

České vysoké učení technické v Praze

Fakulta elektrotechnická

Katedra radioelektroniky



Diplomová práce

Audio testy jednotek infotainmentu

Jan Dvořák

Studijní program: Elektronika a komunikace

Studijní obor: Audiovizuální technika a zpracování signálu

Vedoucí práce: Ing. František Rund, Ph.D.

Praha

2019

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací

V Praze dne: _____

Poděkování

Tímto bych chtěl poděkovat vedoucímu práce panu Ing. Františku Rundovi, Ph.D. za trpělivost odborné rady a vedení v průběhu přípravy této práce. Také bych chtěl poděkovat kolegů Ing. Matúši Čarnému a Ing. Tomáši Lustykovi, Ph.D. za odborné konzultace v oblasti testování infotainmentu.

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Dvořák** Jméno: **Jan** Osobní číslo: **411198**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra radioelektroniky**
Studijní program: **Elektronika a komunikace**
Studijní obor: **Audiovizuální technika a zpracování signálů**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Audio testy jednotek infotainmentu

Název diplomové práce anglicky:

Audio Testing of Infotainment Units

Pokyny pro vypracování:

Seznamte se s problematikou vývoje testování jednotek infotainmentu v automobilovém průmyslu. Zaměřte se na hodnocení zvukového výstupu jednotky. Studujte typy zvukových artefaktů ovlivňujících testování a navrhnete metody jejich detekce. Dále se zaměřte na určování změn ve frekvenční charakteristice výstupu jednotky (v závislosti na nastavení ekvalizéru). Implementujte své řešení pro použití při testování.

Seznam doporučené literatury:

- [1] Huang, Y., et al.: Development of an automated testing system for vehicle infotainment system. Int. J. Adv. Manuf. Technol. vol. 51. pp. 233 - 246. Nov. 2010. doi: 10.1007/s00170-010-2626-2
[2] Kadlec, F.: Zpracování akustických signálů. Skripta. Praha: ČVUT, FEL. 2005.

Jméno a pracoviště vedoucí(ho) diplomové práce:

Ing. František Rund, Ph.D., katedra radioelektroniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **16.02.2018**

Termín odevzdání diplomové práce: **24.05.2019**

Platnost zadání diplomové práce: **30.09.2019**

Ing. František Rund, Ph.D.
podpis vedoucí(ho) práce

prof. Mgr. Petr Páta, Ph.D.
podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

Datum převzetí zadání

Podpis studenta

Abstrakt

Tato práce je zaměřena na metody zpracování zvuku v průběhu automatizovaného testování jednotek infotainmentu. Konkrétně na studium a detekci rušivých artefaktů ve zvukovém signálu, určování frekvenční odezvy systému v závislosti na nastavení ekvalizéru jednotky a kontrolu aktivních zdrojů. Výstupem této práce je aplikace umožňující detekci artefaktů v testovacím zvukovém signálu, kontrolu aktivních zdrojů a určení frekvenční odezvy infotainmentových jednotek. Vytvořené algoritmy byly ověřeny jako vhodné pro použití testováním na reálných jednotkách výrobce Škoda Auto a. s.

Klíčová slova

Infotainment, Testování softwaru, Automatizace testování, zpracování zvuku, detekce artefaktů, frekvenční odezva, MLS, Sine sweep, Spektrální odečítání

Abstract

This paper is focused on audio signal processing methods in automated infotainment unit testing. Specifically, on study and detection audio artefacts, determination of frequency response of the system in dependence of the unit's equalizer settings and checking of active sources. The outcome of this paper an application allowing the user to detect audio artifact in test signal, check active sources and determinate frequency response of infotainment units. Function of the algorithms was assessed as usable for testing on real infotainment units of Škoda auto a. s. manufacturer.

Key words

Infotainment, Software testing, Test automation, sound signal processing, artifact detection, frequency response, MLS, Sine sweep, Spectral subtraction

Obsah

Úvod.....	11
1 Infotainment.....	13
1.1 Testovaná jednotka Infotainmentu.....	15
2 Testování infotainmentu.....	19
2.1 Testování softwaru.....	19
2.2 Proces testování.....	19
2.3 Techniky vývoje testů.....	22
2.4 Úrovně testování.....	25
2.5 Forma testu.....	26
3 Popis testovacího prostředí.....	31
3.1 Popis pracoviště.....	31
3.2 Testovací stav.....	32
4 Studium zpracování zvuku při testech.....	41
4.1 Typ testů.....	41
4.2 Sledované parametry.....	42
4.3 Navrhovaná zlepšení.....	44
4.4 Studium artefaktů.....	45
5 Relevantní metody zpracování signálu.....	51
5.1 Detekce artefaktů.....	51
5.2 Frekvenční odezva systému.....	53
6 Implementace algoritmů a testy.....	61
6.1 Vývojové prostředí.....	61
6.2 Použitá data.....	62
6.3 Vizuální kontrola záznamu testu.....	62
6.4 Kontrola zdrojů.....	65

6.5	Detekce artefaktů	68
6.6	Frekvenční odezva	71
6.7	Popis aplikace	79
7	Závěr.....	85
8	Zdroje.....	87
	Obsah datové přílohy	91

Seznam obrázků

Obrázek 1.1 - Displej infotainmentu vozu Škoda Kodiaq (zdroj: skoda-auto.cz.....	13
Obrázek 1.2 - Digitální přístrojový štít Audi (Zdroj: audi.cz).....	14
Obrázek 1.3 Jednotka Škoda Swing ve voze Škoda Kodiaq (zdroj: skoda-auto.cz)	15
Obrázek 2.1 - Testovací proces ISTQB (zdroj:fas-cosulting.de).....	20
Obrázek 2.2– Porovnání technik černé a bílé skříňky, (zdrojsoftwaretestinggenius.com:)	22
Obrázek 2.3 - V- model vývojového procesu, (zdroj: fas-consulting.de).....	25
Obrázek 2.4– Skladba „test casů“, (zdroj: fas-consulting.de).....	27
Obrázek 3.1- prostředí laboratoře komponentního testování (digiteqautomotive.com)	31
Obrázek 3.2 - ilustrační foto testovacího stavu. (zdroj: digiteqautomotive.com)	32
Obrázek 3.3 - blokové schéma testovacího stavu	33
Obrázek 3.4- Prostředí Vision Builder AI (zdroj:ni.com).....	34
Obrázek 3.5 - Prostředí CANoe (zdroj:vector.com).....	35
Obrázek 3.6 - Vector CAN/LIN interface VN1630A (zdroj:vector.com)	36
Obrázek 3.7 - MGB grabber (zdroj: digiteqautomotive.com).....	37
Obrázek 3.8 - RDS generátor Götting HG 81300 (zdroj goetting-agv.com:)	37
Obrázek 3.9 - blue PiraT mini (zdroj:telemotive.de)	38
Obrázek 3.10 - Diagnostické rozhraní VAS - 6154 (zdroj: skoda-aito.cz)	38
Obrázek 3.11 - USB switcher mini	39
Obrázek 3.12 Zdroje napětí TDK-Lambda (Zdroj: uk.tdk-lambda.com)	40
Obrázek 3.1 - Blokové schéma vstupu AD převodníku NI 9902 (zdroj: ni.com)....	40
Obrázek 4.2 - Nastavení prostorového zvuku ve vozech Audi. (zdroj: myaudiworld.sg.....	43
Obrázek 4.3 - Nastavení ekvalizéru ve vozech Hyundai. (Zdroj: webmanual.hyundai.com).....	44
Obrázek 4.4 - Schéma propojení audio výstupu a AD převodníku.....	46
Obrázek 4.5 - Logaritmické spektrum testovacího signálu FM rádia	47
Obrázek 4.6 - Spektrogram testovacího signálu FM rádia.....	48
Obrázek 4.7 - Logaritmické spektrum signálu v bodě rušení "touch" tónem	49
Obrázek 4.8 - Spektrogram signálu zatíženého "touch" tóny	49

Obrázek 5.1 – Měření v asynchronním režimu. DAC – DA převodník. ADC – AD převodník, XO – krystalový oscilátor, LTI – testovaná lineární soustava.....	58
Obrázek 5.2 Měření v synchronním režimu. DAC – DA převodník. ADC – AD převodník, XO – krystalový oscilátor, LTI – testovaná lineární soustava.....	58
Obrázek 5.3 - Impulsová a frekvenční odezva měřená v asynchronním režimu [33].....	59
Obrázek 5.4 - Autokorelační funkce $R_{yy}[n]$ přijatého signálu $y[n]$. [33].....	59
Obrázek 6.1 - Prostředí Matlab App Designer. (Zdroj: mathworks.com).....	61
Obrázek 6.2 - Prostředí nástroje Audacity. (Zdroj: audacityteam.org).....	62
Obrázek 6.3 - Blokový diagram zobrazení audio záznamu.....	63
Obrázek 6.4 - Testovací signál v časové oblasti	64
Obrázek 6.5 - Spektrogram testovacího signálu.....	64
Obrázek 6.6 - Graf aktivních zdrojů v závislosti na čase.....	67
Obrázek 6.7 - Blokové schéma algoritmu spektrálního odečítání.....	68
Obrázek 6.8 - Chybový signál v čase	70
Obrázek 6.9 - Spektrogram chybového signálu	70
Obrázek 6.10 - Cesta signálu pro měření frekvenční odezvy	71
Obrázek 6.11 - Blokové schéma algoritmu pro získání frekvenční odezvy metodou MLS s korekcí vzorkovací frekvence	72
Obrázek 6.12 - Generovaný MLS signál o délce 1024 vzorků	73
Obrázek 6.13 - Naměřená frekvenční odezva metodou MLS	73
Obrázek 6.14 - Blokové schéma algoritmu metody Sine sweep.....	74
Obrázek 6.15 - Generovaný signál Sine sweep.....	75
Obrázek 6.17 - Frekvenční odezva jednotky získaná metodou Sine sweep.....	76
Obrázek 6.16 - impulsová odezva jednotky získaná metodou Sine sweep	76
Obrázek 6.18 - Blokové schéma algoritmu získávání frekvenční odezvy metodou Segmentace signálu sine sweep	77
Obrázek 6.19 - Frekvenční odezva testované jednotky získaná metodou Segmentace signálu sine sweep.....	78
Obrázek 6.20 - Úvodní strana aplikace	79
Obrázek 6.21- Karta aplikace "Sources and errors"	80
Obrázek 6.22 - Detekce artefaktů v aplikaci	81
Obrázek 6.23 - Spektrogram chybového signálu v aplikaci.....	82
Obrázek 6.24 - Výpočet frekvenční odezvy v aplikaci	83

Seznam tabulek

Tabulka 1.1 - Parametry jednotky Škoda Swing.....	16
Tabulka 2.1- Chyby vyhodnocení testů.....	29
Tabulka 6.1 - Tabulka identifikačních frekvencí zdrojů	65
Tabulka 6.2 - Algoritmus výběru zdroje	66

Úvod

Moderní informační a zábavní technologie jsou stále běžnější součástí lidského života. Výrobci automobilů, kteří si chtějí zachovat konkurenceschopnost na trhu musí na tento podnět reagovat. Moderní automobily jsou vybaveny množstvím informačních, zábavních a komunikačních funkcí, které se jako celek nazývají informačně zábavní systém nebo také infotainment. Posádka vozu má možnost sledovat statistiky o jízdě a stavu vozidla, přehrávat multimediální obsah z mnoha různých zdrojů, propojit vůz s mobilním telefonem nebo využívat navigačního systému s připojením k internetu.

Důležitou součástí vývoje těchto jednotek je testování jejich funkce z pohledu koncového zákazníka. Jak je popsáno v práci v knize „Development of an automated testing system for vehicle infotainment system“ [1], běžným postupem je manuální testování prováděné expertem, který postupně vyvolává jednotlivé funkce systému a dokáže korektně ohodnotit jejich správnou funkci. Tato metoda představuje velká omezení v oblasti pokrytí testování a efektivity kvůli neustále se zvyšujícímu počtu a složitosti funkcí systému a omezenými lidskými možnostmi. Z tohoto důvodu je výrobci kladen velký důraz na automatizaci procesu testování. Automatizovaný testovací systém replikuje práci testovacího experta simulováním interakcí s ovládacími prvky a vyhodnocuje správnou funkci jednotky pomocí rozpoznávání klíčových prvků v obrazovém výstupu systému a zpracování signálu na zvukovém výstupu jednotky.

Tato práce se soustředí na zvukovou stránku testování. Správnost výsledku testu nelze potvrdit pouze na základě obrazové informace z displeje jednotky, zvukový výstup musí vždy korespondovat s obrazovými údaji na displeji. Mezi hlavní funkce vyžadující tuto patří volba zdroje média pro přehrávání. Pokud uživatel například poslouchá FM rádio a rozhodne se vybrat jako nový zdroj hudby například telefon připojený přes bluetooth, nestačí pouze identifikace obrazovky média, ale je třeba s daným zpožděním potvrdit i změnu na výstupu zvukového zesilovače. Mezi další zejména kontrola vah jednotlivých kanálů a nastavení ekvalizéru.

Vzhledem k složitosti testovacího řetězce (pro podrobnější popis vizte [1]) je třeba vyloučit chyby ve zvukovém signálu, které nejsou způsobeny nesprávnou funkcí jednotky. Problémy s externími zdroji zvuku jako např. výpadek signálu generátoru FM signálu přijímaného jednotkou, šum naindukovaný na signálových vodičích od blízkých zdrojů napětí, zkreslení přenosového kanálu, změny v signálu způsobené nečekanou interakcí

člověka s testovanou jednotkou apod. Abychom mohli tyto vlivy vyloučit musíme vědět, zda a kdy v průběhu testu k došlo k podobnému narušení. Studium a metody detekce těchto artefaktů jsou prvním z hlavních témat této práce.

U většiny moderních audiosystémů má uživatel možnost přizpůsobit frekvenční odezvu systému typu obsahu, který si přeje poslouchat. Ať už jde o výrobcem předdefinovaná nastavení pro různé hudební žánry nebo vlastní nastavení pomocí grafického ekvalizéru. K správnému vyhodnocení těchto funkcí je zapotřebí znát přesné údaje o aktuální impulsové odezvě systému. Možné metody jejího určení a problémy s tím spojené jsou dalším tématem této práce.

V první části práce je čtenář seznámen s významem a stručnou historií termínu infotainment. Následně jsou stručně popsány základní parametry infotainmentové jednotky, která byla objektem prováděných testů. V druhé kapitole je čtenáři poskytnut teoretický základ v oblasti testování softwaru infotainmentu. V následující kapitole je stručně představeno pracoviště a testovací stav na kterém byly prováděny pokusy v rámci této práce.

Ve čtvrté kapitole popsáno studium metod zpracování zvuku využívaných při automatizovaných testech na zmíněném pracovišti v rámci práce. Postupně jsou vyjmenovány jednotlivé sledované parametry a metody jejich detekce. Závěrem této kapitoly jsou navržena možná vylepšení, která jsou předmětem dalších částí této práce. Následující pátá kapitola pak seznamuje čtenáře s teorií potřebnou k jejich implementaci.

V šesté kapitole jsou postupně vypsána všechna implementovaná zlepšení, představeny algoritmy použité při jejich implementaci. Závěrem popisu každého algoritmu jsou představeny jimi získané výsledky. Závěrem kapitoly je pak představena samostatná aplikace vytvořená v prostředí Matlab App Designer, která sdružuje všechny implementované algoritmy do uceleného testovacího nástroje.

1 Infotainment

Výraz infotainment má svůj původ v anglických slovech information a entertainment tedy v češtině informace a zábava. Podle článku „Infotainment: Co to je a jak se pozná“ z knihy „Rozumět médiím“ [2] se původně používal pro druh zpravodajství, který vznikl v 70. letech 20. století ve Spojených státech amerických za účelem zvýšení sledovanosti zpravodajství. Smyslem bylo obohatit televizní zpravodajství o prvky zábavních pořadů a nabídnout tak divákům kromě informací i emoce a pobavení. V posledních letech je však tento pojem čím dál častěji spojován s automobilovým průmyslem. V tomto kontextu se jedná o systém, který má za úkol nabídnout stejnou kombinaci informací a zábavy posádce vozidla.

Infotainment vozu sdružuje mnoho funkcí soubor určených pro zvýšení komfortu cestujících. Tvoří ho hlavní jednotka zajišťující většinu jeho funkcí, senzory dodávající jednotce informace o stavu vozidla a okolí a vstupně výstupní prvky zajišťující přenos informací mezi uživatelem a zařízením. Hlavními výstupními komponenty jsou displej autorádia na středovém panelu, který častou slouží i pro ovládání systému, přístrojový štít, u prémiových vozů stále častěji řešený digitální displejem, a zvukový výstup jednotky.



Obrázek 1.1 - Displej infotainmentu vozu Škoda Kodiaq (zdroj: skoda-auto.cz)

Obecně se dá na infotainment nahlížet jako na prostředek sloužící pasažérům jako informační a komunikační prvek. Sdělování je možné formou human – vehicle neboli

člověk – auto a human – human, tedy člověk – člověk. Kapitola Automotive v knize Ergonomie Informace, v originálním znění Information Ergonomics [3], se zabývá zkoumáním uvedených způsobů interakce. Poukazuje se také na vývoj dané problematiky. V článku „Vehicle audio: History: 1920s-1940s.“ [4] jsou popsány počátky vývoje zábavních a informačních systému ve vozidlech. V roce 1920 začaly být radiové přijímače dostupné a zaváděly se téměř do každé domácnosti. Za dva roky na to se začaly instalovat do automobilů amatérská autorádia, která uměla přehrávat analogový signál. V třicátých



Obrázek 1.2 - Digitální přístrojový štít Audi (Zdroj: audi.cz)

letech Američan Paul Galvin přišel oficiálně s prvním integrovaným autorádiem a položil tak základ pro společnost Motorola. Za pár let na to r. 1973 se do rádia začal dávat dekodér, který uměl přijímat dopravní hlášení a informovat tak o dopravních problémech na silnici.

V dnešní době tvoří základní nabídku konkurenceschopných dopravních prostředků zařízení jako je přijímač vysílání FM i AM rádia, přehrávání hudby z externích médií (USB, SD karty, AUX) a zobrazení základních informací o vozidle (počet ujetých kilometrů, aktuální rychlost, aktuální spotřeba, průměrné ukazatele aj.). Vyšší výbava vozu nabízí např. navigační modul, rozšíření dostupných informací o stavu automobilu, Wi-Fi modul pro připojení k internetu, TV-tuner, kvalitnější zvukový výstup s externím zesilovačem a kvalitnějšími reproduktory, možnost párování mobilních telefonů a využití vozidla jako hands-free sady nebo rozšířit možnosti jednotky ve vozidle o vybrané funkce chytrého telefonu pomocí technologie Smartlink, (Bližší popis této technologie lze nalézt v práci Testování použitelnosti pro technologie Smart integration v automobilovém

průmyslu [5]). Dostupné kombinace uvedených prvků ve se odvíjí od výbavového stupně vozu. Sestavení nabízených balíčků funkcionalit má na starost oddělení marketingu, které vede nad tímto tématem různé průzkumy z pohledu poptávky.

1.1 Testovaná jednotka Infotainmentu

Aktuální generace infotainment systémů v ve vozech Škoda auto nese název MIB 2, jedná se druhou generaci modulární platformy pro jednotky infotainmentu koncernu Volkswagen A.G. Tento systém lze nalézt ve vozech většiny značek koncernu, konkrétně ve vozech Škoda od roku 2015 až do současnosti. Jednotky této platformy se dělí do tří úrovní podle vybavenosti a ceny: MIB Entry, MIB Standard a MIB High. [6] Pro test byla zvolena třída MIB Entry, která se prodává ve modelech Fabia, Rapid, Octavia, Karoq, Kodiaq i Superb. Konkrétní testovaná jednotka se aktuálně prodává v modelu Kodiaq pod obchodním názvem „Swing“. Tento typ jednotky byl zvolen na základě konzultací s testery ze Škoda Auto TCI (Test centrum infotainment). Jelikož se jedná o jednotku na konci svého vývojového cyklu, produkuje v testech stabilní výsledky a jako základní model nabídky je zastoupena ve velkém množství prodaných vozů. Navzdory tomu nabízí všechny funkce nezbytné pro potřeby této práce.



Obrázek 1.3 Jednotka Škoda Swing ve voze Škoda Kodiaq (zdroj: skoda-auto.cz)

1.1.1 Parametry

Základní parametry testované jednotky:

Displej	Barevný dotekový, 6,5“
Počet reproduktorů	8
Dostupné hudební zdroje	FM rádio, 3,5 mm jack, SD karta, USB, Smartlink, Bluetooth
USB sloty	2
Konektivita s telefonem	Bluetooth, Smartlink
Další funkce	Car menu

Tabulka 1.1 - Parametry jednotky Škoda Swing

Funkce dostupné v systému se dělí do několika základních skupin, dostupných z hlavního menu.

1.1.2 Rádio

V této sekci má uživatel možnost ladit stanice na krátkých i středně dlouhých vlnách (FM /AM). Stejně tak ukládat své oblíbené stanice do paměti a měnit předvolby pro nastavení RDS protokolu jako dopravní hlášení (TA), automatické přepínání mezi alternativními frekvencemi jednotlivých stanic (AF) apod.

1.1.3 Media

Sekce média sdružuje všechny dostupné interní a externí zdroje hudby. Uživatel má přístup k informacím o přehrávaném titulu, ovládacím prvkům pro přepínání zdrojů a skladem. U zdrojů podporujících přístup do jejich systému souborů, je k dispozici prohlížeč dostupných skladeb.

1.1.4 Sound

Nabídka nastavení zvuku je rozdělena do třech základních částí. První z nich se zabývá nastavením hlasitosti pro jednotlivé typy zvukového výstupu jako hlasitost hudby, parkovacího asistenta, dopravních hlášení a podobně. Další sekce se týká nastavení ekvalizéru jednotky, uživatel může upravovat úroveň hloubek, středů a výšek nebo zvolit jedno z přednastavených režimů od výrobce.

1.1.5 Smartlink

V této nabídce jsou sdruženy technologie integrace chytrých telefonů Android Auto, Apple Carplay a Mirrorlink. V závislosti na modelu a operačním systému může uživatel svůj mobilní telefon používat jako další hudební zdroj nebo navigační systém má také přístup ke kontaktům, hovorům a dalším podporovaným aplikacím.

