.

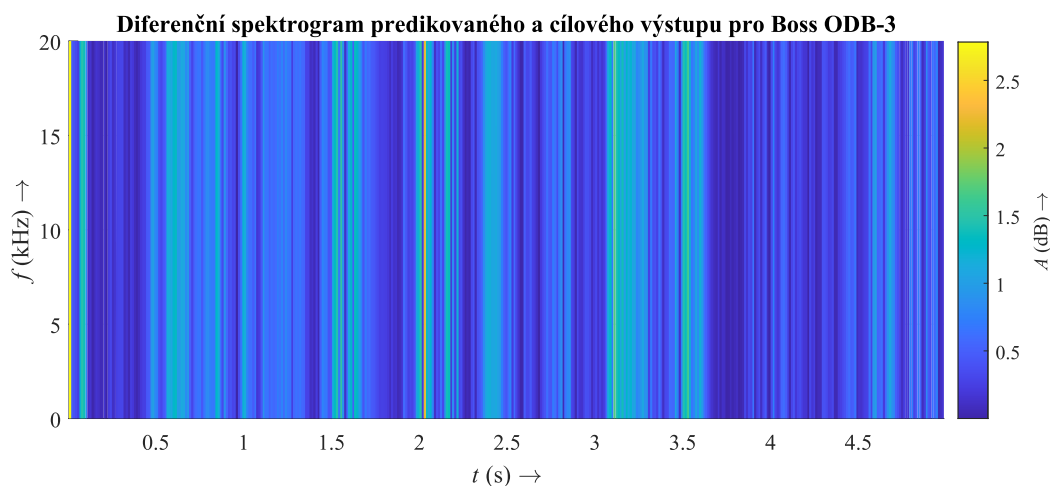
Tab. 6.3: Hodnoty ztrátových funkcí pro Boss ODB-3.

Boss ODB-3					
	Kytarový signál	<b>Train 9</b>	Quadcortex	Kemper	NAM
$\mathcal{L}_{\text{ESR}}(\text{val})$	<b>0,00465</b>	<b>0,00671</b>	0,02660	0,50590	0,00911
$\mathcal{L}_{\text{MAE}}(\text{val})$	0,00224	0,00265	0,00637	0,03149	0,00377
$\mathcal{L}_{\text{ESR}}(\text{test})$	<b>0,00651</b>	<b>0,00470</b>	0,02936	0,42060	0,00984
$\mathcal{L}_{\text{MAE}}(\text{test})$	0,00340	0,00329	0,00934	0,03951	0,00499

Časové průběhy jsou zobrazeny na Obr. 6.9 a spektrogram je zobrazen na Obr. 6.10. Lze zde pozorovat opět mírné rozdíly v časové oblasti signálu, ale nejvyšší hodnoty absolutní chyby. Ve spektrogramu jsou tentokrát patrnější rozdíly než u předchozích efektů.



Obr. 6.9: Časový průběh výstupních signálů a absolutní chyby pro Boss ODB-3.



Obr. 6.10: Diferenční spektrogram pro Boss ODB-3.

## Darkglass Vintage Ultra v2

Basový předzesilovač Darkglass Vintage Ultra v2 je další z modelovaných pedálových

efektů, který nabízí zkreslení signálu typu overdrive až distortion. Efekt disponuje DI výstupem, tudíž nebylo při nahrávání nutné použít DI box, ale rovnou tento výstup pro přivedení výstupního signálu efektu do vstupu zvukové karty. Efekt na tomto výstupu také poskytuje simulaci reproboxu, která byla v průběhu nahrávání vypnutá. Náhled na efekt je zobrazen na Obr 6.11.



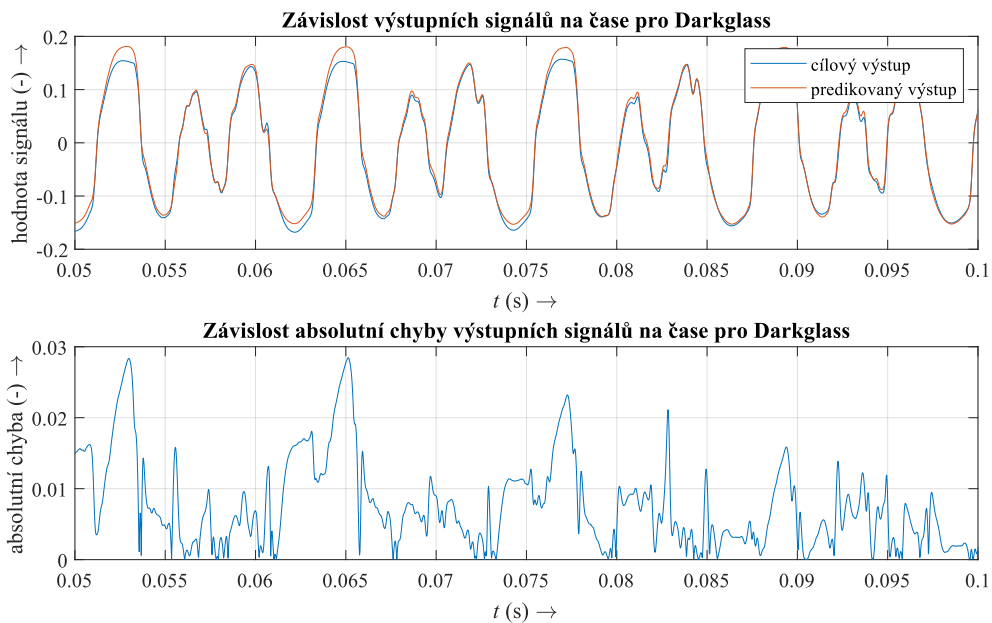
Obr. 6.11: Darkglass Vintage Ultra v2 [62].

Z porovnaných výsledných hodnot ztrátových funkcí v Tabulce 6.4 je patrné, že u tohoto efektu se jedná o nejpřesnější model v kontextu použití trénovacího signálu navrženého v rámci této práce. Výsledná hodnota ztrátové funkce ESR při použití trénovacího signálu Train 9 je zde pro testovací množinu o 0,005 nižší než při použití referenční trénovací sady. Pro validační množinu je tento rozdíl stejný jako u přechodícího efektu a to 0,002.