1.1.6 Car

Nabídka car zprostředkovává informace o jízdě jako spotřebu paliva ujetou vzdálenost a podobně. Vyhodnocuje chybové zprávy na CAN sběrnici dokáže tak upozornit na technický problém vozidla, defekt pneumatiky nebo nízkou hladinu oleje

2 Testování infotainmentu

Tato kapitola se zabývá testováním, zejména softwaru. Z počátku budou vysvětleny základní termíny a s ním spojené a různé testovací metody“. Postupně se zaměříme více na specifika testování softwaru infotainmentových jednotek, konkrétně na automatizované testování zvukového výstupu v rámci testů audio managementu. Závěrem je blíže popsáno testování na pracovišti, na kterém byly prováděny pokusy v rámci této práce.

2.1 Testování softwaru

Softwarové systémy jsou nedílnou součástí lidského života. Přicházíme s nimi do styku každý den, ať už v komerčních produktech, jako mobilní telefony nebo automobily, internetové bankovníctví a sociální sítě, nebo v pracovním prostředí, či zdravotnictví. Nehledě na aplikaci se většina lidí dostala do situace, kdy daný software nepracoval podle očekávání. Vzhledem k širokému využití mohou chyby softwaru přinášet mnoho problémů jako ztráta peněz, času nebo firemní reputace. V určitých aplikacích, například v dopravním průmyslu nebo zdravotnictví, dokonce i k lidským zraněním nebo smrti. Důkladné testování je tedy nedílnou součástí vývoje softwaru. [7]

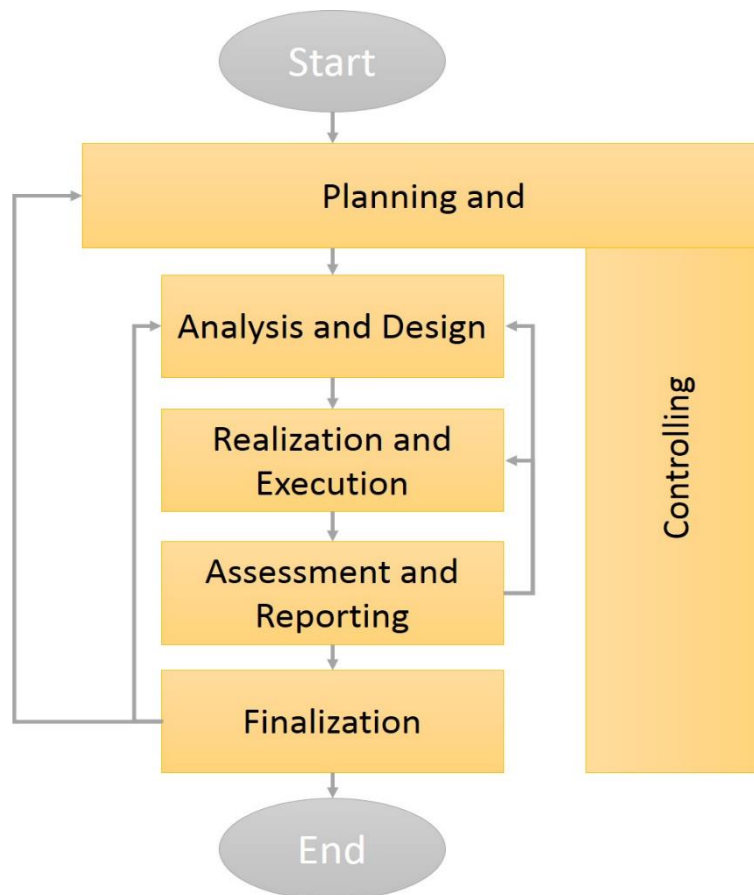
Slovník spisovné češtiny [8] vysvětluje slovo “test“ jako zkoušku k ověřování hodnot, jakosti, vlastností apod. V našem případě se jedná zejména o ověření správné funkce a kvality produktu za pomoci hledání a reportování chyb.

2.2 Proces testování

Aby byla zajištěna požadovaná kvalita testů je třeba při plánování testů, vývoj jednotlivých testů, přípravu všech potřebných nástrojů a vyhodnocení výsledků. Základní testovací proces se podle normy ISTQB [7] skládá z těchto kroků:

- Plánování a kontrola testů
- Analýza a návrh testů
- Implementace a provedení testů
- Vyhodnocení výsledků a reportování
- Finalizace

Ačkoliv jsou tyto aktivity logicky v chronologické posloupnosti, při reálné testování se mohou jednotlivé kroky překrývat nebo běžet paralelně po celou dobu testovacího procesu. Obr.2.1) Bližší informace k jednotlivým částem testovacího procesu je možné nalézt v pramenech [7] a [10], ze kterých byly čerpány informace pro tuto kapitolu.



Obrázek 2.1 - Testovací proces ISTQB (zdroj:fas-cosulting.de)

2.2.1 Plánování a kontrola testů

Ve fázi plánování je důležité stanovit cíle, zda se snažíme například odhalit chyby v rámci vývoje nebo zhodnotit kvalitu už hotového produktu. Definovat strategii a metody vhodné pro testování. Zhodnotit dostupné prostředky a testovací nástroje. V neposlední řadě vybrat vhodné metriky pro kontrolu výsledků testování.

Kontrolou testů se rozumí nepřetržitý dohled nad jednotlivými body procesu. Reportování postupu a průběžných výsledků. V případě potřeby také odhalit nedostatky zvolených testovacích postupů a upravit je, aby nedocházelo ke zkreslení výsledků.

2.2.2 Analýza a návrh testů

Při analýze je hlavním úkolem zhodnotit testovaný objekt, jeho složitost, zaměření a testovatelnost. Informace k tomu potřebné jsou obsaženy dokumentací k testování, která se podle [7] nazývá anglický výrazem „Test basis“, tedy „báze testování“. Součástí této dokumentace jsou typicky funkční požadavky na výsledný produkt, repositář zdrojového kódu a test plán aj. Je třeba zhodnotit testovatelnost jednotlivých částí testovacího plánu a zjistit podmínky nutné pro vývoj a provádění testů.

Na počátku návrhu je třeba identifikovat nejdůležitější části testovacího plánu a prioritizovat vývoj jednotlivých testů. Vytvořit testy s přesně definovanými logickými kroky a jasně verifikovatelným výstupem. Identifikovat nezbytná testovací data, aby bylo možné splnit počáteční podmínky všech testů. Zajistit možnost zpětné kontroly mezi testovou specifikací a jednotlivými testy.

2.2.3 Implementace a provedení testů

Implementací se rozumí hledání a vytváření, vhodných testovacích procedur, vytváření testovacích dat. V případě automatizovaného testování, tvorba testovacích skriptů, simulací interakce s testovaným objektem a podobně. Pro zajištění co nejefektivnějšího postupu se testy skládají do testovacích kampaní, které se zabývají například podobnými funkcemi systému nebo testy se stejnou prioritou. V neposlední řadě se kontroluje správnost navržených testů a testovacích procedur.

Testy se provádějí buď manuálně nebo s využitím automatizovaných nástrojů podle naplánované testovací sekvence. Zapisování výsledků testů, společně se zaznamenáním testované verze produktu, použitých nástrojů a popisu testovacího prostředí. Výsledky testů jsou porovnávány s předpokládanými. Tyto aktivity se opakují s každou novou verzí testovaného produktu, aby bylo možné zkontrolovat, zda byly chyby objevené v předchozí verzi opraveny nebo zda oprava nefunkční části softwaru neodhalila chyby v oblastech, které před opravou nebylo možné spustit. Takový proces se také nazývá regresivní testování.

2.2.4 Vyhodnocení výsledků a reportování

Hodnotí se, zda výsledky a průběh testů vyhovují cílům stanoveným v testovacím plánu. Posuzuje se pokrytí funkční specifikací produktu testy a zda je potřeba vytvořit další testy. Nejdůležitější částí je tvorba testovací zprávy pro zadavatele testu, která je výsledkem

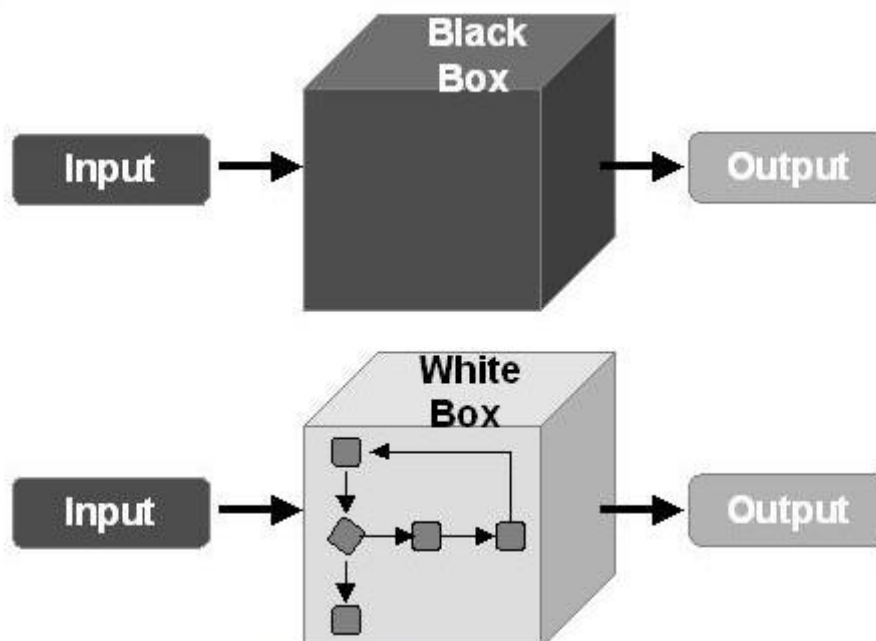
celého procesu a měla by co nejpřehledněji shrnout všechny požadované výsledky definované při plánování testu.

2.2.5 Finalizace

Tato část představuje zakončení testovacího procesu na projektu nebo například zhodnocení dosavadního stavu produktu v předem stanovený časový milník. Kontroluje se, zda jsou celkové výsledky projektu v souladu s plánovaným předpokladem. Rozhoduje se, zda je vyvíjený produkt možné schválit a zařadit například do prodeje. Archivují se výsledky a uvolňují testovací nástroje pro použití v dalších projektech. Analyzuje se průběh projektu za účelem vyvarovat se v budoucnosti případných chyb a využít získané zkušenosti pro zkvalitnění budoucí práce.

2.3 Techniky vývoje testů

Vzhledem k rozmanitosti softwaru a jeho využití ve velkém množství aplikací, je jeho testování velice komplexní disciplína, která dá rozdělit do mnoha kategorií a podkategorií. V této kapitole se přiblížíme nejpoužívanější způsoby testování a jejich definující vlastnosti. Pro hlubší studium tohoto tématu je možné nahlédnout do jednoho ze zdrojů použitých k tvorbě této kapitoly [7] [9].



Obrázek 2.2– Porovnání technik černé a bílé skřínky, (zdrojsoftwaretestinggenius.com:)

2.3.1 Statické a dynamické testování

Rozhodujícím faktorem v tomto rozdělení je možnost zkoušený produkt zpustit.

Statické testování nevyžaduje běh softwaru, proto ho s výhodou provádět ještě před začátkem psaní kódu. Předmětem testů může být i dokumentace. Chyby nalezené v dokumentaci mohou ulehčit práci vývojářům, umožnit lepší odhad náročnosti dalších částí projektu a informace získané v průběhu tohoto testování mohou být využity k efektivnějšímu vývoji dalších testů.

Dynamické testování vyžaduje přístup k spustitelné verzi softwaru a představuje většinu testovacího cyklu. Způsobů dynamického testování je mnoho a některé z nich jsou popsány níže.

2.3.2 Černá a bílá skříňka

Termíny černá a bílá skříňka se označují skupiny technik tvorby testů. Z originálních názvů „Black box testing“ a „White box testing“ nebo také „Clear box testing“ je lépe poznat hlavní dělicí faktor těchto dvou skupin, kterým je míra znalosti zdrojového kódu či vnitřních funkcí testovaného produktu.

Techniky spadající do skupiny „Bílá skříňka“ se vyznačují tím, že má vývojář k dispozici zdrojový kód testovaného produktu. Patří mezi ně například „Statement testing and coverage“. Metoda sledující procentu spustitelných výrazů ve zdrojovém kódu.

Při testování černé skříňky (black box) se zaměřujeme na vstupy a výstupy programu bez znalosti jeho implementace. Produkt je černou skříňkou, do které nelze nahlédnout. Můžeme pozorovat jen její vnější vzhled a interakce s okolním prostředím. Smyslem je analyzovat chování softwaru vzhledem k očekávaným vlastnostem z pohledu uživatele. Příkladem může být „Use case testing“, tedy metoda zkoumající typické příklady použití daného produktu. Definují se akce uživatele a kontroluje se „odpověď“ programu. Příklad takového testu je k nahlédnutí v dále v textu.

2.3.3 Hledání úspěchu, hledání neúspěchu

Rozhodující faktor je v tomto případě zřejmý z názvů kategorií, v prvním případě se hledá pozitivní výsledek testované funkce a v druhém hledáme chyby.

Na začátku vývoje aplikace, např. při první předávce verze k testování, se používají testy z kategorie „Hledání úspěchu“. Začíná se testováním hlavních případů užití programu a kontroluje se, kolik těchto funkcí je funkčních a způsobilých k dalšímu testování.

Příkladem mohou být testy průchodu neboli „Test to pass“, kdy stačí nalézt alespoň jednu cestu při které se dosáhne požadovaného výsledku a test je považován za úspěšný.

Naopak na konci vývoje už se funkcionality považují za stabilizované, úspěšný průchod scénářem od začátku do konce se považuje za samozřejmost. V tuto chvíli nastupují testovací metody „Test to fail“ hledající chyby. Pokud se najde jen jediný případ, kdy testovaná funkcionality selže, je takovýto test považován za neúspěšný.

2.3.4 Automatické a manuální testování

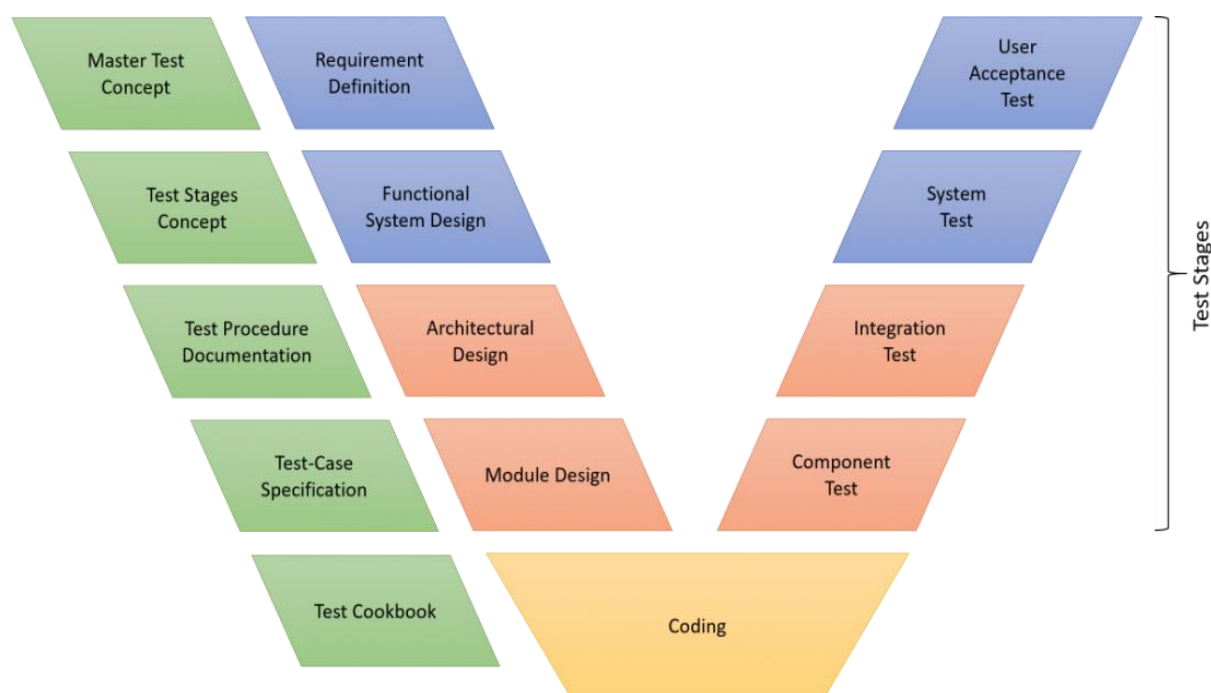
Podle toho, zda jsou testy prováděny člověkem nebo softwarem, se rozlišuje manuální a automatické testování. Každý z těchto přístupů má své výhody a nevýhody a místo v procesu vývoje.

Manuální testování je vhodné zejména ze začátku vývoje, kdy se předpokládá vysoká chybovost, také u nových testů, které ještě nebyly zautomatizovány. U testů vyžadujících lidské ohodnocení a úsudek nebo rozličné přístupy, které není třeba zaznamenat a pravidelně opakovat se používá výhradně manuální testování.

Jsou ale případy, při kterých je vhodnější využít automatizovaného testování. Zejména při opakovaném procházení stejného scénáře, například při pokusech o verifikaci údajně opraveného defektu nebo při testech, které obsahují mnoho jednoduchých nebo se opakujících kroků, které by pro člověka mohly být únavné a monotónní. Takovéto situace zvládají automatizované systémy zpravidla podstatně rychleji a není u nich nebezpečí přehlednutí potenciálních chyb z nepozornosti. Výhodou je také jistota vždy identického průběhu testu a možnost kontinuálního testování v průběhu několika hodin i dní.

2.4 Úrovně testování

Stejně tak jako vývoj produktu prochází od napsání prvního řádku kódu několika fázemi, dělí se i testování v průběhu tohoto vývoje do několika úrovní. Na následujících řádcích si jednotlivé úrovně představíme spolu se stručným popisem jejich definujících charakteristik. Na obrázku (2.2) je vidět, jak níže popsané testovací úrovně zapadají do „V-modelu“ vývoje softwaru. [7] Pro detailnější informace lze nahlédnout do jednoho z použitých zdrojů. [7] [9]



Obrázek 2.3 - V- model vývojového procesu, (zdroj: fas-consulting.de)

2.4.1 Vývojové testování

První fáze kontrolu softwaru se provádí už při samém začátku jeho vývoje. Programátor, který kód píše, ho po sobě také kontroluje.

2.4.2 Komponentní testování

Komponentní testování známé také pod anglickými termíny „Component testing“ nebo „Unit testing“ se snaží nalézt chyby nebo verifikovat správnou funkci jednotlivých tříd, objektů funkcí a jiných částí programů, které jsou samostatně testovatelné. Testování modulu probíhá separovaně od zbytku systému. Jsou-li k jeho funkci potřeba jiné komponenty systému, mohou být v testovacím prostředí simulovány.

2.4.3 Integrovační testování

Naopak komunikační rozhraní mezi jednotlivými komponenty, interakce s ostatními částmi systému jako operační systém, hardware nebo i komunikace mezi zařízeními se kontroluje ve fázi integrovačního testování. Integrovační testování lze provádět na více úrovních:

1. Testování jednotlivých komponent softwaru, „Component integration testing“, které se provádí po komponentním testování
2. „System integration testing“ neboli testování systémové integrace zaměřené na interakce mezi různými systémy nebo mezi hardwarem a softwarem. V tomto případě může společnost provádějící test mít k dispozici jenom jednu stranu rozhraní. Například komunikace se vzdáleným serverem. Tato úroveň může být prováděna až po systémovém testování.

2.4.4 Systémové testování

Tato část testovacího procesu se soustředí na chování systému nebo produktu jako celku. Testovací prostředí by mělo korespondovat s cíleným zaměřením produktu, aby se minimalizoval risk nerozpoznání selhání systému v situacích specifických pro dané prostředí. V této fázi se hojně používají metody z kategorie „Black box“ například „Use case testing“, kdy se hodnotí chování produktu z pohledu uživatele v modelových případech použití.

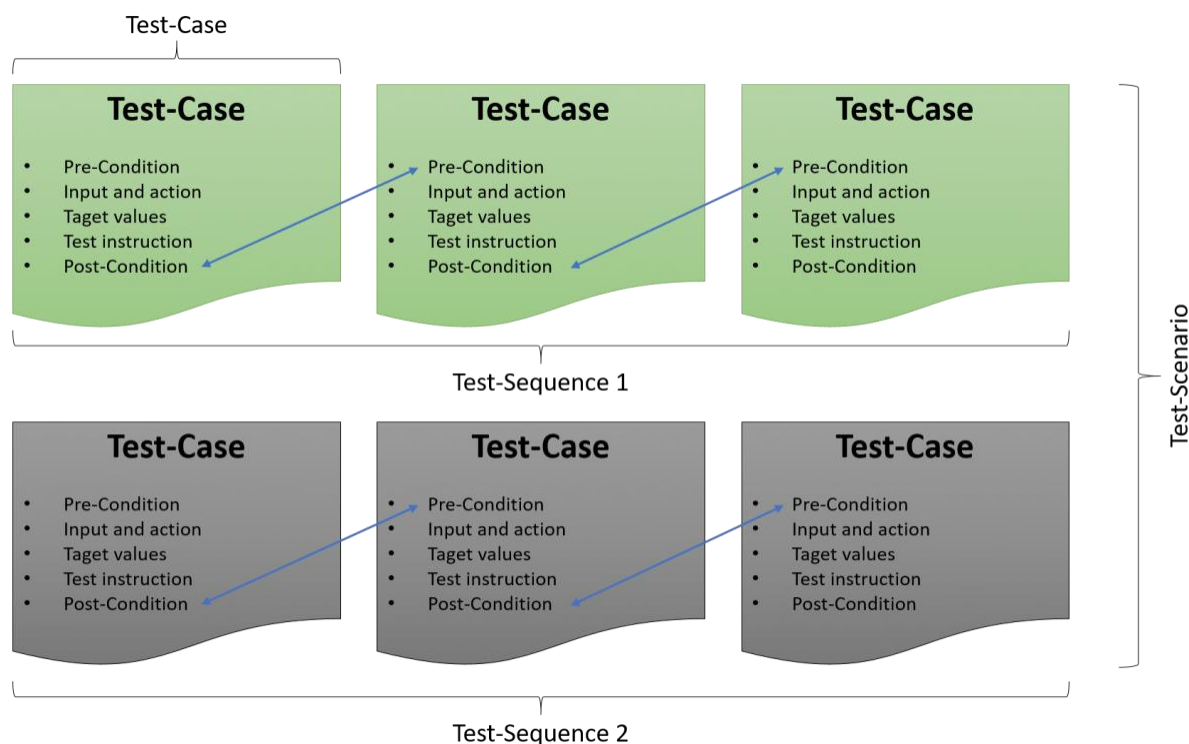
2.4.5 Akceptační testování

Akceptační testování je běžně prováděné klientem, který si sám ověří kvalitu výsledného produktu a míru splnění zadání projektu. Součástí bývá „Pilotní testování“, což znamená ověření všech funkcí produktu při zamýšleném užívání v reálném prostředí na rozdíl od simulací užívaných při testování v laboratořích.

2.5 Forma testu

Směrnice ISTQB [7] definuje několik základních těles, na které lze vlastní proces testování rozdělit. Prvním z nich je „Test case“. Tento anglický termín popisuje jeden testovací případ. V průběhu toho „test casu“ by měla být ověřena vždy jedna funkce systému. Jeho jednotlivé části si popíšeme zanedlouho. Pokud zřetězíme více podobných „test casů“ za sebe, dostane sekvenci testů zvanou „Test sequence“. Tato sekvence typicky řeší několik podobných nebo na sebe navazujících funkcí. Ještě širší záběr v testovacím

procesu pokrývá termín „Test scenario“, neboli česky testovací scénář, či kampaň. Ten v sobě analogicky ukrývá více testovacích sekvencí a může obsahovat například test celé sady funkcí nebo jednoho produktu. Pokud bychom si chtěli uvést příklad z reálného světa, například na testech infotainmentu, v rámci jednoho „test casu“ bychom mohli ověřit reakci



Obrázek 2.4– Skladba „test casů“, (zdroj: fas-consulting.de)

jednotky na otočení ovladače pro změnu hlasitosti směrem „vzhůru“, test sekvence by obsahovala více testů týkajících se hlasitosti výstupu a testovací scénář by obsahoval všechny funkce týkající se úpravy charakteru zvukového výstupu.

2.5.1 Test case

V této kapitole si přesněji popíšeme prvky tvořící „test casy“ neboli jednotlivé testovací případy a jejich význam.

2.5.1.1 Pre-kondice

Kvůli zachování relevance výsledků je potřeba pře začátkem každého testu uvést zkoumaný produkt předem jasně definovaného stavu. V zápisu testu tuto roli plní pre-kondice. Obsahuje přesný popis úkonů, které je potřeba vykonat, pře začátkem testu. Pokud test selže jedna z pre-kondicí, další úkony se nevykonávají, test je označen za neplatný a tester nebo automatizovaný skript přechází k pre-kondicím dalšího testu,

2.5.1.2 Akce

Hlavní část testu obsahující provedení a vyhodnocení úkonů, jejichž vyhodnocení rozhoduje o výsledku testu. V ideálním případě by tato část měla zahrnovat minimum kroků, aby bylo vyhodnocení testu co nejjednoznačnější.

2.5.1.3 Post-kondice

Závěrečná část testu zajišťující návrat testovaného produktu do původního stavu. Zvláště důležitá je tato část při řetězení „test casů“ do sekvencí, kdy, jak je možné vidět na obrázku (2.2), post-kondice jednoho testu může představovat pre-kondici následujícího. Z tohoto důvodu selhání některé z post-kondic, sice neovlivní výsledek aktuálního testu, ale zejména u automatizovaného testování může negativně ovlivnit výsledky následující.

2.5.2 Vyhodnocení testu

Jakýkoli test nebo experiment je bezvýznamný, pokud nejsou jeho výsledky správně vyhodnoceny. O metodiku vyhodnocování testů je možné se dočíst mnoho informací v pramenech [7] a [10]. V následující části si přiblížíme hlavní skupiny možných výsledků a jejich definující vlastnosti.

2.5.2.1 Pass

Pokud je odezva na všechny kroky testu shodná s předpokládanými výsledky podle funkční specifikace testovaného produktu, může být výsledek celého testu označen anglickým termínem „pass“. Český se dá toto slovo přeložit jako „průchozí“, tedy: test proběhl přesně podle předpokladu a je vyhodnocen kladně.

2.5.2.2 Open

Pokud selže jeden z kroků z kategorie pre-kondic (2.5.1.1), je vyhodnocen jako „open“, nebo česky „otevřený“. Výsledek tedy není ani pozitivní ani negativní, ale jelikož test selhal už v přípravě počátečních podmínek, nedá se vyhodnotit. Chybu selhané funkce by měl pokrývat jiný test case.

2.5.2.3 Fail

Význam toho anglického slova znamená v češtině „selhání“. Pokud jedna z hlavních akcí testu selže, celý test se označuje jako „fail“. Při objevení problému je, zejména při automatizovaném testování, důležité ověřit, zda je chyba opravdu v testovaném produktu nebo jí způsobilo špatné provedení nebo implementace testu, nebo selhání jednoho z testovacích nástrojů.

2.5.2.4 Chyby testů

Chyby způsobené selháním testu můžeme rozdělit do dvou hlavních skupin:

- False negative

Tímto pojmem se nazývá situace, kdy test chybně označí správné chování sledované funkce jako nesprávné. Při neodhalení této chyby může dojít ke zbytečnému prodloužení vývojového procesu. Z pohledu statistiky se jedná o „chyby typu I“. Tedy chybné zavrnutí nulové hypotézy. [11]

- False positive

V tomto případě jde o chybné schválení průběhu testu jako správného, jinými slovy o neodhalení chyby testovaného produktu. Jedná se o nejzávažnější chyby při návrhu nebo provedení testu, protože generuje nepravé pozitivní výsledky: při ignorování těchto chyb může dojít k „probublání“ zátažných chyb až do produkčních verzí produktů. Statistika tento jev popisuje jako „chybu typu II“ [11].

Chyby testů		Skutečnost	
		Pass	Fail
Výsledek testu	Pass	OK	"False positive"
	Fail	"False negative"	OK

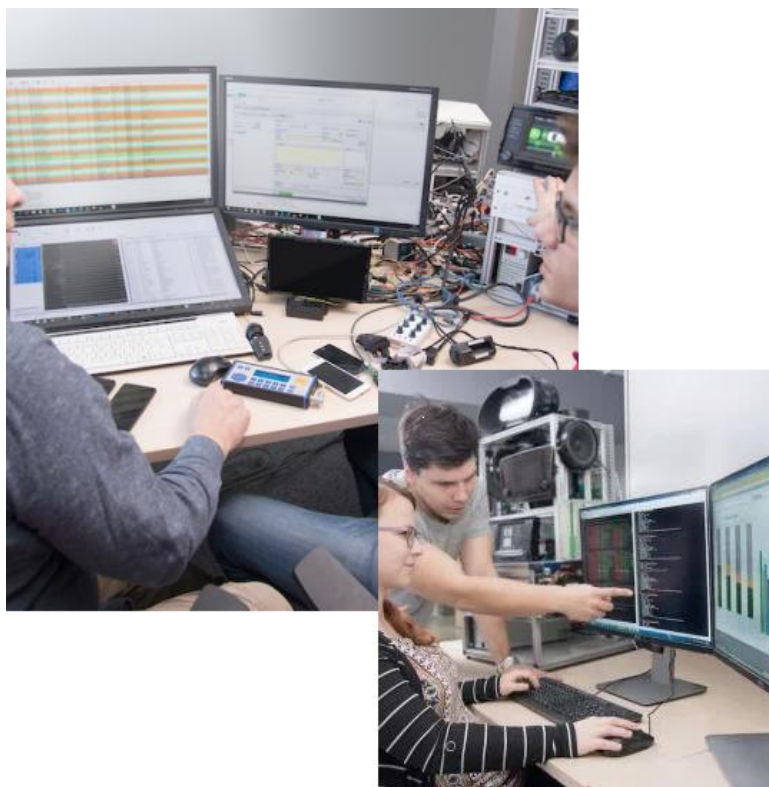
Tabulka 2.1- Chyby vyhodnocení testů

3 Popis testovacího prostředí

V průběhu této práce jsem měl k dispozici testovací stav určený pro automatizované testování „Audio managementu“ v prostorách firmy Digiteq automotive s. r. o. Sledoval jsem techniky místních testerů a programátorů. Podle zadání práce jsem se soustředil na metody hodnocení zvukového výstupu zde testovaných infotainmentových jednotek. V následující kapitole si stručně popíšeme náplň práce v daném oddělení. Techniky testování zde prováděné.

3.1 Popis pracoviště

Laboratoř, ve které byly prováděny pokusy spadá do oddělení test centra Digiteq automotive s.r.o., konkrétně pro oddělení zabývající se kompletním testováním jednotek infotainmentu Škoda MIB 3 do všech aktuálních a budoucích modelů Škoda Auto. Vzhledem k zaměření na komponentní testování, které probíhá z počátku vývoje a dá se očekávat velká chybovost, je většina testů prováděna manuálně na speciální testovacích



Obrázek 3.1- prostředí laboratoře komponentního testování
(digiteqautomotive.com)

stavech umožňujících simulaci reálného prostředí jednotek. Vzhledem velké konkurenci v automobilovém průmyslu je rychlost vývoje nových funkcionalit do vozidel velmi

důležitá. Proto je kladen velký důraz na automatizaci testování. Manuální testování je časově velice náročné, a proto je výhodné ulehčit práci manuálním testerům automatizací co největšího počtu „test casů“ a zkrátit tak čas potřebný pro otestování každé verze softwaru. Místní tým využívá specializovaný automatizační framework a vyvíjí vlastní testovací nástroje a simulátory. Více informací lze na příklad z webových stránek společnosti [12].

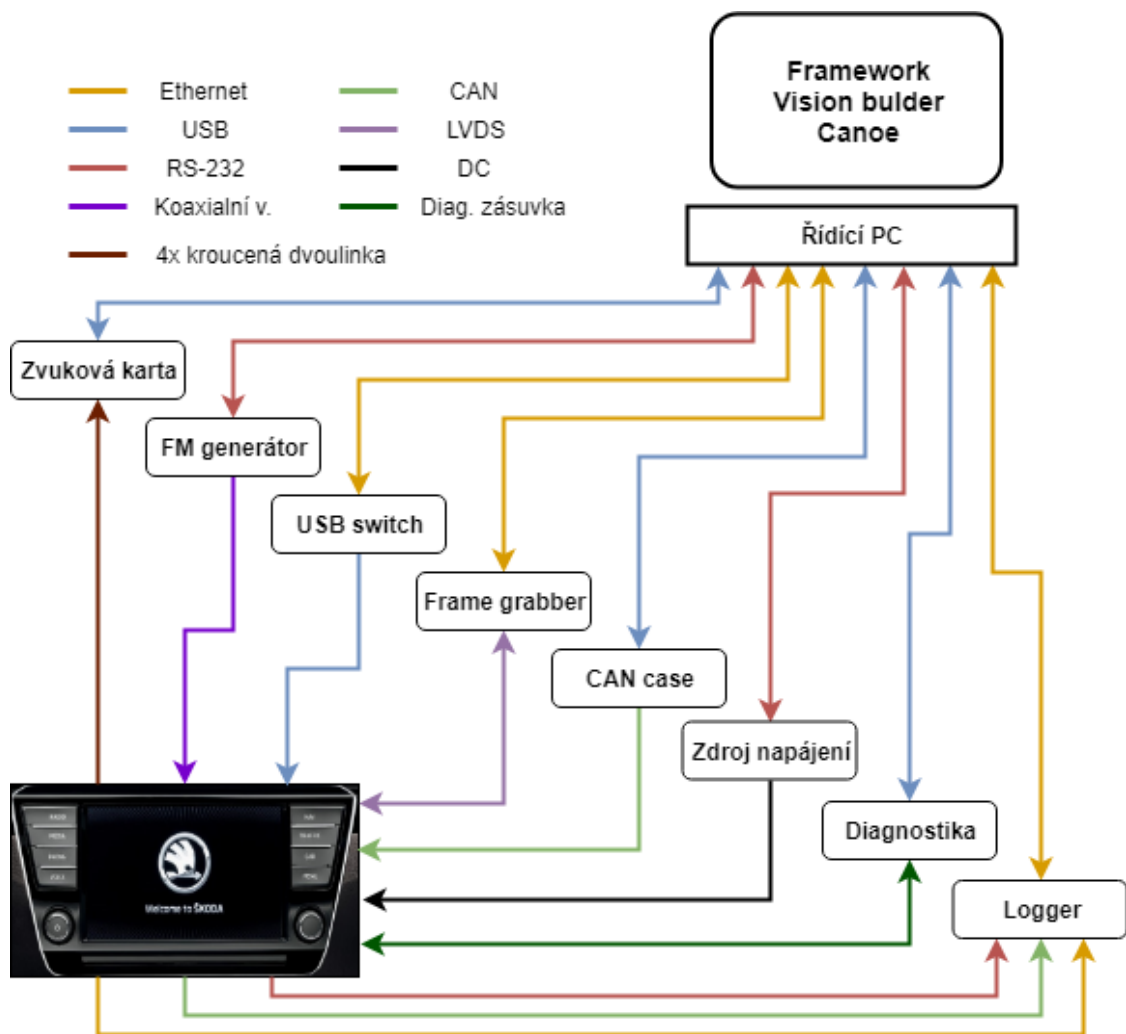
3.2 Testovací stav

Pro testování byl zapůjčen stav určený pro automatizované testování „Audio management“. Jak název napovídá, hlavní náplní těchto testů je kontrola funkcí jednotky starajících se o správu zvuku. V následující kapitole si tyto funkce popíšeme podrobněji. Samotný stav se skládá z řídicího počítače a periférií zajišťujících komunikaci s testovanou



Obrázek 3.2 - ilustrační foto testovacího stavu. (zdroj: digiteqautomotive.com)

jednotou a vyhodnocování výstupních hodnot. Umožňuje komunikaci po CAN a sériové sběrnici. Snímání obrazu přímo z displeje a simulaci interakce uživatele. Specifika „audio stavu“ představuje specializovaná zvuková karta pro nahrávání zvukového signálu do počítače, FM generátor pro simulaci rádia a měnič na USB. V této kapitole si stručně popíšeme všechny stavební prvky testovacího stavu, které nepodléhají utajení.



Obrázek 3.3 - blokové schéma testovacího stavu

3.2.1 Řídící PC a software

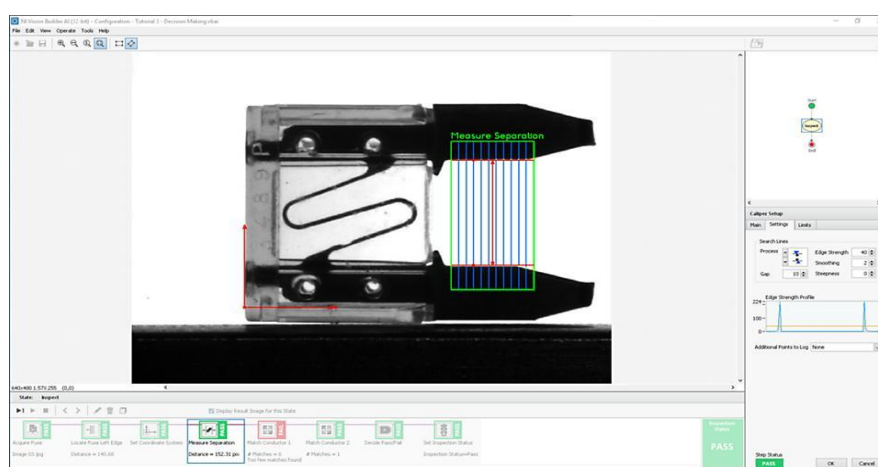
Důležitým stavebním kamenem testovacího stavu, zejména pro automatizaci, je řídicí počítač. Na kterém běží automatizační framework potřebný pro běh testovacích skriptů a další software nutný pro běh testování.

3.2.1.1 Testovací Framework

Pod zkratkou AAAMU se skrývá Framework vyžívaný v několika odděleních testovacích center v koncernu Volkswagen A.G. zabývajících se testováním infotainmentu. Je psaný v programovacím jazyce C++ a dovoluje vývojářům využívat široké množství specializovaných rozhraní pro komunikaci s mnoha variantami jednotek. Jeho přesná funkce však podléhá utajení a není pro tuto práci velmi důležitá.

3.2.1.2 NI Vision builder

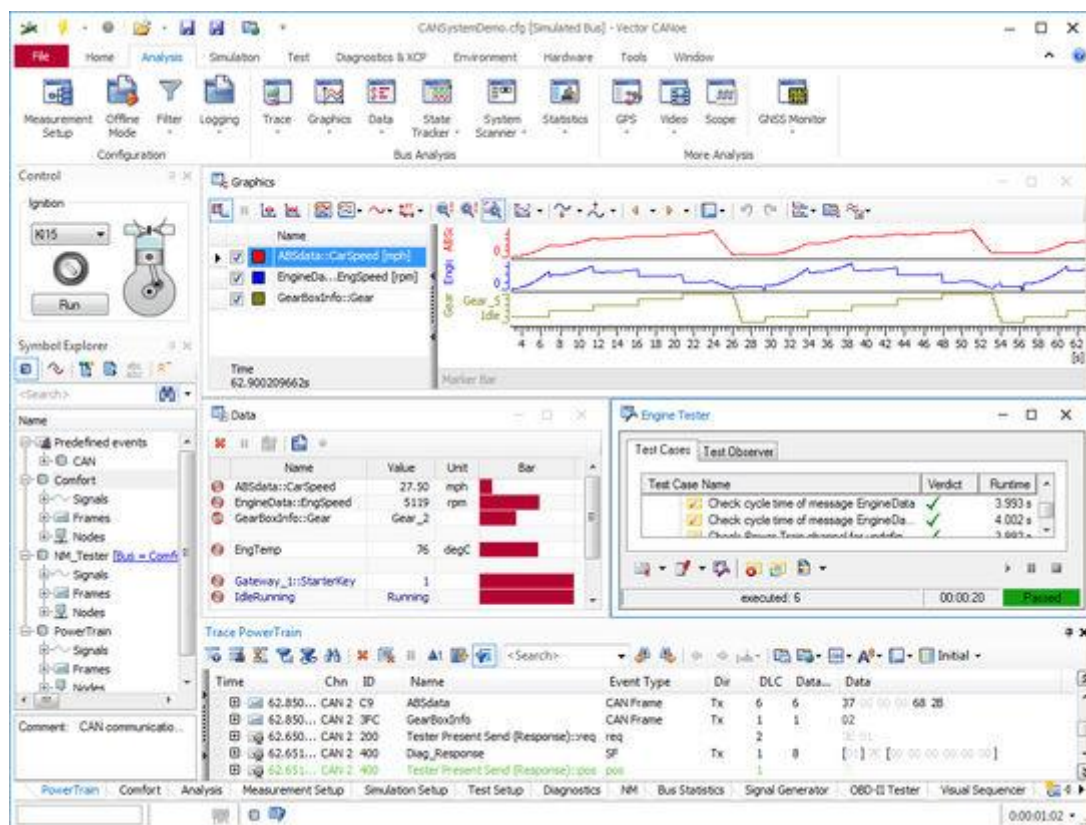
Vision Builder for Automated Inspection je softwarový nástroj od firmy National Instruments navržený specificky pro zpracování obrazu při automatizaci. Je využíván v širokém spektru aplikací nejen při testování, ale například i při kontrole materiálu v průmyslové výrobě. Nabízí vývojářům řadu předpřipravených funkcí pro škálování a filtraci obrazu nebo pro rozpoznávání textu. V naší konkrétní aplikaci je využíván pro rozpoznávání navigačních prvků na snímcích obrazovky rádia získaných za pomoci frame grabberu (3.2.3). Více informací lze nalézt na oficiálních stránkách výrobce. [13]



Obrázek 3.4- Prostředí Vision Builder AI (zdroj:ni.com)

3.2.1.3 CANoe

Nástroj CANoe od společnosti Vektor je využíván v automobilovém průmyslu pro vývoj a analýzu elektronických řídicích jednotek nebo celých systémů. Používají ho síťoví návrháři, vývojoví i testovací inženýři v celém vývojovém procesu od plánování až po systémové testování. O jeho další využitích se lze dočíst na internetových stránkách společnosti. [14]. V rámci test centra je využíván zejména k simulaci řídicích jednotek vozidla, se kterými potřebuje jednotka infotainmentu komunikovat, aby její chování odpovídalo použití v reálném voze. Simulovaná komunikace probíhá zejména po sběrnici CAN. V rámci testování se tento software také používá ke kontrole a komunikace po této sběrnici. Funkce vytvořených simulací lze automatizovat vnitřními skripty nebo pomocí rozhraní softwaru využívat funkce simulací ve vlastních testovacích skriptech.



Obrázek 3.5 - Prostředí CANoe (zdroj:vector.com)

3.2.2 CAN case

Výrazem CAN case se označuje rozhraní pro komunikaci na stejnojmenné automobilové sběrnici. Vzhledem k využívání softwaru CANoe se v test centru Škoda využívají výrobky společnosti Vektor přímo navržené pro spolupráci s jejími nástroji jako CANoe, CANalyzer, CANape nebo pro vlastní aplikace vývojářů. Využívají se pro analýzu komunikace na sběrnících, simulace i kalibraci řídicích jednotek. Všechny ostatní funkce softwaru CANoe. Bližší informace jsou k dispozici na stránkách výrobce [16].



Obrázek 3.6 - Vector CAN/LIN interface VN1630A (zdroj:vector.com)

3.2.3 MGB grabber

Modularer Grabber Baukasten (MGB ~ „Modulární Grabbovací stavebnice“) je zařízení určené pro snímání obrazu z displejů zařízení skupiny Volkswagen, infotainmentové jednotky, digitální přístrojové štíty a podobně. Zařízení bylo vyvinuto firmou Digiteq automotive ve spolupráci se společnostmi Škoda auto, Volkswagen a Audi. MGB je připojeno LVDS kabelem přímo mezi hlavní řídicí jednotku a displej. Zachytává obraz a přeposílá ho do PC pomocí gigabitového ethernetu jako video ve formátech H.264, MJPEG a RAW nebo jako jednotlivé snímky v formátu PNG. Podrobnější informace jsou k dispozici na stránkách výrobce [15]. V rámci automatizovaného testování se používá při navigaci v uživatelském rozhraní jednotky. Po každém simulovaném stisku tlačítka se vytvoří snímek obrazovky, který je nadále zpracováván za pomoci NI Vision Builderu.