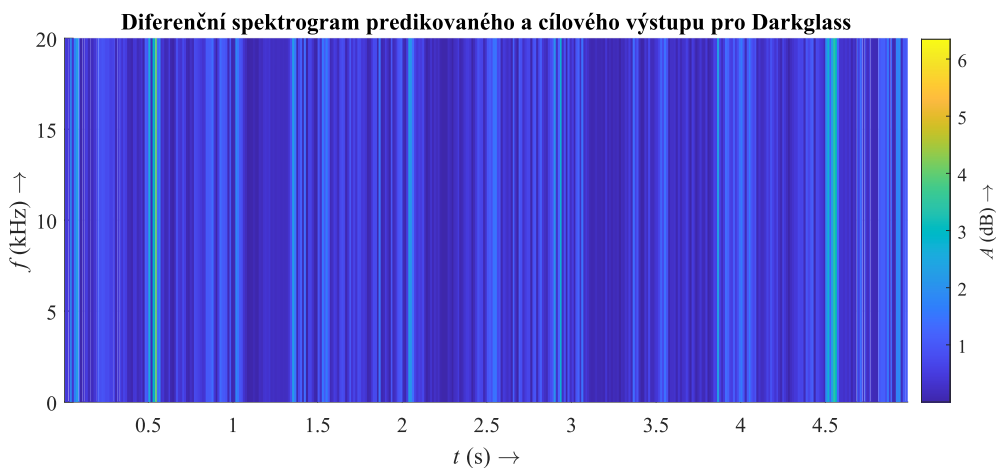
Tab. 6.4: Hodnoty ztrátových funkcí pro Darkglass Vintage Ultra v2.

Darkglass Vintage Ultra v2					
	Kytarový signál	<b>Train 9</b>	Quadcortex	Kemper	NAM
$\mathcal{L}_{\text{ESR}}(\text{val})$	<b>0,00446</b>	<b>0,00636</b>	0,03318	0,65780	0,01052
$\mathcal{L}_{\text{MAE}}(\text{val})$	0,00301	0,00336	0,00970	0,04783	0,00418
$\mathcal{L}_{\text{ESR}}(\text{test})$	<b>0,01110</b>	<b>0,00635</b>	0,07136	0,70430	0,01386
$\mathcal{L}_{\text{MAE}}(\text{test})$	0,00572	0,00409	0,01544	0,05553	0,00606

V grafech časových průběhů (Obr. 6.12) lze pozorovat, že byl efekt modelován při nastavení nižšího zkreslení výstupního signálu. Jedná se tedy o měkký ořez signálu. Rozdíly v časové oblasti jsou zde opět velmi malé. Absolutní chyba vykazuje hodnoty nižší než 0,03. Na spektrogramu (Obr. 6.13) jsou opět zobrazeny nízké hodnoty v rozdílech kmitočtových spekter signálů.



Obr. 6.12: Časový průběh výstupních signálů a absolutní chyby pro Darkglass Vintage Ultra v2.



Obr. 6.13: Diferenční spektrogram pro Darkglass Vintage Ultra v2.

### Marshall JCM900 4100

Poslední z modelovaných efektů je britský elektronkový zesilovač Marshall JCM900 4100. Jedná se o aparát, který má tři elektronky ECC83 v předzesilovači a čtyři elektronky 5881 v koncovém zesilovači. Zesilovač nabízí proslulý zvuk aparátů Marshall z 90. let. Náhled na něj je zobrazen na Obr. 6.14.

V Tabulce 6.5 jsou srovnány výsledné hodnoty ztrátových funkcí pro tento efekt. Rozdíly ztrátových funkcí ESR mezi použitím trénovacích signálů Train 9 a refe-

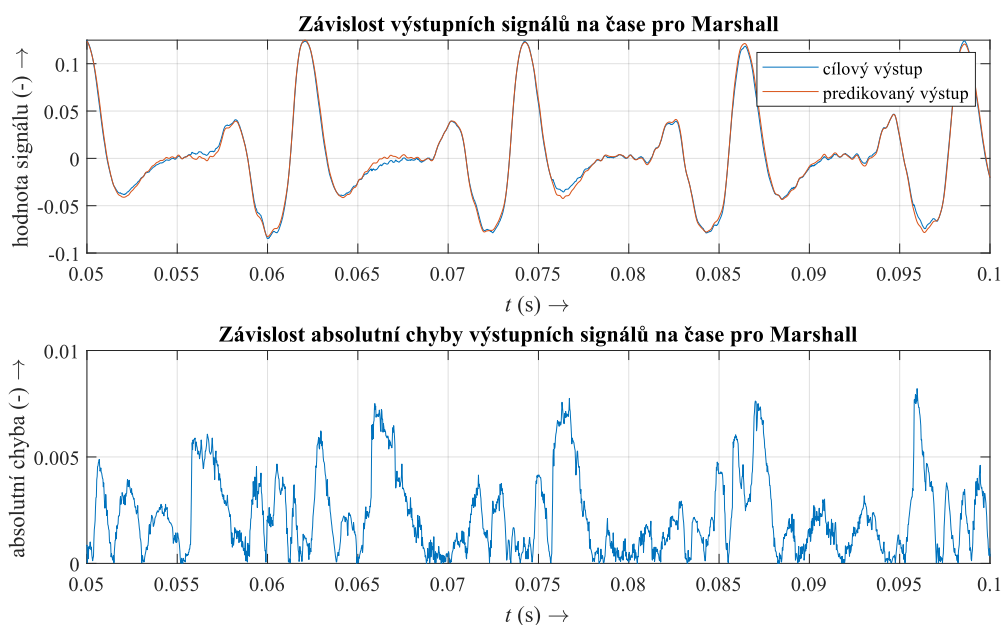


Obr. 6.14: Marshall JCM900 4100 [62].

renční sadou jsou zde pro obě množiny opět velmi nízké a to 0,003 pro testovací a 0,005 pro validační množinu.

Tab. 6.5: Hodnoty ztrátových funkcí pro Marshall JCM900 4100.

Marshall JCM900 4100					
	Kytarový signál	<b>Train 9</b>	Quadcortex	Kemper	NAM
$\mathcal{L}_{\text{ESR}}(\text{val})$	<b>0,00468</b>	<b>0,00977</b>	0,00636	0,10800	0,00608
$\mathcal{L}_{\text{MAE}}(\text{val})$	0,00163	0,00165	0,00207	0,00664	0,00230
$\mathcal{L}_{\text{ESR}}(\text{test})$	<b>0,00423</b>	<b>0,00716</b>	0,00841	0,18100	0,00581
$\mathcal{L}_{\text{MAE}}(\text{test})$	0,00195	0,00191	0,00295	0,01118	0,00276



Obr. 6.15: Časový průběh výstupních signálů a absolutní chyby pro Marshall JCM900 4100.



Obr. 6.16: Diferenční spektrogram pro Marshall JCM900 4100.

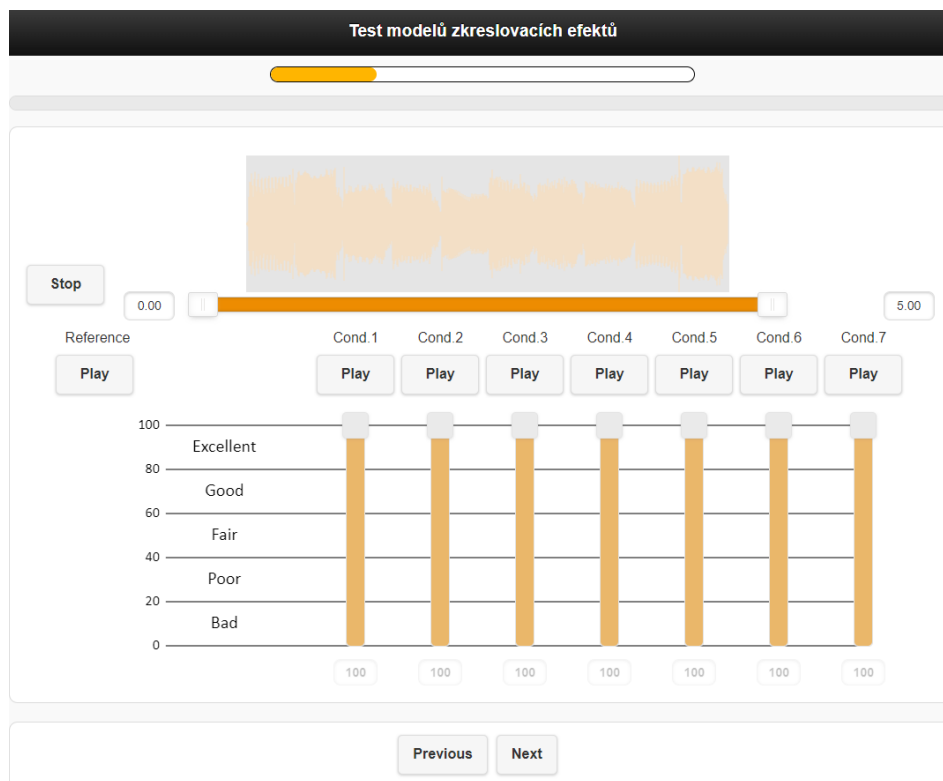
Na grafech časových průběhů výstupních signálů (Obr. 6.15) lze vidět, že efekt byl simulován s ještě menším zkreslením výstupního signálu než efekt minulý. Rozdíly jsou zde téměř nepatrné a absolutní chyba dosahuje hodnot pod 0,01, což je nejmenší hodnota ze všech modelovaných efektů. Spektrogram (Obr. 6.16) ovšem vykazuje velké výkyvy v rozdílech kmitočtových spekter výstupních signálů. Tento jev by mohl být způsoben nekvalitním výsledným modelem, ale vzhledem k ostatním výsledkům by mohl být tento jev způsoben spíše vyšší úrovní šumu nebo jiných parazitních jevů ve výstupním signálu efektu.

## 6.2 Subjektivní hodnocení výsledků

Pro hodnocení výsledků simulace zvukových zařízení je také velmi důležité subjektivní hodnocení výstupních signálů neuronové sítě. V případě zvukových signálů může být totiž výsledek s větší chybou simulace vnímán blíže předloze než ten s chybou menší. Negativně hodnocený výsledek simulace skutečného zařízení může mít tedy alespoň pozitivní výsledek pro sluchový vjem lidského ucha, což může v případě zvukových signálů hrát větší roli. Ideální výsledek modelu zvukového efektu nebo zesilovače by měl být nerozeznatelný od skutečného zařízení.

Pro subjektivní hodnocení výsledných modelů zkreslovacích efektů je využít test pro porovnávání kvality s vícero podněty, skrytou referencí a kotvou – zkráceně MUSHRA. Metoda vykazuje dobré statistické výsledky už při nižším počtu respondentů a to např. od 10 do 20. Principem testu je subjektivní hodnocení výsledků na škále od 0 do 100, kdy 0 znamená špatný výsledek a 100 excelentní. Doporučení pro tento test jsou popsány v normě ITU-R BS.1534-3. Průběh testu spočíval ve srovnávání zvukových souborů získaných učením neuronové sítě pomocí dříve zmiňovaných

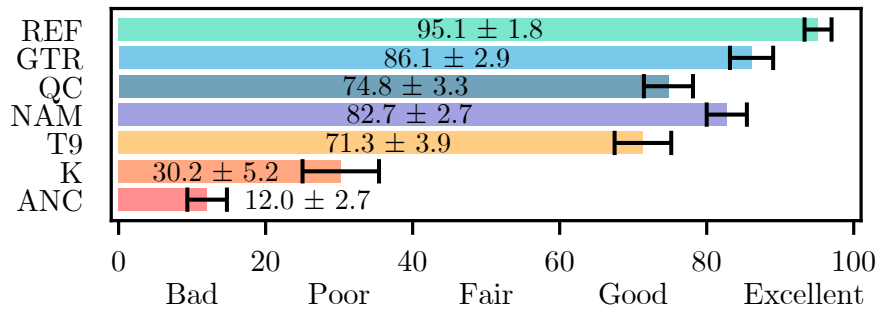
trénovacích signálů s referenční nahrávkou, která byla součástí validační množiny pro učení neuronové sítě. Byly použity dva záznamy hry na elektrickou kytaru a jeden záznam hry na elektrickou baskytaru. Cílem bylo tedy ohodnocení výsledného modelu neuronové sítě trénované pomocí množiny signálů navržené v rámci této práce. Referenční nahrávka byla v průběhu testu stále k dispozici a navíc byla skrytá mezi ostatními hodnocenými stimuly. Byla zde také zahrnuta kotva, tedy znatelně odlišná a nekvalitní nahrávka, u které je očekáváno, že bude hodnocena nejnižšími hodnotami. K testování bylo použito webové prostředí webMUSHRA (viz [64]), které je zobrazeno na Obr. 6.17. Toto prostředí bylo spuštěno pomocí lokálního php serveru. Součástí testu byl návod s požadavky, jako např. udržení konstantní hlasitosti sluchátek v průběhu testu. Na druhé straně testu bylo testovací kolo pro seznámení s prostředím, vyzkoušení hodnocení a nastavení hlasitosti. Jednotlivé stimuly byly v každém kole náhodně promíchány.



Obr. 6.17: Náhled na prostředí webMUSHRA.

Test byl proveden ve zkušebních prostorech pro kapely a to v době, kdy celé prostory nebyly nikým obsazeny pro zamezení zkreslení poslechového testu. Jako přehrávací zařízení byl použit laptop, zvuková karta Steinberg UR22 mkII a sluchátka Beyerdynamic DT 770 PRO s uzavřenou konstrukcí. Respondenti byli složeni ze zvukových inženýrů a muzikantů. Jejich celkový počet byl 13, z čehož byly 4 ženy a 9 mužů. Věkový průměr byl 29 let.

Testování proběhlo pouze pro modely reálných zařízení, tedy pro efekty Boss ODB-3, Darkglass Vintage Ultra v2 a Marshall JCM900 4100. Z výsledků hodnocení respondentů byly vytvořeny průměry a poté byl vykreslen graf, který je zobrazen na Obr. 6.18. Výsledky jsou rozděleny na referenci (REF), kotvu (ANC) a dle použitých trénovacích signálů na kytarový signál (GTR), Quad Cortex (QC), Neural Amp Modeler (NAM), Kemper (K) a trénovací signál Train9 (T9), který byl vytvořen v rámci této práce.