Obrázek 3.7 - MGB grabber (zdroj: digiteqautomotive.com)

3.2.4 FM generátor

Pro testování funkcí FM tuneru mít kontrolu nad kvalitou přijímaného signálu vysílání i nad funkcemi standardu RDS. Za tímto účelem se používá RDS generátor Götting HG 81300. Podle informací na stránkách výrobce [17] je toto zařízení simulovat vysílání radiové stanice v FM pásmu s plným pokrytím RDS služeb. Uživatel může měnit hodnotu nosných i modulačních frekvencí, simulovat dopravní hlášení i bezpečností upozornění nebo pomocí aplikace pro PC a sériového rozhraní vytvářet a nahrávat do paměti generátoru vlastní stanice. Pro účely testování je vytvořeno několik stanic, každá se specifickými

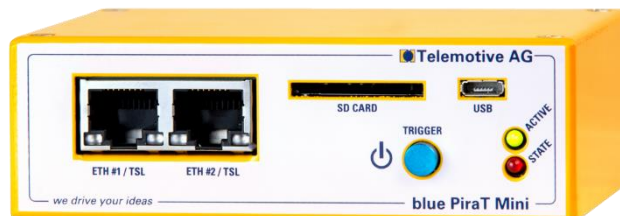


Obrázek 3.8 - RDS generátor Götting HG 81300 (zdroj goetting-agv.com:)

vlastnostmi, mezi se přepíná pomocí sériového rozhraní přístroje a sleduje se korektní reakce jednotky.

3.2.5 Logger

Nedílnou součástí testování je získávání a ukládání „logů“ o chování jednotky v rámci testů, aby bylo možné vývojovým týmům dodat dostatek informací k nezbytným úpravám softwaru. Datalogger blue PiraT mini od společnosti Telemotive AG zvládá zaznamenávat komunikace ze všech potřebných komunikačních kanálů jednotky, tedy na CAN a sériové sběrnici a na ethernetovém výstupu. Přesný počet podporovaných portů a další parametry lze získat na webu výrobce. [18]



Obrázek 3.9 - blue PiraT mini (zdroj:telemotive.de)

3.2.6 Diagnostické rozhraní

Moderní vozidla jsou široce konfigurovatelná, aby vyhověli co největšímu počtu zákazníků a jeden typ infotainmentu se zpravidla montuje do více než jednoho typu vozidel. Aby bylo možné zhodnotit správnou funkci jednotky pro co největší počet variant modelů a výbav, je třeba dynamicky měnit její konfiguraci v průběhu testování. Za tímto účelem je využíváno speciální diagnostické rozhraní Volkswagen VAS 6154 USB, které spolupracuje se specializovaným softwarem, který automatizovaně nahraje do jednotky



Obrázek 3.10 - Diagnostické rozhraní VAS - 6154 (zdroj: skoda-

konfiguraci podle požadovaného typu „vozidla“. Mezi další funkce patří možnost vyčíst chybovou paměť všech elektronických řídicích jednotek ve vozidle nebo na testovacím stavu. Na stránkách Škoda auto [19] lze nalézt další nabízené funkce a přesné parametry hardwaru.

3.2.7 USB přepínač

Mnoho testů z kategorie z kategorie „Audio management“ obsahuje jako jeden z hlavních kroků změnu zdroje zvukového signálu. Patří mezi ně FM nebo digitální rádio, ale také externí multimediální zdroje jako datové nosiče různé druhy mobilních telefonů a hudební přehrávače. Pro automatizaci testů podobného charakteru je nezbytnou schopností měnit tyto zdroje na dálku bez nutného fyzického zásahu člověka. Toho lze dosáhnout například využitím robotické ruky. U zdrojů využívajících k pevnému propojení s jednotkou konektor USB je výhodnější využít síťově řízený USB přepínač, a to jak z finančního pohledu, tak z hlediska spolehlivosti. Na „našem“ stavu je použit přepínač Mini USB switcher vyvinutý firmou Diqiteq automotive [20]. Tento přepínač lze vzdáleně ovládat vzdáleně pomocí ethernetového rozhraní nebo manuálně pomocí tlačítek předním panelu. Připojit lze až osm zařízení, které je možné nabíjet zároveň.



Obrázek 3.11 - USB switcher mini

3.2.8 Zdroj napájení

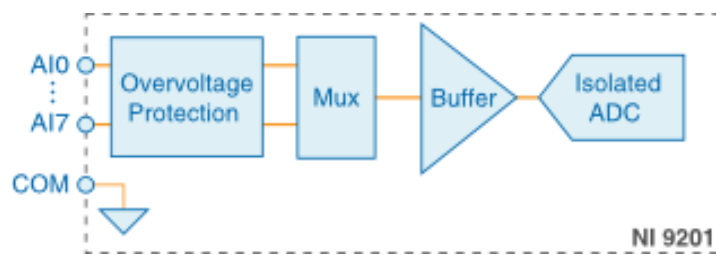
Řízený zdroj TDK-Lambda ZUP 20-20 se kromě napájení testované jednotky a kontrolování jejího odběru využívá i tak zvaným „Stress testům“, při kterých se kontroluje reakce jednotky na přepětí nebo podpětí. Výstupní napětí, proudové omezení a další parametry lze měnit vzdáleně pomocí rozhraní RS-232. Rozsah maximální výstupní napětí je 20 V stejně tak jako maximální proud 20 A a výstupní výkon 400 W. Podrobnější informace je možné získat z dokumentace dostupné na webu výrobce. [22]



Obrázek 3.12 Zdroje napětí TDK-Lambda (Zdroj: uk.tdk-lambda.com)

3.2.9 Zvuková karta

Při testování i k nahrávání audio souborů k pokusům v rámci této diplomové práce byla jako zařízení pro nahrávání zvuku použitý analogově digitální převodník National Instruments 9902. Na každý ze čtyř vstupů převodníku je přímo přiveden výstupní signál čtyř kanálového vestavěného zesilovače jednotky. Převodník je schopen zvládat vstupní napětí $-50\text{ V} - 50\text{ V}$, pracuje se vzorkovací frekvencí 50 kHz a maximálně 24 bity na vzorek. Příložený software dovoluje výsledná data ukládat například holém v „raw“ formátu, nebo jako nekomprimovaný zvukový soubor „wave“. Zařízení také disponuje přepěťovou ochranou na všech vstupech. Řízení i přenos dat do PC obstarává port USB. Bližší informace lze najít v dokumentaci dostupné z webových stránek výrobce. [20]



Obrázek 3.13 - Blokové schéma vstupu AD převodníku NI 9902 (zdroj: ni.com)

4 Studium zpracování zvuku při testech

Vzhledem k zaměření práce na kontrolu zvukového výstupu jednotky si následující kapitole podrobněji popíšeme typy testů, které využívají zpracování zvuku, kontrolované parametry a používané metody. Zaměříme se na zkoumání rušení a vznik zvukových artefaktů, které mohou nastat při testování za stávající konfigurace stavu a na možné rozšíření funkcí zpracování signálu při automatizovaných testech, které bude hlavním předmětem dalších kapitol této práce.

4.1 Typ testů

Jak už bylo řečeno v předchozích kapitolách, předmětem zájmu této práce jsou automatizované komponentní testy zkoumající skupinu funkcí jednotky infotainmentu nazývanou jako „audio management“. Z předchozích kapitol (2.4.2) víme, že se komponentní testy soustředí na produkt jako takové, a ne na jeho zakomponování do systému. Samotný návrh testů probíhá metodou černé skříňky (2.3.2). Tedy neznáme vnitřní chování jednotky, řešíme pouze reakci na interakce uživatele. Z názvu sledované skupiny funkcí vyplývá, že se jedná o práci se zvukovým výstupem jednotky. Nejedná se však o kontrolu kvality výstupního signálu ve smyslu kontroly míry zkreslení, odstupu signálu od šumu nebo přesných frekvenčních charakteristik zesilovače. Těmito měřeními se zabývají specializovaná pracoviště. Testy této kategorie jsou zaměřené na funkčnost ovládacích prvků uživatelského rozhraní a jejich vliv na zvukový výstup. Například ovládání hlasitosti, přepínání zdrojů signálu, nastavení ekvalizéru a podobně. Pro představu můžeme uvést příklad jednoduchého testu podle struktury popsané v kapitole (2.5.2):

4.1.1 Příklad testu – Změna zdroje z FM rádia na USB

- Pre-kondice
 - Jednotka je aktivní
 - FM rádio je zvoleno jako aktivní zdroj
 - Audio výstup je slyšitelný.
- Akce
 - Zvolte externí USB jako aktivní zdroj a aktivujte přehrávání
- Post-kondice
 - Jednotka je ve stavu „zapnuto“
 - Audio výstup slyšitelný
- Očekávaný výsledek
 - Zdroj hudby změněn FM rádia na externí USB
 - Audio výstup USB slyšitelný

4.2 Sledované parametry

K pokrytí požadovaného rozsahu funkcí je třeba kromě správných reakcí uživatelského rozhraní systému několik základních parametrů výstupního signálu zesilovače. V následujících bodech jsou popsány jednotlivé sledované parametry spolu se popisem metod využívaných k jejich sledování v laboratoři, kde byli prováděny pokusy v rámci této práce. Přesné algoritmy zde využívané zde nemohou být popsány z důvodu utajení, popis je tedy pouze rámcový.

4.2.1 Úroveň hlasitosti

Ovládání hlasitosti patří u jakéhokoli přehrávače hudby mezi nejzákladnější funkce. U infotainmentových jednotek ve voze to platí o to více, jelikož je třeba řidiči kromě zábavy v podobě hudby zprostředkovat také mnoho dalších informací. Například navigační instrukce nebo dopravní hlášení. U většiny systémů má řidič možnost zvolit si hlasitost pro každý typ upozornění. Kontrolu nastavení hlasitosti pro jednotlivé typy zdrojů pokrývá velké množství „test casů“.

Kontrola hlasitosti je prováděna škálováním výstupní amplitudy signálu.

4.2.2 Rozpoznávání zdroje

Dalším důležitým faktorem, je kontrola aktivního zdroje. At už ke kontrole funkčnosti jednotlivých zdrojů nebo k detekci upozornění řidiče zmíněném v předchozím

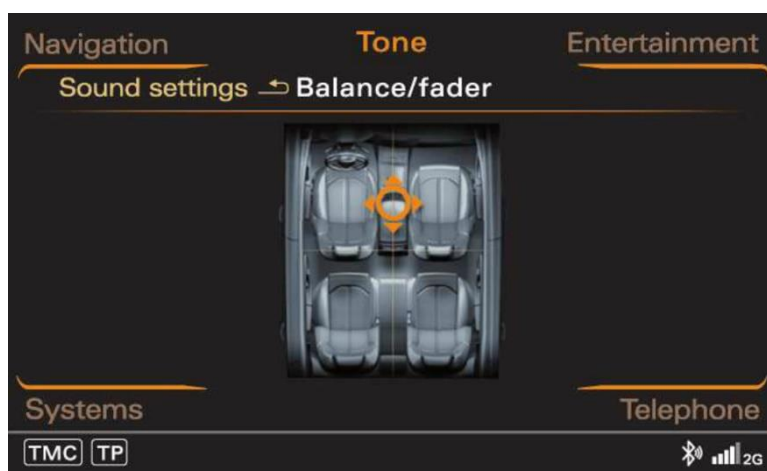
podkapitole. Například reakce rádia na stanici vysílající dopravní hlášení nebo příchozí hovor na spárovaném telefonu.

Při detekci zdrojů se s výhodou využívá možnosti výběru přehrávaných skladeb na externích zařízeních i volby modulační frekvence FM generátoru (3.2.3). Na každém zdroji je přehráván sinový signál o unikátní frekvenci. Algoritmus ověření zdroje při automatickém testu se tedy může omezit na výběr krátkého časového okna a výpočet maxima jeho spektra

4.2.3 Váhy prostorového zvuku

Moderní audio systémy dovolují uživateli přizpůsobit úroveň hlasitosti na jednotlivých kanálech výstupu, tak aby hlasitost audia ve vozidle vyhovovala celé posádce. Možnosti závisí na počtu výstupních audio kanálů. Standardem v moderních vozech bývají čtyř kanálové zesilovače, u kterých je možné přizpůsobit rozložení hlasitosti v kabině v podélném i příčném směru.

Tento parametr se dá poměrně snadno sledovat porovnáním hlasitostí na jednotlivých kanálech. Použitý testovací stav byl připraven na testování čtyř kanálového zvuku.



Obrázek 4.1 - Nastavení prostorového zvuku ve vozech Audi. (zdroj: myaudiworld.sg)

4.2.4 Nastavení ekvalizéru

Aby se výrobci přizpůsobili různým nárokům zákazníku a na charakter zvuku, nabízí většina dnešních systémů možnost upravit si frekvenční odezvu zesilovače pomocí ekvalizéru. Uživatel má tak možnost upravovat zisk zesilovače v různých frekvenčních pásmech. Počet nastavitelných pasem se pak liší podle třídy vozu.

Na pracovišti, na kterém jsem měl možnost pracovat se v tomto ohledu sleduje nastavení pouze na úrovni grafického uživatelského rozhraní a přesné výpočty frekvenčních odezvy se provádějí na jiných specializovaných pracovištích.



Obrázek 4.2 - Nastavení ekvalizéru ve vozech Hyundai. (Zdroj: webmanual.hyundai.com)

4.2.5 Práce se zvukovým signálem v průběhu testů

Na začátku každého testu je spuštěno nahrávání zvuku na všech čtyřech kanálech, tak aby se kdykoliv v průběhu testu mohl odebrat vzorek signálu a provést na něm potřebné operace. Po skončení testu jsou zvuková data uložena do jednoho souboru „wave“ pro každý kanál.

4.3 Navrhovaná zlepšení

V průběhu mého studia zpracování zvuku při automatizovaných testech v rámci semestrálního projektu jsem našel několik oblastí, ve kterých lze nalézt prostor pro vylepšení a rozhodl se tato vylepšení pokusit implementovat v rámci diplomové práce. V následujících řádcích si vyjmenujeme tyto oblasti a důvody, které vedly jejich výběru pro implementaci.

4.3.1 Kontrola změny frekvenční odezvy na nastavení ekvalizéru

V kapitole (4.2.4) jsme se dozvěděli, že kontrola nastavení ekvalizéru probíhá v průběhu automatizovaných testů „audio managementu“ pouze na úrovni uživatelského rozhraní a přesné vyhodnocení frekvenční odezvy probíhá na specializovaných pracovištích. Vzhledem k možnosti vysílat do jednotky prakticky jakýkoliv signál a uložit zaznamenat ho, by bylo možné zpracováním vhodně zvoleného testovacího signálu získat alespoň přibližný průběh frekvenční odezvy zesilovače a odhalit tak případné problémy ještě před podrobným měřením. Touto problematikou se budeme zabývat v dalších kapitolách práce.

4.3.2 Využití zvukových záznamů z testů

Jak bylo řečeno v kapitole (4.2.5), je po každém testu uložen zvukový soubor ve formátu „wave“ na výstupní kanál zesilovače. Tyto soubory jsou archivovány jako prokazatelný materiál pro případné zkoumání nalezených chyb, Nicméně nejsou žádným způsobem dále zpracovávány a je tedy na testerovi, aby v nich hledal problémy poslechem nebo je kontroloval pomocí softwarového nástroje jako například Audacity [36]. Takovýto postup by byl ovšem zdlouhavý a v rámci automatického testování neproveditelný. V případě poslechu se navíc nedá předpokládat, že by člověk vydržel poslouchat monotónní testovací signály a hledal v nich rozdíly nebo problémy.

Proto vznikl nápad vytvořit nástroj, který by po ukončení testu automaticky zpracoval zvukový záznam testů a případně potřeby poskytl testerovi zkontrolovat průběh zvukového výstupu jednotky při testu. Případně znovu překontroloval znovu překontrolovat sledované parametry testu a zvýšit tak jeho přesnost.

4.3.3 Kontinuální kontrola zdrojů

V podkapitole (4.2.2) je popsáno, že se kontrola právě hrajícího zdroje provádí na základě informací získaných z krátkého výřezu signálu v bodě zvoleném tvůrcem testu. Tato metoda bez problému dostahuje pro kontrolu změny zdroje nebo aktivace upozornění (dopravní hlášení, navigace...) za předpokladu je testovací skript napsán přesně. Při využití zvukového záznamu z testu popsaného v minulé podkapitole by bylo možné vytvořit graf závislosti aktivních zdrojů na čase. S jeho pomocí by tester získal kontinuální informace o zdroje v celém průběhu testu, a ne pouze v předem určenou dobu. Toto zlepšení by umožnilo také detekovat případné výpadky audia nebo příliš dlouhé intervaly při změně zdrojů kdykoliv během testování.

4.3.4 Detekce artefaktů

Při zpracování zvukového záznamu testu se nabízí možnost kontrolu nahraného signálu na výskyt rušivých artefaktů, které by mohli ovlivnit výsledky testů. Tímto tématem se podrobněji zabývá následující kapitola.

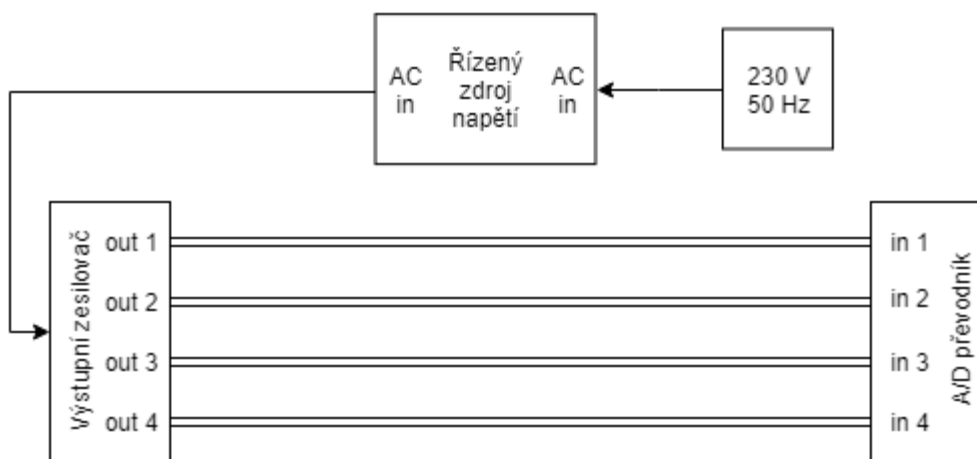
4.4 Studium artefaktů

V předchozí kapitole je vyjádřen záměr doplnit možnosti zpracování signálu při testování o možnost detekovat rušivé artefakty a šum v testovacím signálu. Aby bylo možné zvolit vhodnou metodu detekce bylo nejprve potřeba provést studium testovacího

řetězce, odhalit možné příčiny rušení signálu a zhodnotit, zda se v testovacím signálu opravdu objevují. Následující kapitole si popíšeme jednotlivé předpokládané a zjištěné zdroje artefaktů v testovacím signálu a navrhované metody detekce.

4.4.1 Zapojení pro snímání zvukového výstupu

Prvním sledovaným faktorem byla cesta signálu od zesilovače jednotky do analogově digitálního převodníku použitého k nahrávání signálu (3.2.8).



Obrázek 4.3 - Schéma propojení audio výstupu a AD převodníku

Z plánku zapojení je vidět, že díky přímému propojení není v zapojení mnoho slabých míst kde by mohlo vznikat rušení signálu. Vzhledem k napájení z klasické rozvodové sítě se dá předpokládat rušení na 50 Hz, vzniklé interferencí frekvence střídavého proudu síťového zdroje s testovacím signálem vedeným nestíněnými vodiči. Tento druh zkreslení signálu se dá řešit aplikací číslicových filtrů s vhodně zvolenými koeficienty. Tento problém se často řeší při zpracování EKG signálů, kde je rušení na nízkých frekvencích velmi závažné [23].

4.4.2 Zohlednění typu testovacích signálů

Vzhledem k možnosti zvolit si přehrávaný signál na každém zdroji, není problém zvolit signál jasně definovaný signál s dostatečnou amplitudou. V testu použity sinové funkce. Z tohoto důvodu by u „pevných“ zdrojů jako USB, SD karta nebo připojený telefon přes kabel neměl být prostor k vzniku dalších rušení. Naopak u simulovaného vysílání FM by při nekvalitní přenosu signálu vzduchem nebo při vedení propojovacího kabelu v blízkosti napěťového zdroje mohl vznikat aditivní šum. Problém by také mohl vzniknout při volbě nosné frekvence signálu [24] podobné s některou lokální radiovou stanicí a došlo

by tak k interferencím obou signálů. Tento typ aditivního šumu je možné řešit za pomoci spektrálního odečítání nebo Wienerovou filtrací podobně jako potlačování šumů při rozpoznávání řečového signálu [25].