Obr. 6.18: Výsledky poslechového testu MUSHRA.

Výsledek hodnocení reference a kotvy skončil dle očekávání, kdy tyto možnosti byly hodnoceny největšími a nejmenšími hodnotami. Dle očekávání skončil taktéž výsledek GTR, který byl brán jako reference při vytváření trénovacího signálu v rámci této práce. Jeho velmi dobré výsledky byly tedy zopakovány z předešlého objektivního hodnocení. Očekávaný výsledek byl také získán za pomoci trénovacího signálu K, kterým byl testovací signál ze zařízení Kemper. Bylo zde potvrzeno že tento signál není určen pro učení neuronové sítě. Velmi dobré výsledky jsou také vykázány u NAM, kdy byly tyto výsledky vzhledem k objektivnímu hodnocení taktéž očekávány. Výsledky QC jsou zde mírně překvapivé, protože jsou podstatně lepší než je tomu u objektivního hodnocení. Trénovací signál T9, který byl cílem této práce vykazuje v rámci subjektivního hodnocení horší výsledky než v rámci objektivního, což může být v kontextu hodnocení zvukových efektů lidským sluchovým vjemem negativní výsledek. Hodnoty hodnocení jsou pro tento signál ovšem pouze mírně nižší než u ostatních lépe hodnocených výsledků. Např. oproti trénovacímu signálu QC z modelovacího zařízení Quad Cortex je hodnocení nižší pouze o 3,5.

### 6.3 Shrnutí výsledků

Navržená trénovací sada signálů Train 9 vykazuje dle objektivního hodnocení velmi dobré výsledky, kdy jsou hodnoty ztrátových funkcí ESR pro testovací množinu u dvou z reálných efektů dokonce nižší než hodnoty pro referenční sadu signálů.

U jednoho z efektů je pak hodnota vyšší pouze o 0,003. Nejpřesnější z modelů dle objektivních metrik byl natrénován pro efekt Darkglass Vintage Ultra v2. Při modelování již vytvořených simulací efektů byly získány výsledky ztrátových funkcí přibližně o 0,008 vyšší. Tento výsledek je také horší, jelikož bylo u těchto efektů použito silnější zkreslení výstupních signálů.

Negativním výsledkem pro signál navržený v rámci této práce byl ovšem poslechový test MUSHRA, kdy výstupy získané pomocí tohoto signálu byly hodnoceny jako jedny z nejhorších z testovaných. Zde byla potvrzena důležitost subjektivního hodnocení, neboť signál s velmi dobrým objektivním hodnocením byl subjektivně hodnocen hůře. Hodnocení má ovšem hodnotu 71,3 ze 100, což je v rámci MUSHRA testu hodnoceno jako dobrý výsledek. V následující tabulce je uvedeno srovnání ztrátových funkcí ESR a výsledků poslechového testu pro referenční sadu signálů GTR a sadu signálů T9, která byla vytvořena v rámci této práce.

Tab. 6.6: Srovnání výsledných hodnot pro GTR a T9.

	ODB-3	Vintage Ultra v2	JCM900 4100
$\mathcal{L}_{\text{ESR}}(\text{test}) - \text{GTR}$	0,00651	0,0111	0,00423
$\mathcal{L}_{\text{ESR}}(\text{test}) - \text{T9}$	<b>0,00470</b>	<b>0,00635</b>	<b>0,00716</b>
rozdíl $\mathcal{L}_{\text{ESR}}$	0,00181	0,00475	-0,00293
MUSHRA – GTR		86,1	
MUSHRA – T9		<b>71,3</b>	
rozdíl MUSHRA		14,8	

## Závěr

V rámci této diplomové práce byl nejdříve popsán teoretický úvod k signálům, systémům a jejich reprezentaci. Ve druhé kapitole byl potom uveden popis nelineárních zkreslovacích zvukových efektů včetně jejich vlivu na kmitočtové spektrum výstupního signálu. Konkrétně se jedná o efekty overdrive, distortion, fuzz a elektronkový zesilovač. Třetí kapitola pojednává o metodách pro číslicovou simulaci nelineárních efektů. Byl zde popsán větší a vnitřní popis systému, včetně jejich kombinace zvané greybox. V závěru této kapitoly je potom popsána návaznost stavového popisu na rekurentní neuronové sítě. Čtvrtá kapitola je již zaměřena na neuronové sítě. Byl zde popsán jejich historický vývoj a součásti jako je umělý neuron nebo aktivační funkce. Dále jsou zde popsány typy neuronových sítí jako sítě dopředné, konvoluční, temporální konvoluční a sítě rekurentní, kde byla popsána Elmanova rekurentní neuronová síť a výpočetní buňky LSTM a GRU. Kapitola pokračuje popisem optimalizace neuronové sítě, kde je nejdříve popsán algoritmus zpětného šíření chyby a poté optimalizační algoritmy jako gradientní sestup, RMSprop a Adam. V závěru kapitoly jsou popsány ztrátové funkce MAE, MSE, ESR a ztráta STFT.

Pátá kapitola se již věnuje signálům pro identifikaci nelineárních systémů. Nejprve je zde popsán způsob využití přeladovaného sinusového signálu při nelineární konvoluci. Druhá část kapitoly je zaměřena na používané signály pro modelování efektů nebo zesilovačů v komerčních zařízeních jako jsou Quad Cortex od firmy Neural DSP a Profiler Stage od firmy Kemper. Dále je zde popsána knihovna jazyka Python s názvem Neural Amp Modeler, která pro toto modelování používá neuronové sítě, a taktéž nabízí vlastní trénovací sadu signálů. Knihovna je zde uvedena jako volně dostupná alternativa zmíněných komerčních zařízení. Druhá část kapitoly se již dostává k praktické části této práce. Nejprve jsou zde popsány nástroje, které byly použity pro návrh trénovacího signálu, který by měl nahradit trénovací sadu signálů složenou ze záznamů hry na elektrickou kytaru a baskytaru. Při návrhu signálu byl použit program MATLAB, DAW Studio One 5 a pro testování byla použita rekurentní neuronová síť navržená v jazyce Python za pomoci knihovny PyTorch a vývojového prostředí PyCharm.

V Závěrečné kapitole je nejprve popsán postup při nahrávání datasetů pro trénování sítě a poté jsou zde popsány parametry neuronové sítě, která byla použita pro testování trénovacích signálů. Další část je věnována popisu modelovaných efektů a objektivních výsledků modelů získaných při trénování neuronové sítě pomocí referenční sady signálů, trénovacích signálů ze zařízení Quad Cortex a knihovny Neural Amp Modeler, testovacího signálu ze zařízení Kemper a pomoci trénovací sady signálů navržené v rámci této práce. Nejprve byly modelovány simulace distortion efektu Big Muff a elektronkového zesilovače Orange TT15 Tiny Terror. Poté byly

modelovány reálné zkreslovací efekty a to konkrétně basový overdrive efekt Boss ODB-3, basový předzesilovač Darkglass Vintage Ultra v2 a kytarový elektronkový zesilovač Marshall JCM900 4100. Po zhodnocení výsledků dle objektivních metrik bylo provedeno subjektivní hodnocení pomocí poslechového testu MUSHRA a nakonec bylo provedeno shrnutí výsledků, kde bylo uvedeno srovnání výsledných hodnot ztrátových funkcí ESR pro testovací množinu při učení neuronové sítě pomocí referenčního signálu a navrženého signálu Train 9 a výsledky poslechového testu MUSHRA pro tyto signály.

Cílem této práce bylo tedy navržení umělé trénovací sady signálů, která by nahradila záznamy elektrické kytary nebo baskytary jakožto trénovací množiny signálů pro neuronovou síť. Výsledné hodnoty ztrátových funkcí ESR pro testovací množinu jsou pro tento signál vyšší nejvýše o 0,008. Pro efekty Boss ODB-3 a Darkglass Vintage Ultra v2 jsou tyto hodnoty dokonce nižší až o 0,004. Tyto hodnoty se tedy dají považovat za velmi dobrý výsledek. V grafech časových průběhů, kde byly porovnány průběhy predikovaného a cílového výstupu neuronové sítě, lze u všech modelovaných efektů sledovat pouze malé rozdíly a absolutní chyby zde dosahují maximálních hodnot přibližně 0,06. Diferenční spektrogramy těchto výstupů potom zobrazují taktéž nízké hodnoty rozdílů kmitočtových spekter. Poslechový test MUSRA ovšem přinesl pro navržený signál relativně negativní výsledek. Hodnocením 71,3 se stal model vytvořený pomocí navrženého signálu téměř nejhorším. Výsledek je ovšem velmi blízký hodnocení modelu vytvořeného pomocí trénovacího signálu ze zařízení Quad Cortex a pouze o 14,8 nižší než při použití referenčního trénovacího signálu. Výsledek se tedy dá i v tomto případě považovat za dobrý. V této práci bylo tedy dosaženo závěru, že při dalším výzkumu by měla existovat možnost nahradit záznamy hry na elektrickou kytaru a baskytaru umělými signály jakožto trénovací množiny pro učení neuronové sítě určené pro modelování nelineárních zkreslovacích efektů.

# Literatura

- [1] SMĚKAL, Zdeněk. *Analýza signálů a soustav - BPC-ASI: elektronická skripta pro bakalářská studia*. Ústav telekomunikací, FEEC, VUT Brno: 2012. [cit. 2024-04-19].
- [2] HOLČÍK, Jiří, et al. *Signály, časové řady a lineární systémy*. [online]. Multimediální podpora výuky klinických a zdravotnických oborů Portál Lékařské fakulty Masarykovy univerzity. ISSN 1801-6103. Dostupné z: <<https://portal.med.muni.cz/clanek-594-signal-y-casove-rady-a-linearni-systemy.html>>. [cit. 2023-12-18].
- [3] SUCHÁNEK, Tomáš. *Detektor tempa hudebních nahrávek na bázi neuronové sítě*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2021, 87 s. Diplomová práce. Vedoucí práce: Ing. Matěj Ištváněk.
- [4] DIXON, Simon. *Onset Detection Revisited*. [online]. Proc. of the 9th Int. Conference on Digital Audio Effects. 2006. Dostupné z: <<https://www.eecs.qmul.ac.uk/~simond/pub/2006/dafx.pdf>>. [cit. 2020-11-11].
- [5] ZÖLZER, Udo, ed. *DAFX - Digital Audio Effects*. [online]. Chichester, UK: John Wiley, 2011. ISBN 9781119991298. Dostupné z: <<https://onlinelibrary.wiley.com/doi/book/10.1002/9781119991298>>. [cit. 2023-10-17].
- [6] POLÁK, Kamil. *Realizace digitálních audio efektů na signálovém procesoru TMS320C6416*. Liberec, 2011, 84 s. Diplomová práce. Technická univerzita v Liberci, Fakulta mechatroniky, informatiky a mezioborových studií, Ústav informačních technologií a elektroniky. Vedoucí práce: Ing. Zbyněk Koldovský, Ph.D.
- [7] PEŠÁN, Michele. *Audio signal modelling using neural networks*. Brno, 2021, 71 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Štěpán Miklánek.
- [8] BALÍK, Miroslav. *Číslicové zpracování akustických signálů*. Vysoké učení technické v Brně: elektronicky, 2013. [cit. 2023-10-17].
- [9] SMĚKAL, Zdeněk. *Systémy a signály: 1D a 2D diskrétní a číslicové zpracování*. Praha: nakladatelství Sdělovací technika, 2013. ISBN 978-80-86645-23-0 [cit. 2024-05-12].

- [10] KEEN, R. G. *The technology of the fuzz face* [online]. 1998. Dostupné z: <[http://www.geofex.com/article\\_folders/fuzzface/fffram.htm](http://www.geofex.com/article_folders/fuzzface/fffram.htm)>. [cit. 2024-03-31].
- [11] SMRČKA, Jaroslav. *Kytarové zkreslení? Nic jednoduššího!* [online]. In: *Muzikus.cz*, 2003. Dostupné z: <<https://www.casopismuzikus.cz/clanky/kytarove-zkresleni-nic-jednodussiho>>. [cit. 2023-10-17].
- [12] TechieScience Core SME. *Kdy Je Zesilovač Ve Stavu Nasycení Nebo Přerušení? Pochopení Provozních Limitů*. [online]. <https://techiescience.com/cs/>, 13.1.2024. Dostupné z: <<https://shorturl.at/paLsF>>. [cit. 2024-04-30].
- [13] ČÍŽEK, Lukáš. *Kytarový DSP multieffekt*. Plzeň, 2018, 85 s. Diplomová práce. Západočeská univerzita v Plzni, Fakulta elektrotechnická, Katedra aplikované elektroniky a telekomunikací. Vedoucí práce: doc. Ing. Vladimír Pavlíček, Ph.D.
- [14] ALTUNIAN, Gary. *Co je intermodulační zkreslení (IMD)?* [online]. In: *cs.eyewated.com*. Dostupné z: <<https://cs.eyewated.com/co-je-intermodulacni-zkresleni-imd/>>. [cit. 2024-04-30].
- [15] MARTÍNEZ RAMÍREZ, Marco A., Emmanouil BENETOS a Joshua D. REISS. *Deep Learning for Black-Box Modeling of Audio Effects*. *Applied Sciences* [online]. 2020, 10(2). DOI: 10.3390/app10020638. ISSN 2076-3417. Dostupné z: <<https://www.mdpi.com/2076-3417/10/2/638>>. [cit. 2023-10-18].
- [16] PAKARINEN, J. and D. T. Yeh. *A review of digital techniques for modeling vacuum-tube guitar amplifiers*. *Computer Music J.*, vol. 33, no. 2, pp. 85–100, Jun. 2009.
- [17] VRBÍK, Matouš. *Modelování lineárního zkreslení zvukových zařízení*. Brno, 2020, 102 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: doc. Ing. Jiří Schimmel, Ph.D.
- [18] WRIGHT, Alec, Eero-Pekka DAMSKÄGG, Lauri JUVELA a Vesa VÄLIMÄKI. *Real-Time Guitar Amplifier Emulation with Deep Learning*. *Applied Sciences*. [online]. 2020, 10(3). ISSN 2076-3417. Dostupné z: <[doi:10.3390/app10030766](https://doi.org/10.3390/app10030766)>. [cit. 2023-10-18].
- [19] MIŠUREC, Jiří a Zdeněk SMÉKAL. *Číslicové zpracování signálů*. Ústav telekomunikací, FEEC, VUT Brno: 2011. [cit. 2023-10-18].