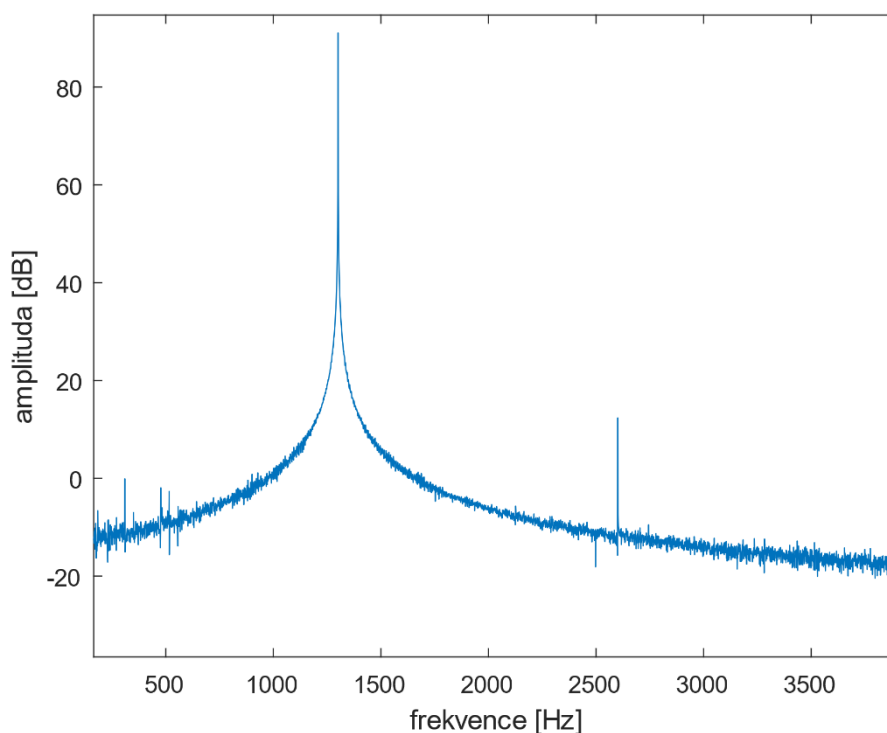
4.4.3 Studium reálného signálu

S ohledem na předem zmíněné předpoklady bylo provedeno studium zvukových záznamů z reálných testů. Jelikož tyto soubory představují hodiny záznamu, Soustředil jsem se na složitější testovací sekvence, při kterých by mohly nastat problémy, jako časté změny zdrojů, ztráty napájením nebo manipulace s nastavením charakteristik zvukového výstupu jednotky.

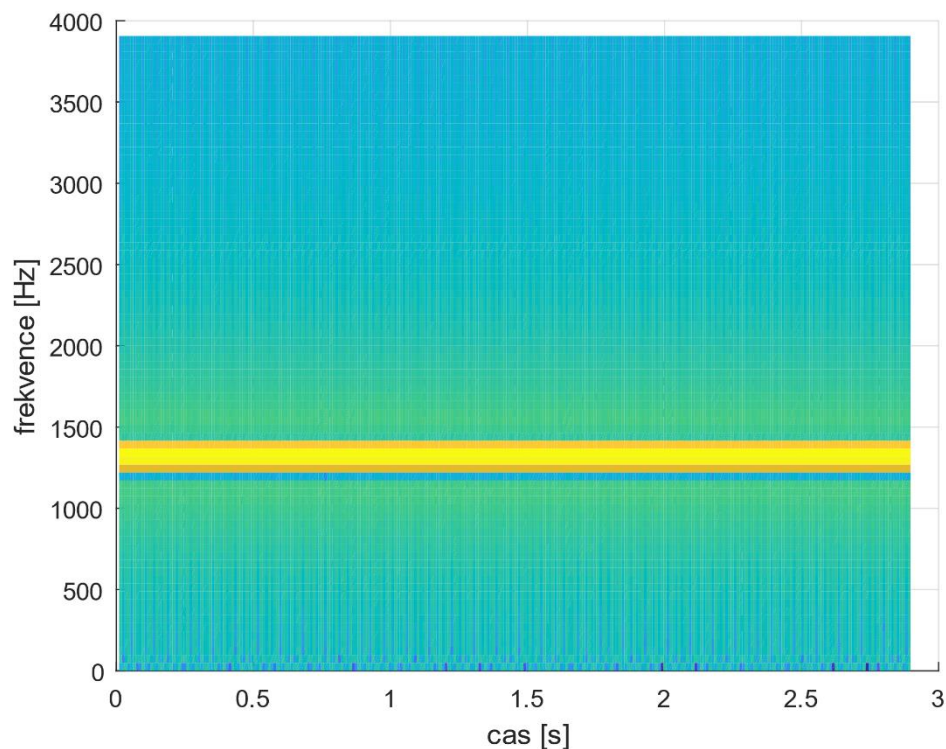
4.4.3.1 Rušení kanálu

Jak bude vidět v následujících kapitolách, ukázalo se, že testovací stav byl z pohledu kvality snímaného signálu navržen velmi dobře a ve většině situací se nepodařilo nalézt závažnější problém s čistotou nahraného signálu.

Na grafech níže je zobrazen vzorek signálu nahraného z reálné jednotky. Zdrojem bylo FM rádio vysílané generátorem na stavu. Ve spektrální oblasti si můžeme všimnout drobného zkreslení zejména nižších frekvencích, jeho amplituda oproti účinnému signálu zanedbatelná. Vliv síťového rušení také není zřejmý.



Obrázek 4.4 - Logaritmické spektrum testovacího signálu FM rádia



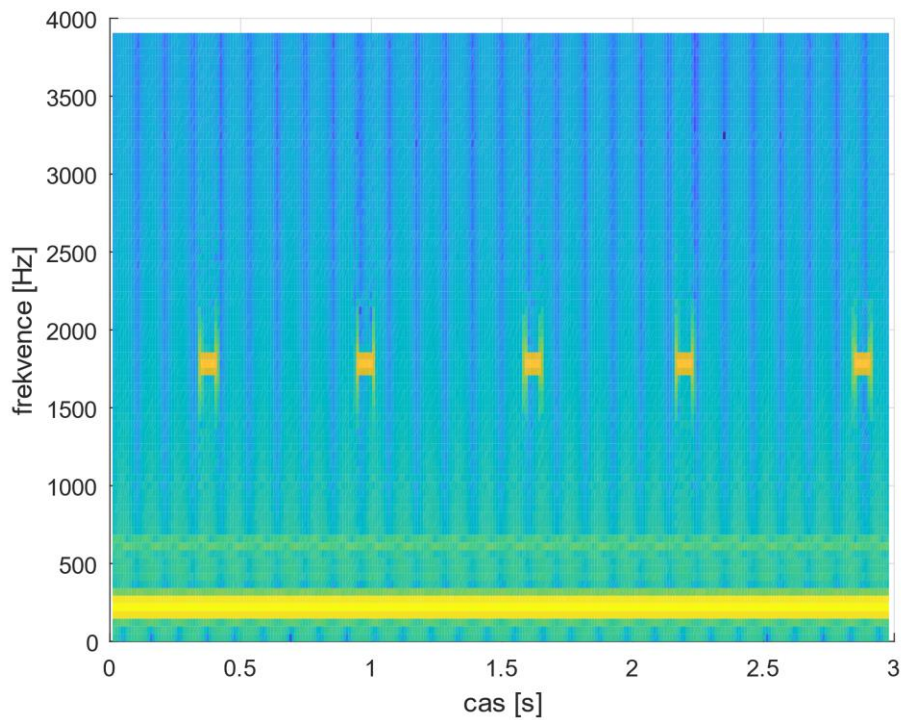
Obrázek 4.5 - Spektrogram testovacího signálu FM rádia

I přesto, že při hledání rušivých elementů v testovacím signálu způsobeným rušením na kanálu nebyl nalezen závažnější problém, nedá se předpokládat stejný výsledek u všech testovacích stavů s jiným zapojením

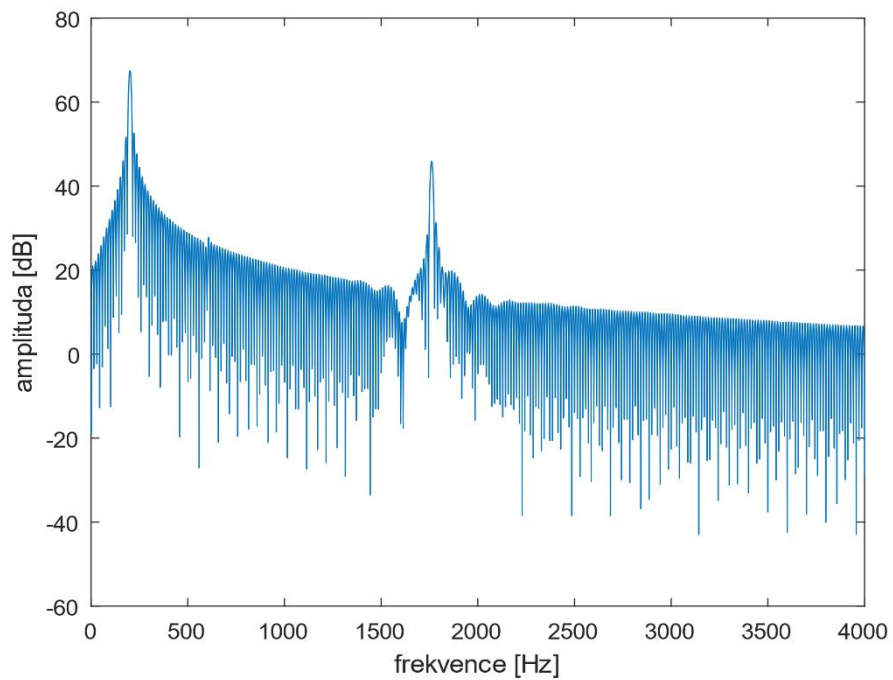
4.4.3.2 Rušení způsobené vnějšími vlivy

Dalším krokem bylo kontrola vlivu vnějších faktorů, jako je ovládaní jednotky a manipulace se zdrojem napětí na kvalitu testovacího signálu. Ukázalo se, že častým zdrojem rušení při testech je tak zvaný „touch tón“. Tedy zvuková odezva jednotky na interakci uživatele s dotekovou obrazovkou. Tato funkce se sice dá v nastavení většiny infotainmentových jednotek deaktivovat, ale zejména v raných částech vývoje toto nastavení nemusí být k dispozici nebo nemusí být persistentní například po restartu nebo ztrátě napájení. Vezmeme-li v úvahu způsob ovládaní jednotky při komponentních testech, tedy simulací této uživatelské interakce, a neznalost vnitřní struktury softwaru jednotky, musíme předpokládat možnost vzniku těchto artefaktů při testování. Možný vliv těchto artefaktů na výsledky testů je znázorněn ve spektrogramu na obrázku (4.8). Na snímku (4.7) je znázorněno logaritmické spektrum signálu v bodě rušení. Můžeme si všimnout

výrazného poklesu poměru signálu ku šumu v poměru s obrázkem (4.5). Je tedy zřejmé že tyto artefakty by měly být detekovány.



Obrázek 4.7 - Spektrogram signálu zatíženého "touch" tóny



Obrázek 4.6 - Logaritmické spektrum signálu v bodě rušení "touch" tónem

4.4.4 Vyhodnocení a návrh metod ke zpracování

Vzhledem k tomu, že v průběhu studia nebyl zjištěn žádný výrazný konstantní šum. Ale spíše lze předpokládat náhodné artefakty způsobené druhem ovládaní jednotek, nebo manipulací s testovacím stavem, bylo nutné zvolit metodu schopnou odhalit široké spektrum artefaktů a rušení.

Při výběru možných metod jsem se inspiroval pracemi zabývajícími se potlačování šumu při rozpoznávání řeči [25], [26], [27]. Na rozdíl od zvýraznění lidského hlasu se v našem případě snažíme oddělit od šumu testovací signál a případné artefakty zaznamenat. Metody zvýraznění řeči většinou předpokládají stacionární charakter aditivního šumu, případné změny jeho spektrálních charakteristik by měly být pomalé ve srovnání se spektrálními charakteristikami řeči [27]. V našem případě sice tento předpoklad nemůžeme vždy splnit, ale můžeme využít stacionarity použitých testovacích signálů. Hlavní účel oddělit účinný signál od rušení zůstává stejný.

Mezi nejčastěji používané metody ve zkoumaných zdrojích patří spektrální odečítání a Wienerova filtrace. Tyto metody jsou podrobněji popsány v následující kapitole. Konkrétní implementace detekce artefaktů je popsána v kapitole číslo pět.

5 Relevantní metody zpracování signálu

V této kapitole jsou postupně popsány všechny komplexnější metody využití nebo zvažované v této práci. Výčet je rozdělen podle řešených problémů. Tato kapitola má čtenáři poskytnout teoretické informace o využitých metodách, pro konkrétní implementaci mohou být metody přizpůsobeny konkrétnímu řešení. Postup při implementaci je popsán v kapitole pět.

5.1 Detekce artefaktů

V této části jsou popsány metody vybrané jako vhodné pro detekci rušivých artefaktů. Důvody k výběru těchto technik jsou popsány v kapitole (4).

5.1.1 Spektrální odečítání

V článku „Popis metody spektrálního odečítání“ [28] je přehledně popsán princip metody spočívá v odečtení spektra šumu od spektra řečového signálu kontaminovaného aditivním šumem, přičemž se předpokládá, že signál šumu není korelován s řečovým signálem. U signálu předpokládáme stacionaritu šumu nebo jeho pomalou proměnlivost v čase. Dále předpokládáme stacionaritu přenosového kanálu ve smyslu neměnnosti spektrálních charakteristik v čase, případně dostatečně pomalých změn těchto charakteristik ve srovnání se změnami spektrálních charakteristik signálu. V případě, že přenosový kanál bude ideální můžeme obraz signálu $Y(\omega, k)$, který obsahuje aditivní šum, popsat rovnicí:

$$Y(\omega, k) = S(\omega, k) + N(\omega, k) \quad (4.1)$$

$S(\omega, k)$ je spektrum čistého signálu nesoucího informaci (běžně řečový signál nebo hudba) a $N(\omega, k)$ je spektrum aditivního šumu. Proměnná k vyjadřuje část spektra daného segmentu. Odhad spektrálních charakteristik čistého signálu $|S(\omega, k)|$ lze provést dle rovnice:

$$|\hat{S}(\omega, k)| = |Y(\omega, k)| - \alpha \frac{1}{I} \sum_{i=0}^{I-1} |N(\omega, i)| = |Y(\omega, k)| - \alpha |\overline{N(\omega)}| \quad (4.1)$$

kde $|Y(\omega, k)|$ je amplitudové spektrum signálu kontaminovaného šumem, $|N(\omega)|$ je amplitudové časově zprůměrované spektrum aditivního šumu a α je parametrem, který udává poměrnou část průměrného spektra šumu, kterou následně odečítáme od $|Y(\omega, k)|$. Odhad lze také provést ve výkonové oblasti dle rovnice.

$$|\hat{S}(\omega, k)|^2 = |Y(\omega, k)|^2 - \alpha |\overline{N(\omega)}|^2 \quad (4.2)$$

kde $|S(\omega, k)|^2$ je výkonové spektrum odhadovaného řečového signálu $|Y(\omega, k)|^2$, je výkonové spektrum signálu obsahujícího aditivní šum a $|N(\omega)|^2$ je časově zprůměrované výkonové spektrum aditivního šumu.

5.1.2 Wienerova filtrace

Metoda Wienerovy filtrace se hojně používá při zvýraznění řečového signálu nebo při rekonstrukci obrazových dat zatížených šumem [27], [29]. Je specifická tím, že klade požadavek stacionarity na šum i signál nesoucí informaci, který chceme zvýraznit. Návrh Wienerova filtru vychází ze střední kvadratické chyby mezi originálním a odhadem tohoto signálu. Odhad původního signálu je výstupem Wienerova filtru, jak se píše ve zdroji [30], ze kterého bylo čerpáno při následujícím popisu metody. Nejobecnější tvar Wienerova filtru popisuje lze vyjádřit rovnicí:

$$M(\omega) = \frac{S_{XY}(\omega)}{S_{YY}(\omega)} \quad (5.4)$$

kde $M(\omega)$ je frekvenční charakteristika restauračního filtru, $S_{xy}(\omega)$ vyjadřuje vzájemné spektrum mezi originálním a zkresleným signálem a $S_{yy}(\omega)$ je výkonové spektrum zkresleného signálu. Rovnice (5.5) představuje součin inverzního filtru a Wienerova korekčního faktoru, který nabývá pouze hodnot v intervalu $< 0, 1 >$. Cílem použití Wienerova korekčního faktoru je snížit amplitudový přenos v daných frekvenčních oblastech.

$$M(\omega) = \frac{1}{G(\omega)} \frac{|G(\omega,)|^2}{|G(\omega,)|^2 + \frac{S_{vv}(\omega)}{S_{xx}(\omega)}} \quad (5.5)$$

kde $G(\omega)$ je frekvenční charakteristika odpovídající zkreslení a její převrácená hodnota odpovídá inverznímu filtru. $S_{vv}(\omega)$ je dále výkonové spektrum šumu a $S_{xx}(\omega)$ je výkonové spektrum původního signálu.

Zmíněné rovnice předpokládají přístup k originálnímu signálu. Pro aplikace, ve kterých není možný originální signál získat je vhodný upravený tvar Wienerova filtru využívající pouze členy, jejichž hodnoty lze získat měřením.

$$M(\omega) = \frac{1}{G(\omega)} \frac{S_{YY}(\omega) - S_{vv}(\omega)}{S_{YY}(\omega)} \quad (5.6)$$

5.2 Frekvenční odezva systému

Měření přenosových funkcí a s nimi spojených impulsových odezev je jednou z nejčastěji využívaných technik ve všech oblastech akustiky. Ať už při zjišťování doby dozvuku místnosti v prostorové akustice, měření přenosových parametrů telekomunikačních řetězců ve studiích i u satelitní komunikace, frekvenčních vlastností izolačních materiálů ve stavebnictví nebo frekvenční odezvy při vývoji reproduktorů a zesilovačů. V naší práci se soustředíme na získání frekvenční odezvy zesilovače autorádia. Frekvenční odezva se dá obecně popsat jako Fourierův obraz Impulsové odezvy systému. [34]

$$FR = FT\{IR\} \quad (5.7)$$

Metod získávání impulsových odezev je mnoho v následující části práce si přiblížíme několik nejpoužívanějších v závislosti na druhu měřícího signálu.

5.2.1 Sine sweep

Anglickým názvem Sine sweep je označován přeladovaný sinový signál, jehož okamžitá frekvence $f(t)$ roste v určitém intervalu frekvencí buď lineárně, nebo logaritmicky. V této práci je využívám logaritmicky přeladovaný sinus. Oproti měření jiným metodám má tato metoda lepší imunitu proti harmonickému zkreslení. Produkty harmonického zkreslení mohou být z výsledné impulsní odezvy snadno izolovány. Dále jsou tato měření dobře odolná vůči nedokonalé časové synchronizaci měřeného systému, proto se dobře hodí ke zjištění odezev analogových zařízení nebo při měření v prostředí s výraznějším rušením. Bližší informace k měření touto metodou lze získat například z článku „Transfer-Function Measurement with Sweeps” z roku 2001. [34]

Sine sweep signál je vyjádřen sinovou funkcí, která v daném intervalu od f_{min} do f_{max} zvyšuje svou frekvenci. V tomto intervalu má lineárně přeladovaný signál bílé (ploché) spektrum, logaritmicly přeladovaný růžové (sklon PSD -3 dB/okt., s výjimkou nejnižších frekvencí). Pro logaritmicly přeladovaný sinus platí vztah:

$$x(t) = A \sin \left(\frac{2\pi f_{min} T_p}{\ln \frac{f_{max}}{f_{min}}} \left(\left(\frac{f_{max}}{f_{min}} \right)^{\frac{t}{T_p}} - 1 \right) \right) \quad (5.8)$$

Kde A je amplituda signálu, t je čas, T_p je perioda signálu a f_{min} a f_{max} jsou okraje frekvenčního intervalu. Pro výstup soustavy platí vztah:

$$y(t) = x(t) * h(t) \quad (5.9)$$

kde $h(t)$ je impulsová odezva soustavy kterou lze získat konvolucí s inverzním filtrem, který má tyto vlastnosti:

$$x(t) * f(t) = \delta(t) \quad (5.10)$$

$$X(\omega) * F(\omega) = 1 \quad (5.11)$$

Tedy výsledkem konvoluce vstupního sine sweep signálu s inverzním filtrem je Dirakov impuls $\delta(t)$ a vynásobením spekter vstupního signálu a inverzního filtru bude 1. Pro splnění těchto podmínek musí mít inverzní filtr tvar:

$$f(t) = x(-t)e^{-\alpha t} \quad (5.11)$$

$$\alpha = \frac{0.3 \ln(10) \log_2 \left(\frac{f_{max}}{f_{min}} \right)}{T_p} \quad (5.11)$$

Impulsovou, respektive frekvenční odezvu systému lze tedy získat:

$$h(t) = y(t) * f(t) \quad (5.12)$$

$$H(\omega) = Y(\omega) \cdot F(\omega) \quad (5.13)$$

Bližší informace ke generování sine sweep signálů, metodách jejich použití a odvození vztahu pro inverzní filtr lze nalézt v publikaci „Zpracování akustických signálů“ [31].

5.2.2 Segmentace signálu sine sweep

Máme-li širokopásmový signál jako například logaritmičticky přeladovaný sinus popsaný v předchozí kapitole rovnicí (5.8). Můžeme využít jeho konstantního spektra v rozmezí od jeho počáteční frekvence f_{min} až f_{max} :

$$S(\omega) = konst; \text{ pro } f_{min} \leq \omega \leq f_{max} \quad (5.14)$$

Za předpokladu konstantní amplitudy signálu a logaritmičtějšího stoupání frekvenci signálu podle rovnice (5.8) je efektivní hodnota signálu podle [31]:

$$X_{RMS} = \sqrt{\frac{1}{T} \int_0^T x(t) dt} \quad (5.15)$$

Kde T interval měření signálu. Při nasegmentování signálu na stejně dlouhé intervaly $T = konst.$ stejná pro každý měřený interval. Pokud ale předpokládáme průchod signálu lineární soustavou podle rovnice (5.9), bude jeho spektrum odpovídat rovnici:

$$Y(\omega) = H(\omega) \cdot X(\omega) \quad (5.16)$$

Kde $Y(\omega)$ je spektrum výsledného signálu a $H(\omega)$ přenosová funkce soustavy ve frekvenční oblasti. Za předpokladu platnosti rovnice (5.14) napsat že:

$$Y(\omega) \sim H(\omega) \quad (5.17)$$

Hodnoty Y_{RMS} pro stejně dlouhé segmenty výstupního signálu v časové oblasti $\langle y(t), y(t + T) \rangle$ se budou rovnat průměru hodnot frekvenční odezvy $H(\omega)$ v intervalu $\langle f_{i1}, f_{i2} \rangle$ kde f_{i1} a f_{i2} jsou okamžité hodnoty frekvenci signálu podle rovnice (5.8).v časech t a $t + T$.

5.2.3 MLS

Měření přenosových vlastností akustických soustav pomocí pseudonáhodných signálů (MLS) se v současné době využívá pro mnoho aplikací. Jako je měření

přenosových funkcí reproduktorů nebo sluchátek, měření doby dozvuku místnosti, měření přenosových parametrů telekomunikačních řetězců ve studiích i u satelitní komunikace. V porovnání s jinými metodami, jako je Sine Sweep využívající přeladovaný sinus signál, má MLS výhodu v podobě větší odolnosti vůči rušení, ale nebere v úvahu nelinearity. Podrobný princip metody a vlastnosti MLS uvedené v této práci jsou podrobně popsány v publikaci „Zpracování akustických signálů“. [31]

5.2.3.1 Princip

Binární MLS signál je periodická dvouúrovňová pseudonáhodná sekvence délky $L = 2^N - 1$, kde N je celé nezáporné číslo. Signál je tvořen posuvným registrem se zpětnou vazbou a počtem bitů rovným řádu generované posloupnosti N . Výsledkem je tedy pseudonáhodná posloupnost „0“ a „1“. Autokorelační funkce má velmi důležitou vlastnost a sice tu, že je rovna jednotkovému pulsu:

$$\begin{aligned} R_{xx}[n] &= x[n] \oplus x[n] \\ &= \frac{1}{L+1} \sum_{k=0}^{L-1} x[k] \cdot x[k+n] = \delta[n] - \frac{1}{L-1} \end{aligned} \quad (5.2.1)$$

Výkonové spektrum MLS signálu $S_{xx}[k]$ je popsáno pomocí diskrétní Fourierovy transformace autokorelační funkce $R_{xx}[n]$.

$$\begin{aligned} S_{xx}[k] &= DFT\{R_{xx}[n]\}, \quad S_{xx}[0] = \frac{1}{L}, \quad S_{xx}[k] = \frac{L+1}{L}, \quad k \\ &\neq 0 \end{aligned} \quad (5.2.2)$$

Z rovnice (3.5) je zřejmé, že výkon spektra MLS sekvence je $L+1$ krát vyšší než výkon samotného jednotkového pulsu, což potvrzuje vyšší odolnosti vůči šumu metody MLS. Signál vzniklý průchodem periodického MLS signálu systémem s periodickou impulsovou odezvou $h[n]$ je popsán takto: "

$$y[n] = x[n] \oplus h[n] = \frac{1}{L+1} \sum_{k=0}^{L-1} x[k] \cdot h[n-k] \quad (5.2.3)$$

Korelační funkce $R_{xy}[n]$ mezi vstupní a výstupní signálem lze vyjádřit následovně:

$$R_{xy}[n] = x[n] \oplus y[n] = x[n] \oplus \{x[n] \oplus h[n]\} = R_{xx}[n] \oplus h[n] \quad (5.2.4)$$

Dosazením rovnice (5.2.4) za $R_{xx}[n]$ můžeme určit impulsovou odezvu systému:

$$R_{xy}[n] = \left[\delta[n] - \frac{1}{L-1} \right] \oplus h[n] = h[n] - \frac{1}{L+1} \sum_{k=0}^{L-1} h[k] \quad (5.2.5)$$

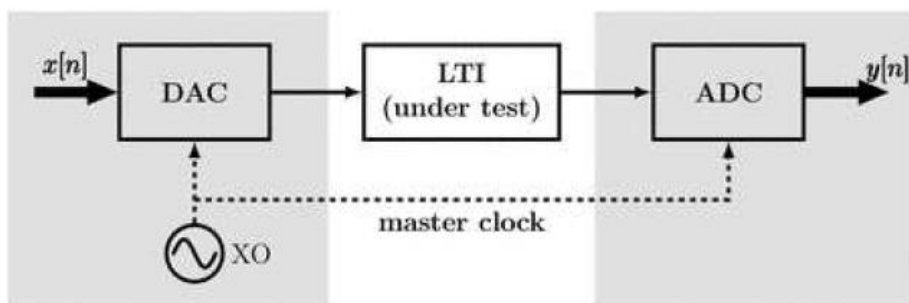
$$h[n] = R_{xy}[n] + \frac{1}{L+1} \sum_{k=0}^{L-1} h[k] \cong R_{xy}[n] \quad (5.2.6)$$

Z poslední rovnici vyplívá, že korelační funkce vstupního a výstupního MLS signálu $R_{xy}[n]$ je rovna impulsové odezvě systému $h(n)$ liší se pouze stejnosměrnou složkou, kterou je možné potlačit při velkém počtu vzorků MLS signálu. Impulsová odezva systému je tedy získána buzením soustavy sekvencí MLS signálů a zaznamenáním odezvy na výstupu $y[n]$. Po zprůměrování všech naměřených výstupů systému je impulsová odezva $h[n]$ rovna korelační funkci $R_{xx}[n]$. Frekvenční odezvu systému je pak možné určit pomocí Fourierovy transformace:

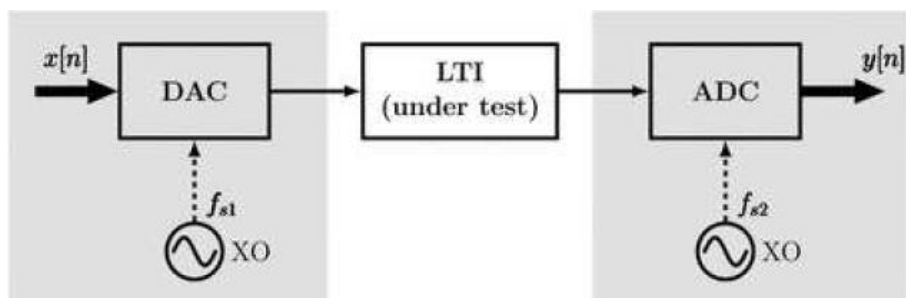
$$H[\Omega] = DFT\{h[n]\} \quad (5.2.7)$$

5.2.3.2 Synchronní a asynchronní měření

Měření impulsové odezvy lineárního systému se ve většině případů provádí v synchronním módu, kdy je použito pro generování a vyhodnocení signálu stejné zařízení. Například při měření v prostorové akustice, jak je popsáno v článku [32]. Jsou ale případy, ve kterých není možné zajistit dokonalou synchronizaci generátoru a přijímače testovacího signálu. V typickém digitálním audiosystému je pro generování či příjem signálu použitý analogově digitální, respektive digitální analogový převodník vybavenými krystalovým oscilátorem, který má za úkol generovat vzorkovací frekvenci f_s . Žádný oscilátor ale nedokáže vygenerovat naprosto přesnou vzorkovací frekvenci. Z tohoto důvodu se vzorkovací frekvence rozchází i když jsou na generátoru i přijímači nastaveny na stejnou hodnotu. Tato chyba v taktovací frekvenci způsobuje rozcházení vzorků vysílaného a přijímaného signálu v čase. I když může jít o velmi malé rozdíly v setinách procent, při



Obrázek 5.2 Měření v synchronním režimu. DAC – DA převodník. ADC – AD převodník, XO – krystalový oscilátor, LTI – testovaná lineární soustava



Obrázek 5.1 – Měření v asynchronním režimu. DAC – DA převodník. ADC – AD převodník, XO – krystalový oscilátor, LTI – testovaná lineární soustava

vyhodnocování MLS sekvencí mohou vznikat velké nepřesnosti, jak je popsáno v článku [33], který navrhuje metodu eliminace těchto problémů popsanou dále.

Pokud máme MLS signál o délce L převedený ideálním AD převodníkem o vzorkovací frekvenci f_{s1} jeho perioda T se bude rovnat:

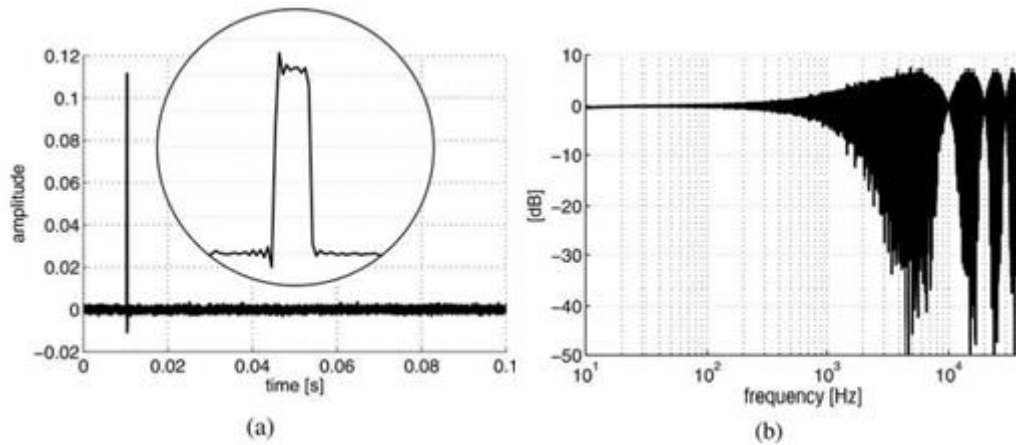
$$T = \frac{L}{F_{s1}} \quad (5.2.8)$$

Rekonstruovaný analogový signál $x(t)$ je poté navzorkován ideálním AD převodníkem s vzorkovací frekvencí f_{s2} zpět na diskretní signál $y[n]$. Vzhledem k tomu, že délka vysílané i přijaté sekvence v čase je stejná, tak pokud platí že $f_{s2} > f_{s1}$, je délka periody T_2 větší o ΔL podle rovnice:

$$T = \frac{L + \Delta L}{F_{s2}} \quad (5.2.9)$$

z toho vyplývá, že rozdíl v počtu vzorků ΔL bude:

$$\Delta L = L \left(\frac{F_{s2}}{F_{s1}} - 1 \right) \quad (5.2.10)$$

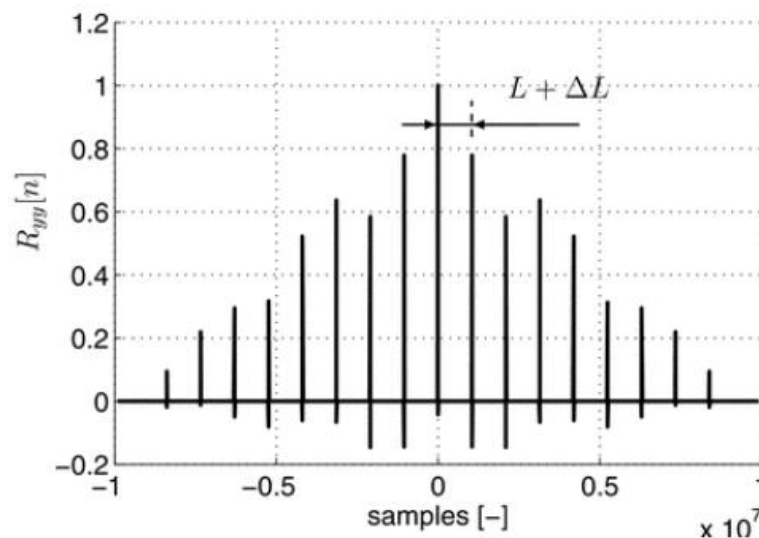


Obrázek 5.3 - Impulsová a frekvenční odezva měřená v asynchronním režimu [33]

Při konvoluci signálů $x[n]$ a $y[n]$ je po tom delší signál zkrácen na délku L . Takto získaná impulsová a následně i frekvenční odezva měřeného systému je zatížena značnou chybou. Na obrázku (5.2) je vidět impulsová (a) a frekvenční (b) odezva získaná měřením v asynchronním režimu při vzorkovací frekvenci obou převodníků nastavené na 96 kHz, zpracovaná ve vědeckém článku [33].

5.2.3.3 Korekce vlivu rozdílných vzorkovacích frekvencí

V první řadě je třeba zjistit o kolik vzorků se liší generovaný a přijatý signál. Pokud se náš budící signál skládá alespoň ze tří MLS period, můžeme první periodu přijatého signálu odstranit pro eliminaci zkreslení [34]. Provedeme-li autokorelační funkci zbývajících dvou period přijatého signálu $y[n]$, získáme funkci $R_{yy}[n]$ zobrazenou na



Obrázek 5.4 - Autokorelační funkce $R_{yy}[n]$ přijatého signálu $y[n]$. [33]

obrázku (5.3). Vzdálenost mezi prvními dvěma maximy autokorelační funkce se bude rovnat rozdílu v počtu vzorků ΔL .

Daný rozdíl je třeba vzít v úvahu v dalším kroku výpočtu, kterým je cyklická korelace signálů $x[n]$ a $y[n]$. Tento krok je možné provést v časové i ve frekvenční rovině. Při práci v čase je třeba kratší ze dvou signálů interpolovat z délky L na $L + \Delta L$, aby byl počet vzorků obou signálů shodný, a poté provést cyklickou korelaci. Rozdíl ΔL je platný i pro spektra obou signálů. Jejich tvar je popsán následujícími rovnicemi:

$$X[k] = \sum_{n=0}^{L-1} x[n] e^{-j \frac{2\pi kn}{L}} \quad (5.2.11)$$

$$Y[k] = \sum_{n=0}^{L+\Delta L-1} y[n] e^{-j \frac{2\pi kn}{L+\Delta L}} \quad (5.2.12)$$

Pro srovnání délky obou spekter můžeme ΔL nul ke spektru signálu s nižší vzorkovací frekvencí nebo odebrat ΔL vzorků ze spektra delšího signálu podle rovnice (5.2.13).

$$Y[k] = \begin{cases} Y[k], & 0 \leq k \leq \frac{L-1}{2} \\ Y^*[L-k], & \frac{L-1}{2} \leq k \leq L-1 \end{cases} \quad (5.2.13)$$

Poté lze cyklickou korelaci provést přímo ve frekvenční rovině vynásobením spekter $X[k]$ a $Y[k]$.

6 Implementace algoritmů a testy

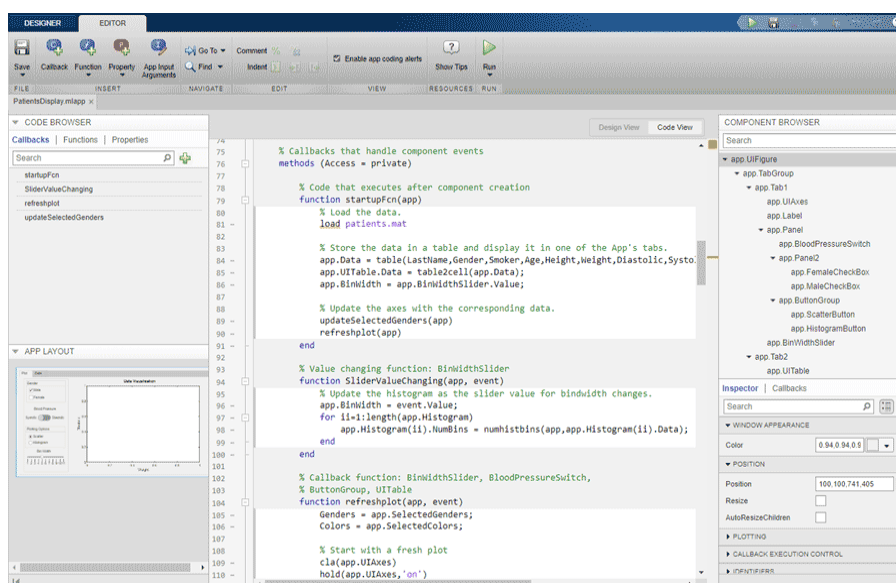
Jako výstup této práce byla vytvořena aplikace v prostředí Matlab App Designer, jejichž hlavním účelem je demonstrovat funkčnost navrhovaná zlepšení zpracování signálu při testech zvukového výstupu infotainmentových jednotek a ověřit funkčnost zvolených algoritmů a implementace na reálných datech získaných při testování. V následující kapitole si postupně představíme zvolené algoritmy pro realizaci jednotlivých bodů, důvody, které vedly k jejich výběru a konkrétní implementaci. Následně se seznámíme s funkcí a ovládáním vytvořené aplikace.

Hlavními implementovanými body jsou:

- Nástroj pro vizuální kontrolu záznamu testu
- Kontinuální kontrola aktivních zdrojů
- Detekce rušivých artefaktů
- Nástroj pro získání frekvenční odezvy

6.1 Vývojové prostředí

Vzhledem k záměru sdužit jednotlivé implementované algoritmy do uceleného testovacího nástroje, byl vybrán jako vývojové prostředí Matlab App Designer [35]. Tento nástroj je součástí vývojového prostředí Matlab, určeného pro vývoj aplikací a grafickým uživatelským rozhraním. Na rozdíl od předchozích nástrojů pro vytváření uživatelského rozhraní jako Matlab GUIDE nabízí širší možnosti strukturování kódu a integrovaný ladící

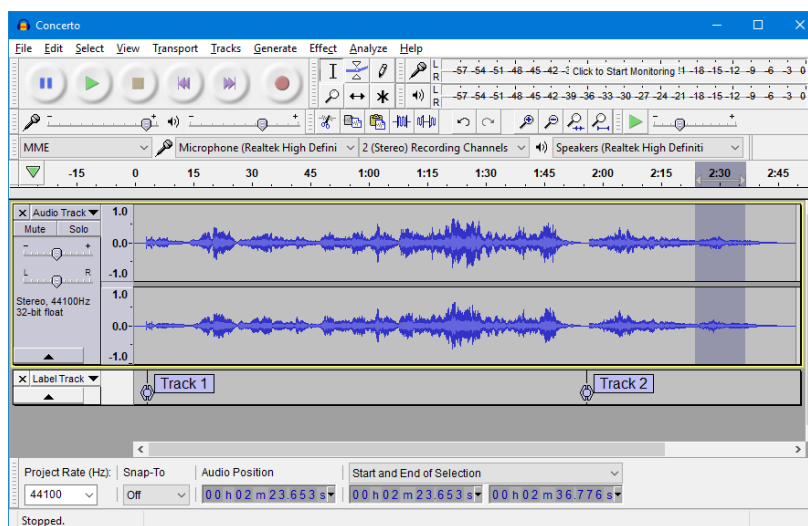


Obrázek 6.1 - Prostor Matlab App Designer. (Zdroj: mathworks.com)

nástroj pro lepší kontrolu na vytvářeném kódem. Další informace jsou k dispozici na stránkách výrobce [35].

6.2 Použitá data

K vývoji a testování jednotlivých algoritmů byla využita reálná data z infotainmentové jednotky Škoda Swing získaná za pomoci nahrávacího zařízení na reálném testovacím stavu určenému k automatizovanému testování (3.2). Jelikož v rámci práce nebyla řešena problematika vícekanálového zvuku, pracuje se v jeden moment vždy s daty z jednoho kanálu výstupu. Data byla nahrávána v nekomprimovaném formátu „wave“ pro minimalizaci zkreslení. Z nahraných dat byly následně vybrány pasáže vhodné demonstraci funkcí aplikace. Úpravy byly provedeny za pomoci volně dostupného nástroje Audacity



Obrázek 6.2 - Prostředí nástroje Audacity. (Zdroj: audacityteam.org)

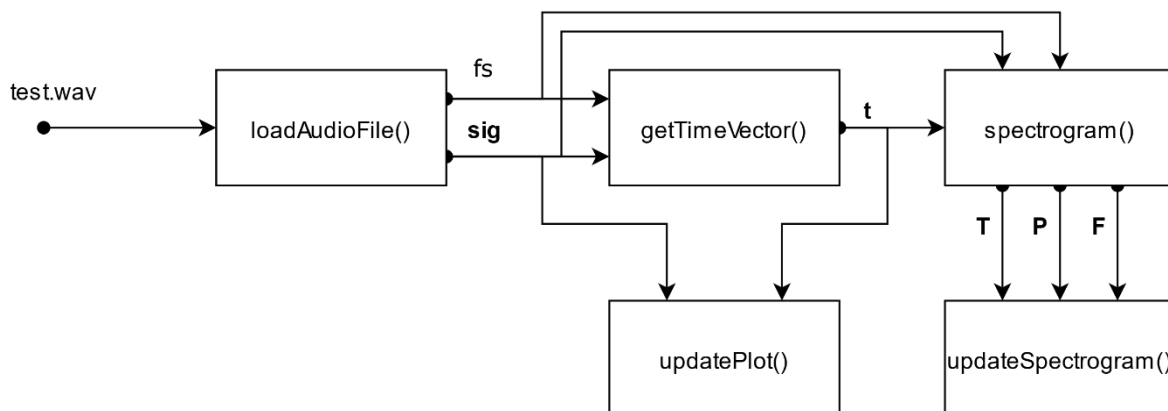
[36].

6.3 Vizuální kontrola záznamu testu

Jedním z prvních důvodů pro vytvoření nových funkcionalit zpracování zvuku na testovací stavu bylo nevyužití zvukových záznamů z testu (4.3.2). Cílem bylo vytvořit nástroj který by uživateli dovolil zobrazit zvukový záznam vizuálně v časové i frekvenční oblasti. Tohoto lze v prostředí Matlab dosáhnout poměrně jednoduše, stejně tak jako zakomponovat tyto obrazové informace do uživatelského prostředí aplikace. Data získané v této fázi zpracování budou ale důležité pro další funkce aplikace.