- [20] YEH, T. David. *Digital Implementation of Musical Distortion Circuits by Analysis and Simulation*. Ph.D. dissertation, Center for Computer Research in Music and Acoustics, Stanford, 2009. [cit. 2024-01-05].
- [21] YEH, T. D, J. S. ABEL and J. O. SMITH. *Automated physical modeling of nonlinear audio circuits for real-time audio effects - part i: Theoretical development*. Audio, Speech, and Language Processing, IEEE Transactions on speech and audio processing, vol. 18, no. 3, March 2010. [cit. 2024-01-05].
- [22] MAČÁK, Jaromír. *Real-time digital simulation of guitar amplifiers as audio effects*. Ph.D. thesis, Brno, 2012. [cit. 2024-01-05].
- [23] EICHAS, Felix, Marco FINK, Martin HOLTERS a Udo ZÖLZER. *Physical Modeling of the MXR Phase 90 Guitar Effect Pedal*. [online]. Proc. of the 17th Int. Conference on Digital Audio Effects (DAFx-14), Erlangen, Germany, September 1-5, 2014. Dostupné z: <[https://dafx.de/paper-archive/2014/dafx14\\_felix\\_eichas\\_physical\\_modeling\\_of\\_the\\_.pdf](https://dafx.de/paper-archive/2014/dafx14_felix_eichas_physical_modeling_of_the_.pdf)>. [cit. 2024-01-05].
- [24] EICHAS, Felix, Stephan MÖLLER a Udo ZÖLZER. *Block-oriented Gray Box Modeling of Guitar Amplifiers*. [online]. Proc. of the 20th Int. Conference on Digital Audio Effects (DAFx-17), Edinburgh, UK , September 5-9, 2017. Dostupné z: <[https://www.dafx17.eca.ed.ac.uk/papers/DAFx17\\_paper\\_35.pdf](https://www.dafx17.eca.ed.ac.uk/papers/DAFx17_paper_35.pdf)>. [cit. 2024-01-03].
- [25] FETTWEIS, Alfred. *Wave digital filters: Theory and practice*. [online]. Proceedings of the IEEE, 1986, 74.2: 270-327. Dostupné z: <<https://ieeexplore.ieee.org/abstract/document/1457726>>. [cit. 2024-03-31].
- [26] PARKER, D. Julian, Fabián ESQUEDA a André BERGNER. *Modelling of nonlinear state-space systems using a deep neural network*. [online]. Proc. of the 22th Int. Conference on Digital Audio Effects (DAFx-19), Birmingham, UK , September 2-6, 2019. Dostupné z: <[https://www.dafx.de/paper-archive/2019/DAFx2019\\_paper\\_42.pdf](https://www.dafx.de/paper-archive/2019/DAFx2019_paper_42.pdf)>. [cit. 2024-01-03].
- [27] VALOVÁ, A. *Realizace počítačové podpory symbolických modelů v Matlabu*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2015. 31 s. Vedoucí prof. RNDr. Ing. Tomáš Březina, CSc.
- [28] JANČÍKOVÁ, Zora. *Teorie systémů*. 2012. [cit. 2023-10-18].
- [29] PACAL, Tomáš. *Redukce šumu v audiosignálu pomocí deep learning*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2023, 67 s. Bakalářská práce. Vedoucí práce: Ing. Ondřej Mokřý.

- [30] KRAMÁŘ, Denis. *Analýza zvukových nahrávek pomocí hlubokého učení*. Brno, 2021, 64 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Jiří Přinosil, Ph.D.
- [31] BLÁHA, Milan. *Matematická biologie učebnice: Koncept umělé neuronové sítě*. [online]. 2003 Dostupné z: <<https://rb.gy/uagup4>>. [cit. 5. 11. 2022].
- [32] KULHÁNEK, Petr. *Fyzika II: Studijní text*. [online]. 1. vydání. Praha: AGA, 2021. Kapitola 1. Dostupné z: <<https://www.aldebaran.cz/studium/f2.pdf>>. [cit. 2024-05-01].
- [33] *CS231n: Convolutional Neural Networks for Visual Recognition*. [online]. Stanford: Stanford University, 2020. Dostupné z: <<https://cs231n.github.io/neural-networks-1/>>. [cit. 2024-01-07].
- [34] HE, Kaiming, Xiangyu ZHANG, Shaoqing REN a Jian SUN. *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. [online]. Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1026-1034. Dostupné z: <<https://arxiv.org/pdf/1502.01852.pdf>>. [cit. 2024-01-07].
- [35] LEMAIRE, Quentin and André HOLZAPFEL. *Temporal Convolutional Networks for Speech and Music Detection in Radio Broadcast*. [online]. International Society for Music Information Retrieval Conference, 2019. Dostupné z: <<https://archives.ismir.net/ismir2019/paper/000026.pdf>>. [cit. 2024-01-14].
- [36] LECUN, Yann, Yoshua BENGIO a Geoffrey HINTON. *Deep learning*. [online]. Nature. 2015, 521(7553), 436-444. ISSN 0028-0836 Dostupné z: <<https://www.nature.com/articles/nature14539>>. [cit. 2024-01-28].
- [37] MEHLIG, Bernhard. *Machine learning with neural networks: an introduction for scientists and engineers*. [online]. Cambridge University Press, 2021. Dostupné z: <<https://arxiv.org/abs/1901.05639>>. [cit. 2024-01-28].
- [38] WRIGHT, A.; DAMSKÄGG, E.-P.; VÄLIMÄKI, V. *Real-Time Black-Box Modelling With Recurrent Neural Networks*. [online]. Proc. of the 22th Int. Conference on Digital Audio Effects (DAFx-19), Birmingham, UK , September 2-6, 2019. Dostupné z: <[https://dafx.de/paper-archive/2019/DAFx2019\\_paper\\_43.pdf](https://dafx.de/paper-archive/2019/DAFx2019_paper_43.pdf)>. [cit. 2024-01-28].

- [39] MONTAÑO MORENO, J. Juan, Alfonso PALMER POL, Pilar MUÑOZ GRACIA. *Artificial neural networks applied to forecasting time series*. [online]. *Psicothema*. 2011 Apr;23(2):322-9. PMID: 21504688. Dostupné z: <<https://www.psicothema.com/pdf/3889.pdf>>. [cit. 2024-02-07].
- [40] HOCHREITER, S.; SCHMIDHUBER, J. *Long Short-Term Memory*. *Neural Computation*. [online]. IEEE, 1997, 2018, 9(8), 1735-1780. ISBN 978-1-5386-2238-4. ISSN 0899-7667. Dostupné z: <<https://www.bioinf.jku.at/publications/older/2604.pdf>>. [cit. 2024-02-21].
- [41] STEINMETZ, C. J.; REISS, J. D. *auraloss: Audio focused loss functions in PyTorch*. [online]. In Digital Music Research Network One-day Workshop (DMRN+15), 2020. Dostupné z: <[https://eecs.qmul.ac.uk/~josh/documents/2020/DMRN15\\_auraloss\\_Audio\\_focused\\_loss\\_functions\\_in\\_PyTorch.pdf](https://eecs.qmul.ac.uk/~josh/documents/2020/DMRN15_auraloss_Audio_focused_loss_functions_in_PyTorch.pdf)>.
- [42] ZHANG, Zhilu, Mert SABUNC. *Generalized cross entropy loss for training deep neural networks with noisy labels*. [online]. *Advances in neural information processing systems*, 2018, 31. Dostupné z: <<https://arxiv.org/abs/1805.07836>>. [cit. 2024-03-04].
- [43] GUPTA, Sparsh, Brennan WHITFIELD. *The 7 most common machine learning loss functions*. [online]. <https://builtin.com>, 2023. Dostupné z: <<https://builtin.com/machine-learning/common-loss-functions>>. [cit. 2024-03-24].
- [44] NOVAK, Antonin, Laurent SIMON, Pierrick LOTTON. *Synchronized Swept-Sine: Theory, Application, and Implementation*. [online]. *Journal of the Audio Engineering Society*, 2015, 63 (10), pp.786-798. Dostupné z: <<https://hal.science/hal-02504321>>. [cit. 2024-03-10].
- [45] FARINA, Angelo. *Simultaneous measurement of impulse response and distortion with a swept-sine technique*. [online]. *Journal of The Audio Engineering Society*, únor 2000. Dostupné z: <[https://www.researchgate.net/publication/277293870\\_Simultaneous\\_measurement\\_of\\_impulse\\_response\\_and\\_distortion\\_with\\_a\\_swept-sine\\_technique\\_Audio\\_Engineering\\_Society\\_Convention\\_108\\_Audio\\_Engineering\\_Society](https://www.researchgate.net/publication/277293870_Simultaneous_measurement_of_impulse_response_and_distortion_with_a_swept-sine_technique_Audio_Engineering_Society_Convention_108_Audio_Engineering_Society)>. [cit. 2024-03-10].
- [46] NOVAK, Antonin, Laurent SIMON, Pierrick LOTTON. *Nonlinear System Identification Using Exponential Swept-Sine Signal*. [online]. *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 8, pp. 2220-2229, Aug.

2010. Dostupné z: <<https://ieeexplore.ieee.org/document/5299278>>. [cit. 2024-03-10].
- [47] Neural DSP. *Quad Cortex User Manual: CorOS version 2.3.0*. [online]. Version 2.3.0. 2024. Dostupné z: <<https://support.neuraldsp.com/help/quad-cortex-user-manual>>. [cit. 2024-03-04].
- [48] *Kytarové zesilovače, efekty a procesory* [online]. In: Kytary.cz. Dostupné z: <<https://kytary.cz/>>. [cit. 2024-03-18].
- [49] *Electric Guitar, Amplifier Box, Microphone, Sound Box, Headphones* [online]. In: flaticon.com. Dostupné z: <<https://www.flaticon.com/>>. [cit. 2024-05-16].
- [50] ŠVAMBERK, Alex. *Budujeme studio XLVII: chvála DI boxu* [online]. In: Frontman.cz 2017. Dostupné z: <<https://www.frontman.cz/budujeme-studio-ylvii-chvala-di-boxu>>. [cit. 2024-05-17].
- [51] Kemper. *Profiler Stage* [online]. Dostupné z: <<https://www.kemper-amps.com/products/cz/profiler-stage.html>>. [cit. 2024-03-04].
- [52] Kemper, C. *Musical instrument with acoustic transducer* [online]. 2008, US Patent App. 11/881,818. Dostupné z: <<https://www.google.com/patents/US20080134867>>. [cit. 2024-05-16].
- [53] EICHAS, Felix a Udo ZÖLZER. *Virtual analog modeling of guitar amplifiers with Wiener-Hammerstein models* [online]. In: 44th Annual Convention on Acoustics (DAGA 2018). 2018. Dostupné z: <<https://www.google.com/patents/US20080134867>>. [cit. 2024-05-16].
- [54] PASZKE, A.; GROSS, S.; MASSA, F.; et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. [online]. In: Advances in Neural Information Processing Systems 32, editácia H. Wallach; H. Larochelle; A. Beygelzimer; F. d'Alché-Buc; E. Fox; R. Garnett, Curran Associates, Inc., 2019, s. 8024-8035. Dostupné z: <<https://arxiv.org/abs/1912.01703>>. [cit. 2024-03-18].
- [55] Tvůrci PyTorch Lightning. *Welcome to PyTorch Lightning* [online]. Dostupné z: <<https://lightning.ai/docs/pytorch/stable/>>. [cit. 2024-03-18].
- [56] PyTorch. *How to use TensorBoard with PyTorch* [online]. 2024 Dostupné z: <[https://pytorch.org/tutorials/recipes/recipes/tensorboard\\_with\\_pytorch.html](https://pytorch.org/tutorials/recipes/recipes/tensorboard_with_pytorch.html)>. [cit. 2024-03-18].