6.3.1 Algoritmus

Nejprve je načten soubor ve formátu „wave“, ze kterého jsou pomocí funkcí *loadAudioFile()*, *getTimeVector* a *spectrogram()* získány potřebná data. Následně je v blocích *updatePlot()* a *updateSpectrogram()* uživateli vizuálně zobrazen signál v časové



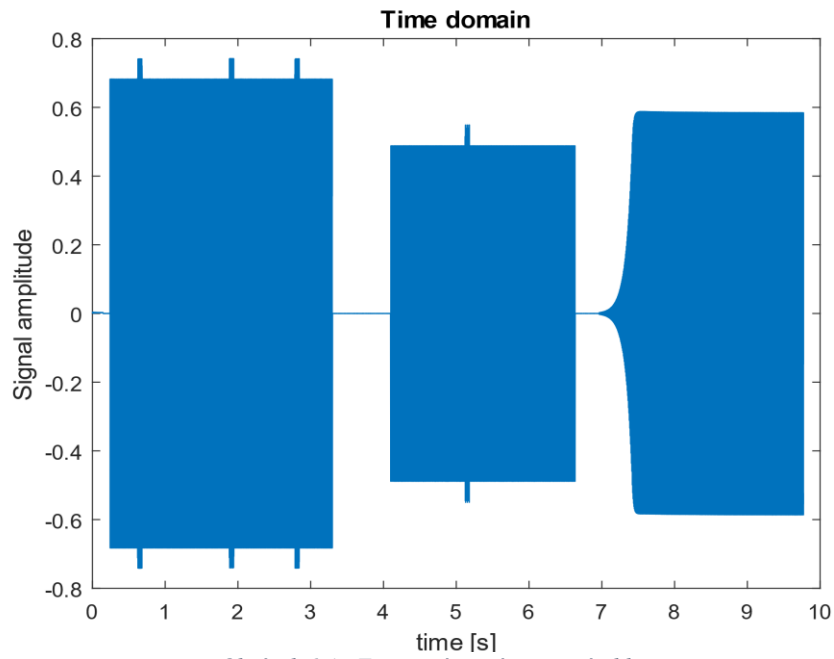
Obrázek 6.3 - Blokový diagram zobrazení audio záznamu

i frekvenční oblasti. Podle následujícího blokového schématu:

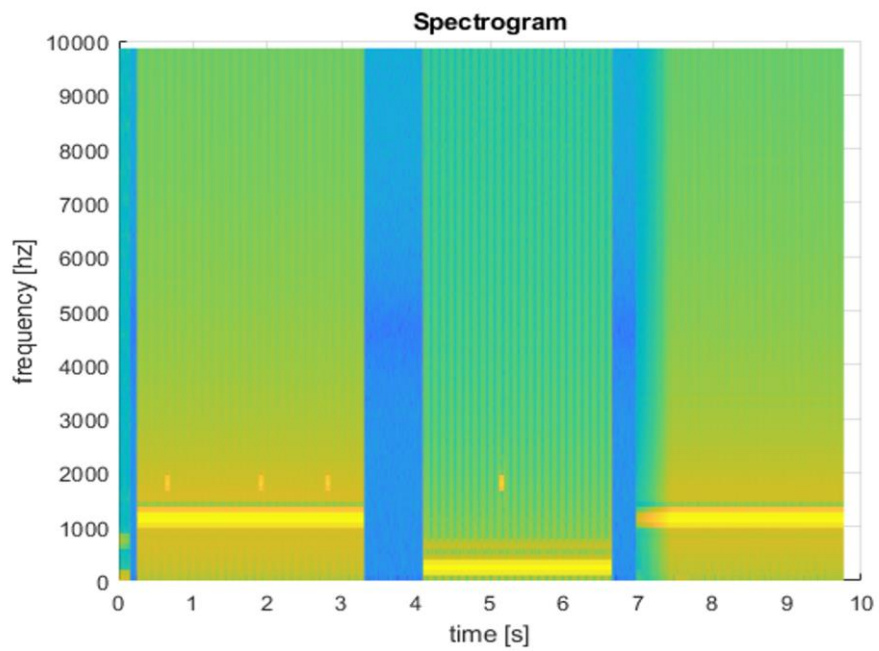
- fs – vzorkovací frekvence získaného signálu
- sig – vektor vzorků získaného signálu
- t – vektor časových značek jednotlivých vzorků
- P – matice vektorů spektrálních výkonových hustot (PSD) pro okno signálu
- F – vektor značek pro frekvenční osu spektrogramu
- T – vektor značek pro časovou osu spektrogramu

Spektrogram je implicitně počítán s délkou okna 512 vzorků a polovičním přesahem. Data získaná v této části jsou dále využívána dalšími funkcemi aplikace.

6.3.2 Ukázka výsledků



Obrázek 6.4 - Testovací signál v časové oblasti



Obrázek 6.5 - Spektrogram testovacího signálu

6.4 Kontrola zdrojů

Dalším bodem k implementaci byla kontinuální kontrola aktivních zdrojů v průběhu testu (3.3.3.3). V tomto bodě byl využit fakt, že tester má plnou kontrolu na přehrávanými soubory z externích zdrojů hudby i na modulační frekvenci FM rádia. Pro každý zdroj byl tedy vybrán sinový signál o unikátní frekvenci.

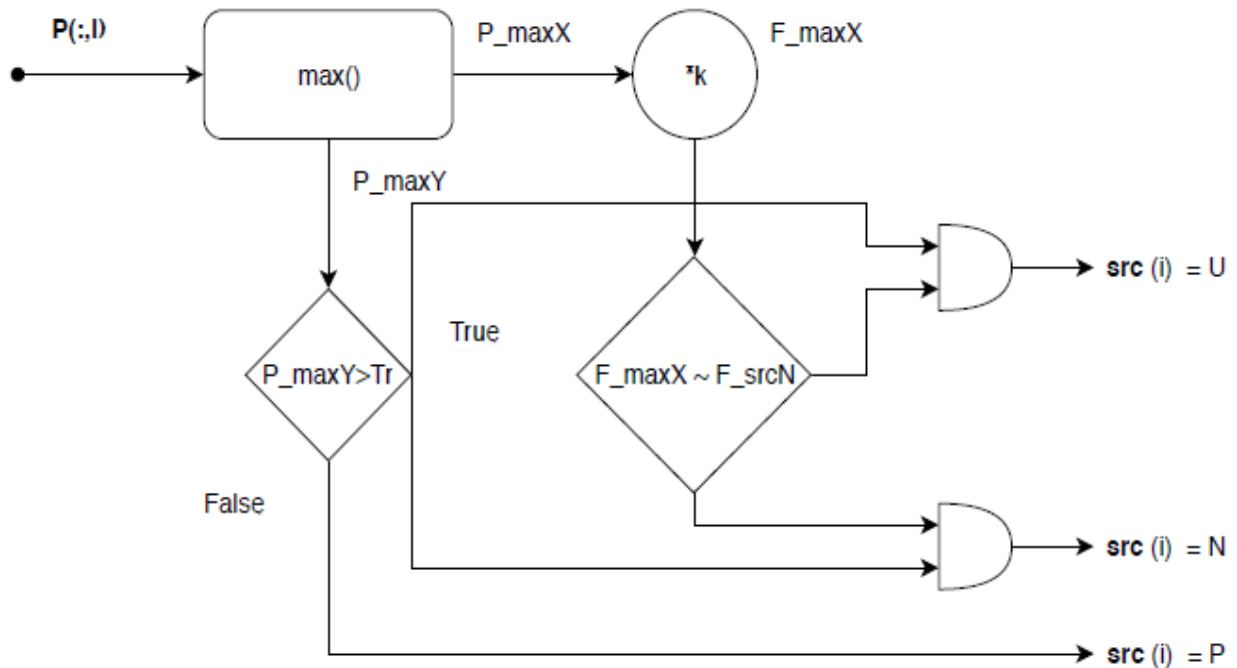
Zdroj	Frekvence [Hz]
FM rádio	1200
Aux	800
USB	1100
Bluetooth audio	300
Smartlink	1950

Tabulka 6.1 - Tabulka identifikačních frekvencí zdrojů

Tyto testovací signály byly vytvořeny pomocí nástroje Audacity a jsou k dispozici v datové příloze.

6.4.1 Algoritmus

Vzhledem k tomu, že předchozího kroku máme signál kompletně popsáný i ve frekvenční oblasti maticí \mathbf{P} můžeme porovnat pozici maxima spektrální výkonové hustoty pro každý časový krok spektrogramu a porovnat ho s identifikačními frekvencemi jednotlivých zdrojů. Pozice maxima musí být nejprve korigována koeficientem k daným frekvenčním rozlišením spektrogramu.

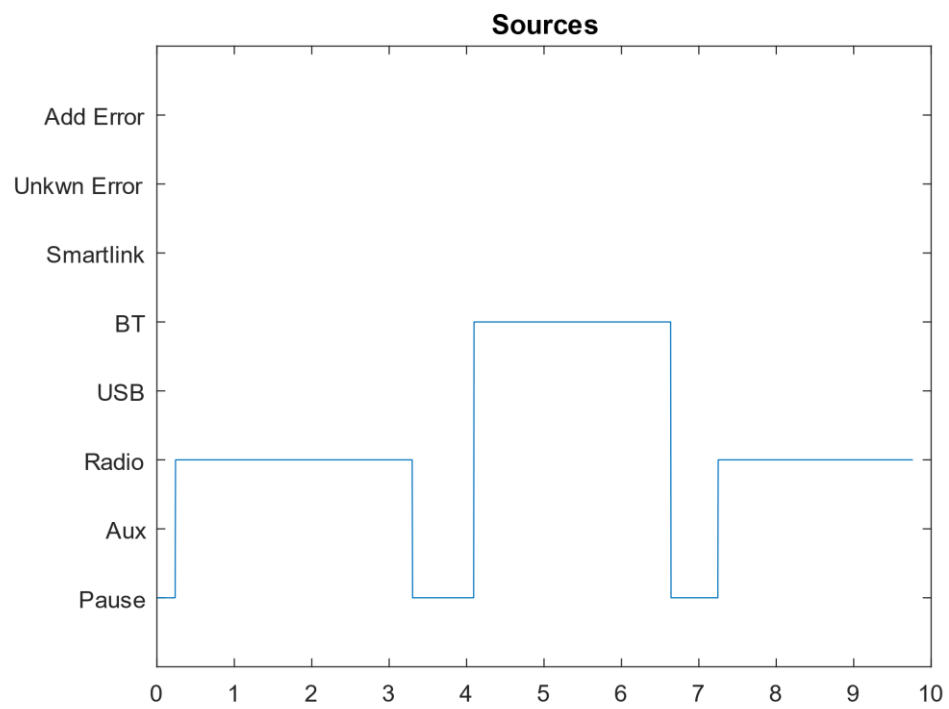


Tabulka 6.2 - Algoritmus výběru zdroje

- $P(:,i)$ – vektor spektrální výkonové hustoty pro dané časové okno
- P_{maxX} – pozice maxima PSD
- P_{maxY} – hodnota maxima PSD
- k – korekční koeficient $k = fs/wlen$; když $wlen$ je délka okna
- F_{maxX} – korigovaná pozice maxima PSD v Hz
- F_{srcN} – identifikační frekvence zdroje N
- Tr – práh aktivity zdroje
- U – neznámý zdroj
- N – zdroj N
- P – nezjištěn aktivní zdroj, pauza.
- src – vektor zdrojů

V tomto bodě máme pro každý bod v čase testu informaci tom, zda byla úroveň amplitudy zvukového výstupu dostačená pro testy, který ze zdrojů byl aktivní, případně zda nepřijímáme na výstupu neočekávaný signál. Tyto data využijeme v další kapitole.

6.4.2 Ukázka výsledků



Obrázek 6.6 - Graf aktivních zdrojů v závislosti na čase

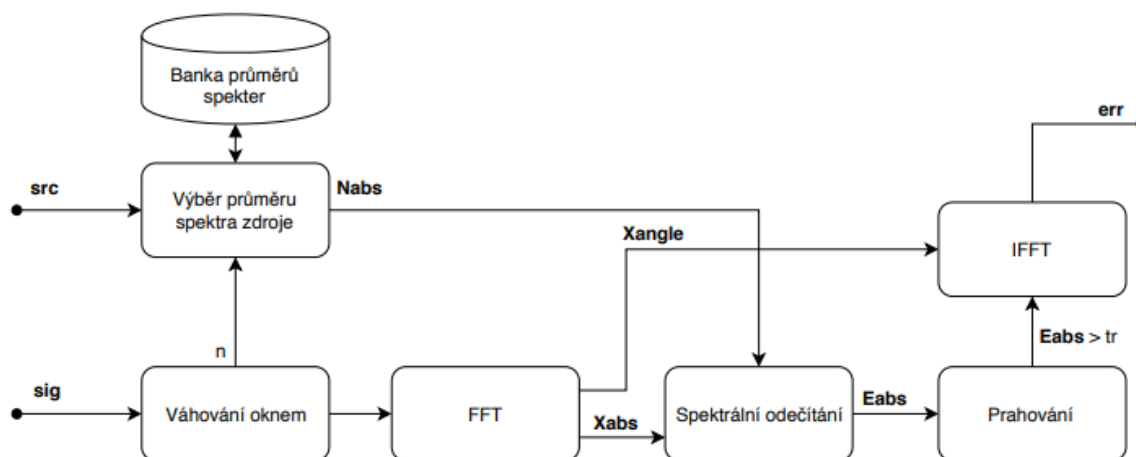
6.5 Detekce artefaktů

U návrhu algoritmu detekce artefaktů bylo využito opět informací získaných z předešlých funkcí, informací o aktivních zdrojích v čase a charakteru signálu na nich přehrávaném. V kapitole (3.4.4), kde je řešen výběr možných metod detekce, byly vybrány jak dvě nejpřípustnější metody Wienerova filtrace (4.1.2) a spektrální odečítání (4.1.1). Důvodem k jejich výběru bylo časté využití při potlačování šumu u zpracování řečového signálu.

Obě metody jsou si principem podobné a jejich rozdíly jsou přehledně popsány v kapitole „Porovnání spektrálního odčítání s Wienerovou filtrací“ v práci [25]. Parametrem rozhodujícím pro naše využití byla skutečnost, že zatímco obě metody využívají časově zprůměrovaný odhad výkonového spektra šumu, využívá Wienerova filtrace časově průměrovaného výkonového spektra odhadu užitečného signálu obsahujícího aditivní šum a spektrální odečítání využívá okamžitého výkonového spektra originálního signálu [27], [30]. Na základě těchto poznatků bylo přikročeno k mírně upravenému algoritmu spektrálního odečítání. S využitím znalosti testovacího signálu lze průměrovaný odhad jeho výkonového spektra považovat za odhad „šumu“, ten poté odečíst od aktuálního spektra signálu a získat tím rozdílové výkonové spektrum chybového signálu. Algoritmus je podrobněji popsán pomocí blokového schématu (5.7).

6.5.1 Algoritmus

Prvním krokem tedy byla tvorba banky průměrovaných odhadů výkonového spektra pro každý testovací signál. Odhad byl vždy vytvořen z několika sekund čistého testovacího signálu nahraného na testovacím stavu. Banka je uložena ve formátu matice vektorů



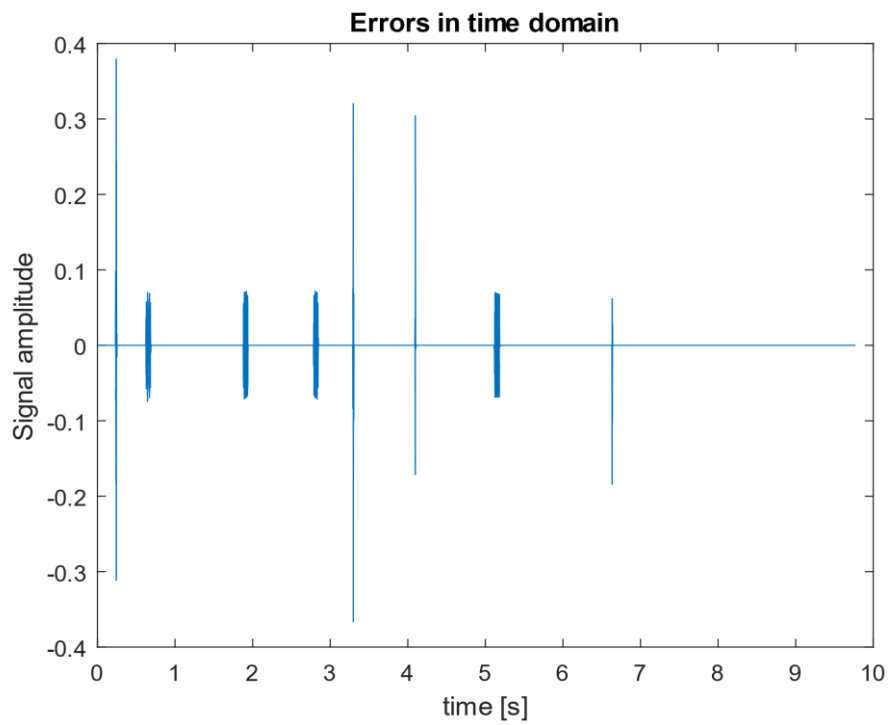
Obrázek 6.7 - Blokové schéma algoritmu spektrálního odečítání

Matlabu „*.mat“ a je k dispozici v datové příloze práce. Podle předpokládaného aktivního zdroje je pak vybrán korespondující odhad výkonového spektra pro odečtení. Tímto se eliminovala nutnost výpočtu odhadu šumu v průběhu vlastního chodu programu.

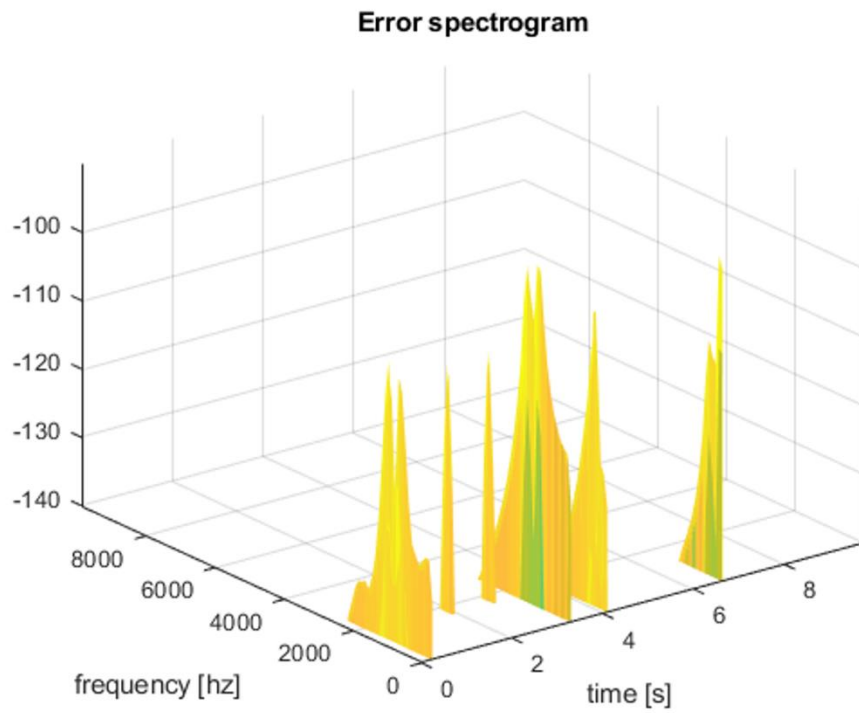
- *src* – vektor aktivních zdrojů pro každé okno spektrogramu
- *sig* – vstupní signál
- *Nabs* – modul průměrovaného spektra pro aktuální zdroj
- *n* – číslo okna
- *Xabs* – modul spektra aktuálního okna signálu
- *Xanle* – fáze spektra aktuálního okna signálu
- *Eabs* – modul spektra chybového signálu
- *tr* – práh přijetí chyby
- *Err* – vektor chybového signálu

Je použita stejná délka okna 512 vzorků jako u výpočtu spektrogramu. Výkon chybového spektra je prahován, aby bylo zanedbáno drobné rušení neovlivňující měření. Implementaci algoritmu lze nalézt v datové příloze pod názvem funkce *spectralSubstraction()*.

6.5.2 Ukázka výsledků



Obrázek 6.8 - Chybový signál v čase



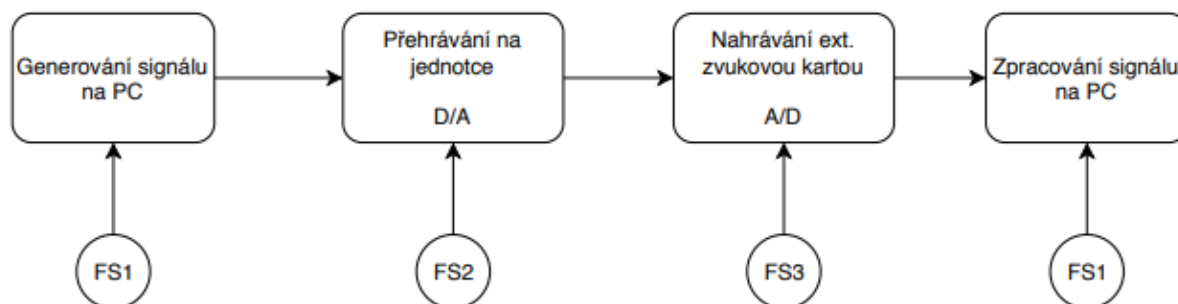
Obrázek 6.9 - Spektrogram chybového signálu

6.6 Frekvenční odezva

Poslední funkcionalitou k implementaci byla schopnost získávání frekvenční odezvy ke kontrole reakce na nastavení ekvalizéru (3.3.3.1). Za tímto účelem byly naimplementovány tři metody: MLS, Sine sweep a Segmentování sine sweep signálu. Při získávání aplikaci prvních dvou metod se projeví problémy spojené s asynchronním měřením, které jsou podrobněji popsány v podkapitolách popisujícími dané metody.

6.6.1 Nahrávání testovacích signálů

Rozdíl mezi synchronním a asynchronním měřením je popsán podrobně v kapitole (4.2.2.2). Při synchronním měření se generace i vyhodnocení signálu provádí na jednom zařízení, kdy je převod signály a digitální do analogové podoby a zpět proveden s identickou vzorkovací frekvencí. Naopak při asynchronním měření se vzorkovací frekvence mohou lišit i v případě, že jsou u obou zařízení nastavena na stejnou hodnotu [32]. Tento problém je znázorněn na obrázku (4.1-2). V našem případě je situace navíc zkomplikována neznámými parametry digitálně analogového převodníku testované jednotky.