- [57] BECHTOLD, Bastian, Matthias GEIER. *python-soundfile* [online]. Revision 7deb23a0, 2015. Dostupné z: <<https://python-soundfile.readthedocs.io/en/0.11.0/>>. [cit. 2024-03-18].
- [58] Weights & Biases. *Weights & Biases Documentation* [online]. Dostupné z: <<https://docs.wandb.ai/>>. [cit. 2024-03-18].
- [59] KINGMA, Diederik P. a Jimmy Lei BA. *Adam: A Method for Stochastic Optimization*. [online]. IPublished as a conference paper at ICLR 2015. 2015, 2015, 1-15. Dostupné z: <<https://arxiv.org/pdf/1412.6980.pdf>>. [cit. 2024-03-18].
- [60] BOTTOU, Léon a Olivier BOUSQUET. *The Tradeoffs of Large Scale Learning*. In Sra, Suvrit; Nowozin, Sebastian; Wright, Stephen J. (eds.). *Optimization for Machine Learning*. Cambridge: MIT Press. pp. 351–368. ISBN 978-0-262-01646-9. 2012. [cit. 2024-03-18].
- [61] SPOTIFY. *Pedalboard: A Python library for adding effects to audio*. [online]. Dostupné z: <<https://pypi.org/project/pedalboard/>>. [cit. 2024-05-04].
- [62] *Boss ODB-3, Darkglass Vintage Ultra v2 a Marshall JCM900 4100* [online]. In: gear4music.cz. Dostupné z: <<https://www.gear4music.cz/cs/>>. [cit. 2024-05-10].
- [63] *PALMER DACCAPO* [online]. In: Kytary.cz. Dostupné z: <<https://kytary.cz/palmer-daccapo/HN155694/>>. [cit. 2024-05-17].
- [64] Schoeffler, M. e. a. *webMUSHRA Ů A Comprehensive Framework for Webbased Listening Tests* [online]. *Journal of Open Research Software*, 2018: str. 8. Dostupné z: <<https://openresearchsoftware.metajnl.com/articles/10.5334/jors.187>>. [cit. 2024-05-10].

## Seznam zkratk

<b>2D</b>	Dvourozměrný – Two-Dimensional
<b>A/D</b>	Analogově číslicový převod – Analog to Digital conversion
<b>Adam</b>	Adaptivní odhad momentu – Adaptive Moment Estimation
<b>ANC</b>	Kotva použitá při poslechovém testu
<b>ANN</b>	Umělé neuronové sítě – Artificial Neural Networks
<b>BPTT</b>	Algoritmus zpětné propagace – Backpropagation Through Time
<b>CNN</b>	Konvoluční neuronové sítě – Convolutional Neural Networks
<b>DAW</b>	Virtuální prostředí digitálního zvukového studia – Digital Audio Workstation
<b>DI</b>	Přímý vstup – Direct Input
<b>DSP</b>	Digitální Signálový Procesor – Digital Signal Processor
<b>ESR</b>	Typ ztrátová funkce – Error-to-Signal Ratio
<b>GPU</b>	Grafický procesor – Graphics Processing Unit
<b>GRU</b>	Typ výpočetní buňky – Gated Recurrent Unit
<b>GTR</b>	Kytarový trénovací signál
<b>K</b>	Kemper
<b>LSTM</b>	Typ výpočetní buňky – Long Short-Term Memory
<b>LTI</b>	Lineární časově invariantní – Linear Time-Invariant
<b>MAE</b>	Střední absolutní chyba – Mean Absolute Error
<b>MIMO</b>	Obecný systém s několika vstupy a několika výstupy – Multiple-Inputs Multiple-Outputs
<b>MLP</b>	Vícevrstvý peceptron – Multilayer Perceptron
<b>MSE</b>	Střední kvadratická chyba – Mean Square Error
<b>MUSHRA</b>	Test s vícero podněty, skrytou referencí a kotvou – Multiple Stimulus test with Hidden Reference and Anchor

<b>NAM</b>	Knihovna jazyka Python – Neural Amp Modeler
<b>NN</b>	Neuronové sítě – Neural Networks
<b>QC</b>	Quad Cortex
<b>REF</b>	Reference použitá při poslechovém testu
<b>ReLU</b>	Typ aktivační funkce – Rectified Linear Unit
<b>RMSProp</b>	Root Mean Squared Propagation
<b>RNN</b>	Rekurentní neuronová síť – Recurrent Neural Network
<b>SDG</b>	Stochastický gradientní sestup – Stochastic Gradient Descent
<b>STFT</b>	Krátkodobá Fourierova transformace – Short-Time Fourier Transform
<b>T9</b>	Navržený trénovací signál
<b>TCN</b>	Temporální konvoluční sítě – Temporal Convolutional Networks
<b>WDF</b>	Vlnový digitální filtr – Wave Digital Filter

## Seznam symbolů a veličin

$A, B, C, D$	matice stavového popisu
$b$	práh – bias
$c$	vektor vnitřních stavů
$d$	dilatační faktor
$f(\cdot)$	funkce
$f$	kmitočet
$f_{\max}$	maximální kmitočet
$f_{vz}$	vzorkovací kmitočet
$h$	vektor skrytých stavů
$N$	počet, délka okna
$n$	index diskrétního času, počet kvantovacích hladin
$q$	kvantovací krok
$t$	čas
$u$	vstupní vektor
$U_{\max}$	maximální hodnota napětí
$W$	váha
$x$	stavový vektor
$y, y$	výstup, výstupní vektor

## A Obsah elektronické přílohy

Elektronická příloha obsahuje zdrojové kódy pro generování, testování a hodnocení trénovacích signálů. Adresář *matlab* obsahuje skripty programu MATLAB pro generování trénovacích signálů. Adresář *mushra* obsahuje zdrojové kódy pro poslechový test, včetně použitých zvukových souborů a zdrojového kódu pro vytvoření grafu výsledných hodnot testu. V adresáři *nonlinear-modeling* jsou popsány pouze nejvyužívanější soubory pro tuto práci. Adresář kromě zdrojových kódů neuronové sítě obsahuje také použité datasety při trénování sítě. Pro implementaci neuronové sítě byla použit jazyk Python verze 3.9.18. Seznam potřebných knihoven pro spuštění je uveden v textovém souboru *requirements.txt*. Vzhledem k omezené velikosti přílohy je do archivu vložen textový soubor s odkazem na elektronické uložení, kam byly tyto soubory nahrány.

/ .....	Kořenový adresář přiloženého archivu
├─ matlab.....	Zdrojové kódy pro generování trénovacích signálů
├─ mushra.....	Soubory použité pro poslechový test MUSHRA
├─ nonlinear-modeling .....	Adresář obsahující zdrojové kódy neuronové sítě
│   └─ dataset.....	Adresář obsahující datasety pro trénování sítě
│       └─ data_loader.py .....	Načtení datasetu
│       └─ loss_funcs.py.....	Ztrátové funkce
│       └─ requirements.txt .....	Seznam potřebných knihoven pro spuštění
│       └─ train.py.....	Trénování sítě
├─ odkazy.txt.....	Textový dokument s odkazy na elektronické uložení
├─ Train_9.wav.....	Navržený trénovací signál
└─ Diplomova_prace_Kuba.pdf.....	Elektronická verze diplomové práce