Obrázek 6.10 - Cesta signálu pro měření frekvenční odezvy

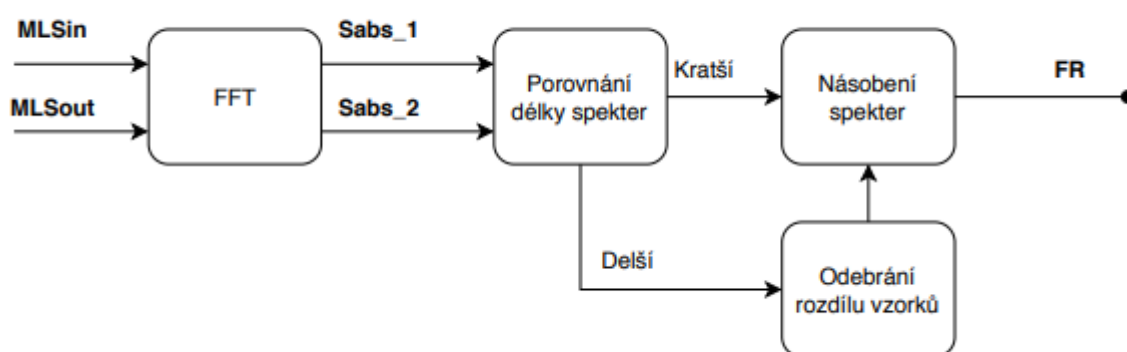
Tudíž můžeme vzorkovací frekvence $fs1$ a $fs3$ nastavit na stejné hodnoty a počítat tak pouze se zkrácením způsobným rozdílem v základním hodinovém taktu jednotlivých zařízení, ale na frekvenci $fs2$ nemáme kontrolu. Tyto rozdíly způsobují problémy hlavně v metodách pracujících s korelací nebo konvolucí jako MLS a Sine sweep.

6.6.2 MLS

Základní princip této metody je popsán v kapitole (4.2.3.1). Ve snaze eliminovat problémy způsobené asynchronním měřením popsané v kapitolách (4.2.3.2) a (5.6.1) byl implementován algoritmus použitý v článku [32] a popsáný podrobně v kapitole (4.2.3.2).

6.6.2.1 Algoritmus

Tento algoritmus využívá práce se spektrem generovaného i nahraného signálu s dvěma hlavními výhodami: První výhodou je nižší výpočetní náročnost násobení dvou spekter oproti konvoluci signálů v časové oblasti. Druhou pak možnost korekce rozdílu vzorkovacích frekvencí pomocí uprav ve spektru, kdy můžeme odečíst několik vzorků delšího spektra namísto převzorkování signálu podle rovnice (4.2.13). Takto upravená spektra lze následně vynásobit a získat tím frekvenční odezvu soustavy [32]. Tento algoritmus je však efektivní pouze v případě celočíselného rozdílu vzorkovacích frekvencí přijatého a generovaného signálu.



Obrázek 6.11 - Blokové schéma algoritmu pro získání frekvenční odezvy metodou MLS s korekcí vzorkovací frekvence

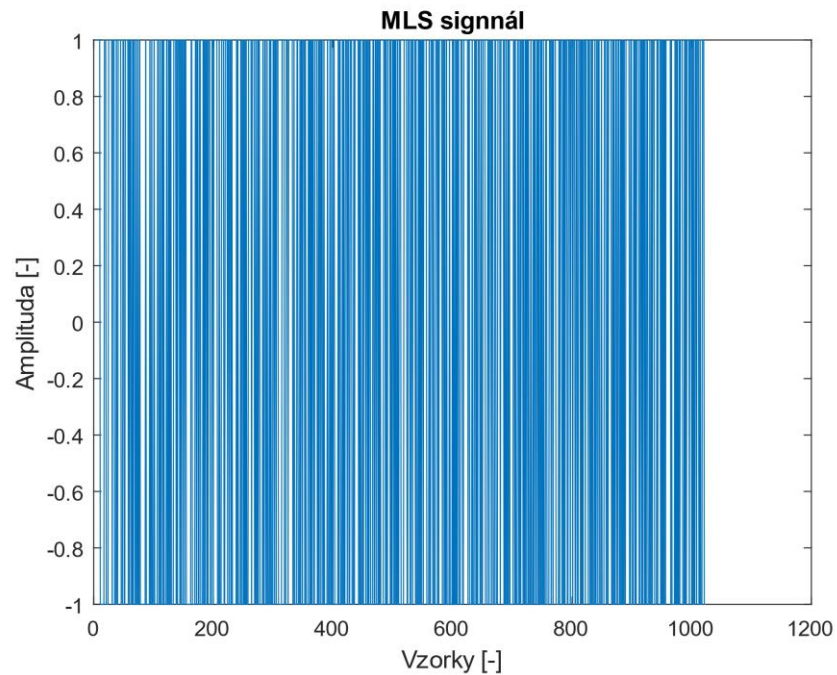
- **MLSin** – vektor generovaného MLS signálu
- **MLSout** – vektor nahraného MLS signálu
- **Sabs1** – modul spektra generovaného MLS signálu
- **Sabs2** – modul spektra nahraného MLS signálu
- **FR** – frekvenční odezva

Podrobně lze implementaci algoritmu prostudovat v datové příloze v kódu hlavní aplikace kde je označený názvem funkce *getFrequencyResponseMLS()*. MLS signál pro testování (5.12) byl vygenerován a otestován pomocí skriptu „MLS.m“, který lze nalézt v datové příloze.

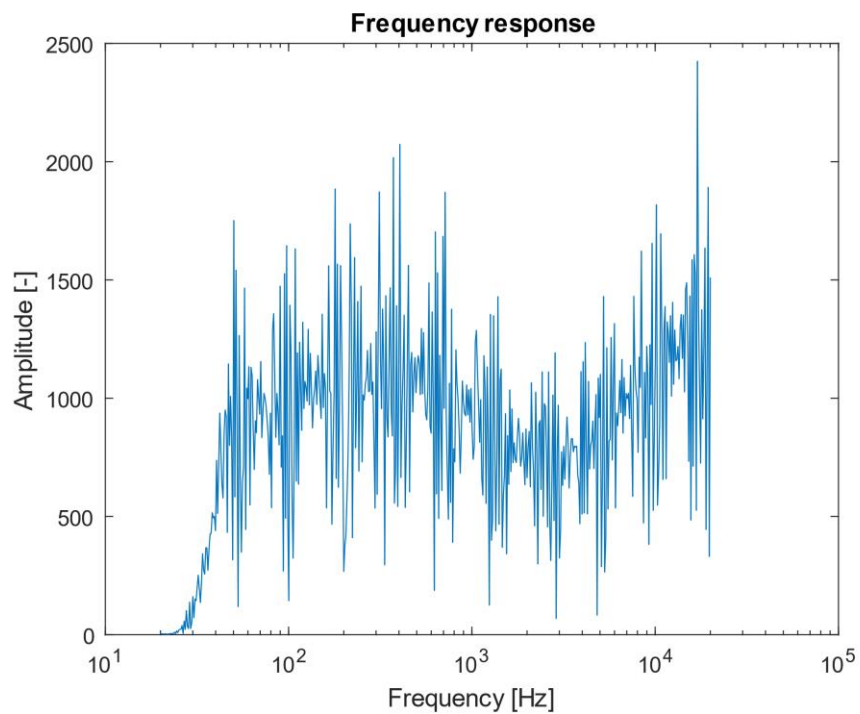
Navzdory aplikaci korekce vzorkovací frekvence se touto metodou nepodařilo dosáhnout uspokojivých výsledků, jak je možné vidět z grafu (5.13). Tento problém mohl mít několik příčin. Jako první se jeví nejasný poměr vzorkovacích frekvencí vzniklý důvody popsány v kapitole (5.6.1), který mohl pravděpodobně způsobit necelý rozdíl

vzorkovacích frekvencí. Další příčinou mohla být nedostatečně přesná práce při předzpracování nahraných souborů z testovacího stavu, které byly ručně vyříznuty ze získaných „wave“ souborů.

6.6.3 Ukázka výsledků



Obrázek 6.12 - Generovaný MLS signál o délce 1024 vzorků



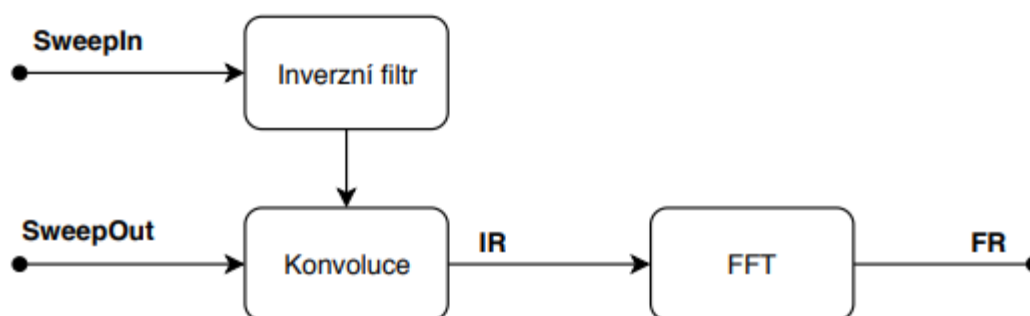
Obrázek 6.13 - Naměřená frekvenční odezva metodou MLS

6.6.4 Sine sweep

Druhou implementovanou metodou byl Sine sweep. Popsaná detailně v kapitole (4.2.1). Vzhledem k harmonickému charakteru přeladovaného sinu popsaného rovnicí (4.8) se předpokládala větší odolnost vůči posuvu vzorkovacích frekvencí než u signálu MLS tvořeného pseudonáhodnou sekvencí impulsů (4.2.2.1).

6.6.4.1 Algoritmus

Základním principem je tvorba inverzního filtru vycházejícího z rovnice testovacího signálu (4.8) podle rovnice (4.11). Další krokem je konvoluce nahraného signálu s inverzním filtrem, kterou získáme impulsní odezvu podle rovnice (4.12). Frekvenční odezva je poté Fourierovým obrazem impulsové odezvy podle rovnice (4.7).



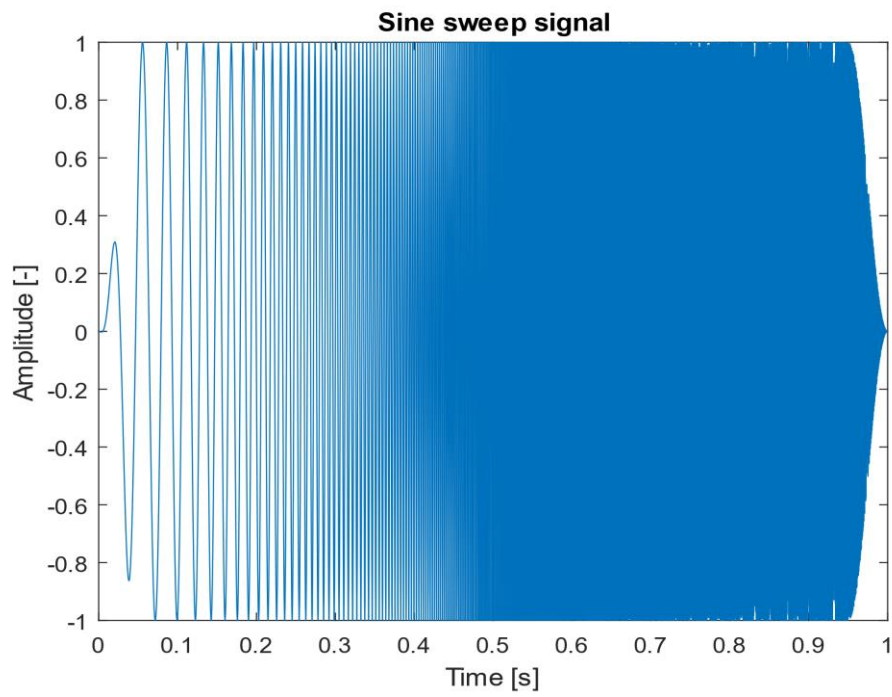
Obrázek 6.14 - Blokové schéma algoritmu metody Sine sweep

- **SweepIn** – vektor generovaného Sine sweep signálu
- **SweepOut** – vektor nahraného Sine sweep signálu
- **IR** – impulsová odezva
- **FR** – Frekvenční odezva

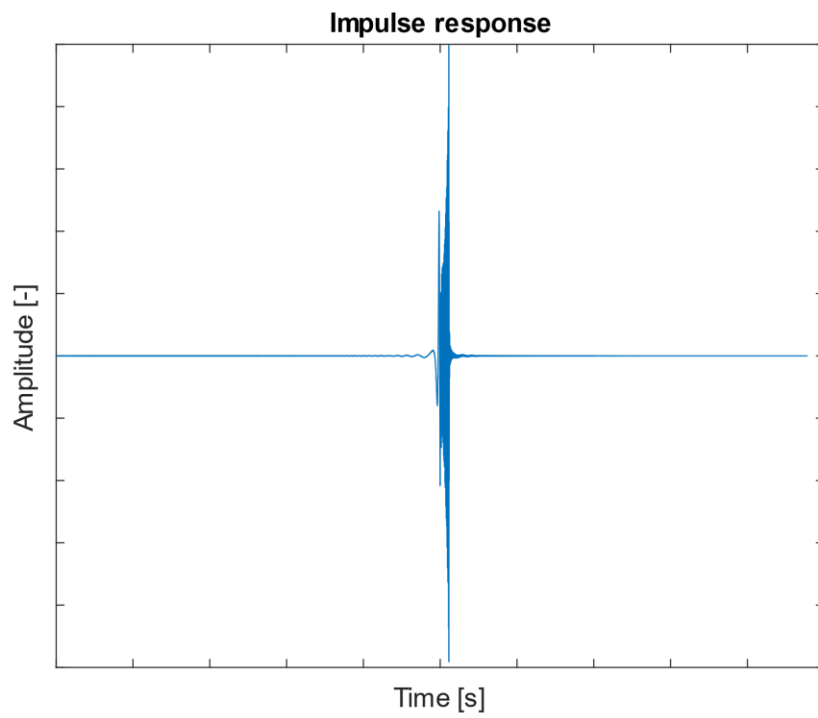
Délka použitého signálu je 1 vteřina, signál je také ošetřen Hammingovým oknem. Výsledky získané touto metodou jsou blíže tíženému výsledku než předchozí metody. Nicméně vzhledem k nutnosti konvoluce vstupního a výstupního signálu a problémům popsaných v předchozí kapitole jsou výsledku stále zatíženy nepřesností. Při porovnání s frekvenční odezvou získanou v následující kapitole, vidíme na obrázku (5.17), že tvar se podobná, ale celá křivka se zdá být posunuta směrem k vyšším frekvencím, což je možné přisoudit nedostatečnému přesnému ořezu výstupního signálu. Generování a kontrola

testovacího signálu jsou řešeny ve skriptu „SineSweep.m“, který je součástí datové přílohy této práce.

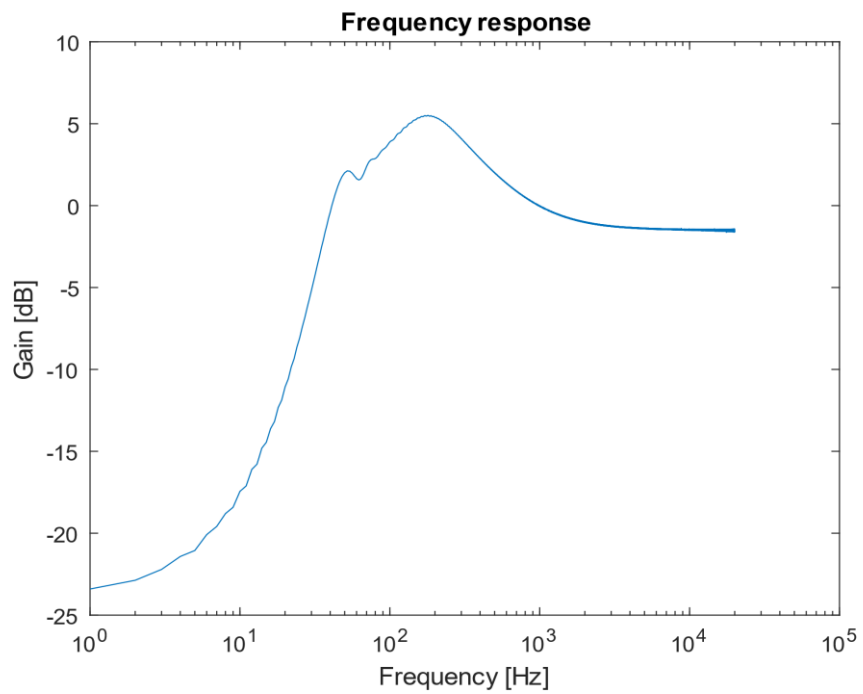
6.6.4.2 Ukázka výsledků



Obrázek 6.15 - Generovaný signál Sine sweep



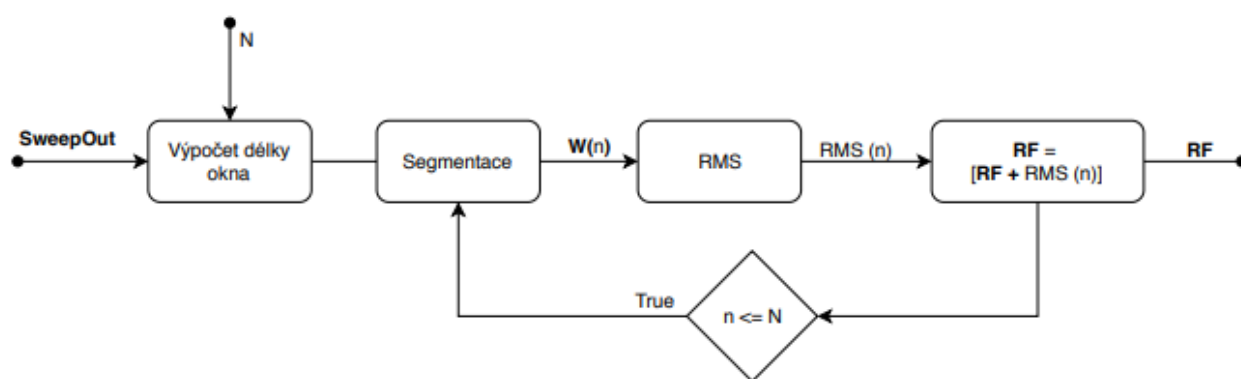
Obrázek 6.17 - impulsová odezva jednotky získaná metodou Sine sweep



Obrázek 6.16 - Frekvenční odezva jednotky získaná metodou Sine sweep

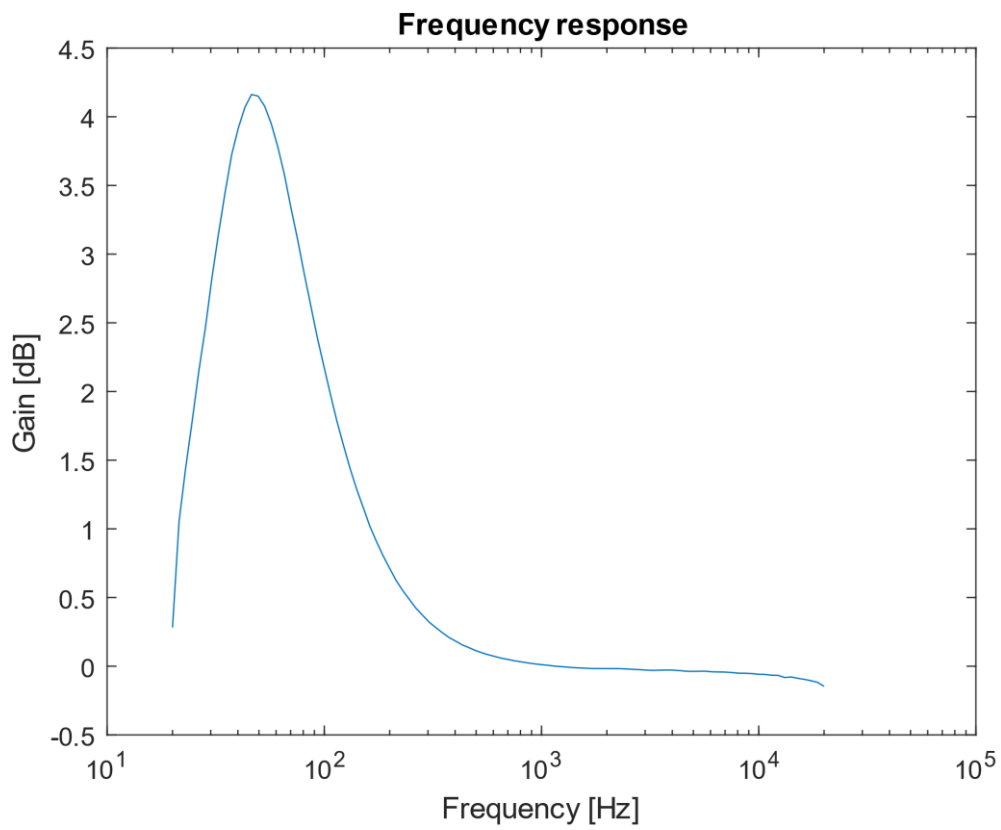
6.6.5 Segmentace signálu sine sweep

Po problémech s předchozími metodami jsem se rozhodl použít metodu Segmentace signálu sine sweep, která nevyužívá konvoluce a měla by tak být robustnější vůči nepřesnostem vycházejících z problému popsáných v kapitole (5.6.1). Metoda je podrobněji popsána v kapitole (4.2.2). Její princip spočívá v přehrání delšího signálu sine sweep (4.8), který je následně segmentován na předem daný počet dílů, v testu 100. Z každého takto připraveného okna je započítán průměrný výkon RMS. Vzhledem k rovnici (4.17) jsou pak tyto hodnoty úměrné frekvenční odezvě pro frekvenční pásmo pokryté signálem sine sweep v intervalu segmentu.



Obrázek 6.18 - Blokové schéma algoritmu získávání frekvenční odezvy metodou Segmentace signálu sine sweep

Výsledná frekvenční odezva je vykreslována v logaritmické škále vztažené k referenční hodnotě v pásmu 1 kHz. Vzhledem k pevnému počtu segmentů je tato metoda odolná vůči rozdílným délkám zpracovávaného signálu, pokud není signál příliš krátký, aby se velikost okna nepřibližovala velikosti periody signálu na nízkých frekvencích. V testu použity signály o délce 30 s, 60 s. Výsledky získané touto metodou odpovídají reálné frekvenční charakteristice měřené jednotky. Na grafu (5.19) je znázorněna frekvenční odezva pro výchozí nastavení ekvalizéru, můžeme si všimnout zvýraznění hloubek, které bylo zvoleno výrobcem za účelem zábavnějšího zvukového výstupu jednotky. Další průběhy jsou k dispozici v aplikaci „Audio Test Postprocessor“, která je k dispozici v datové příloze práce a také tématem další kapitoly.



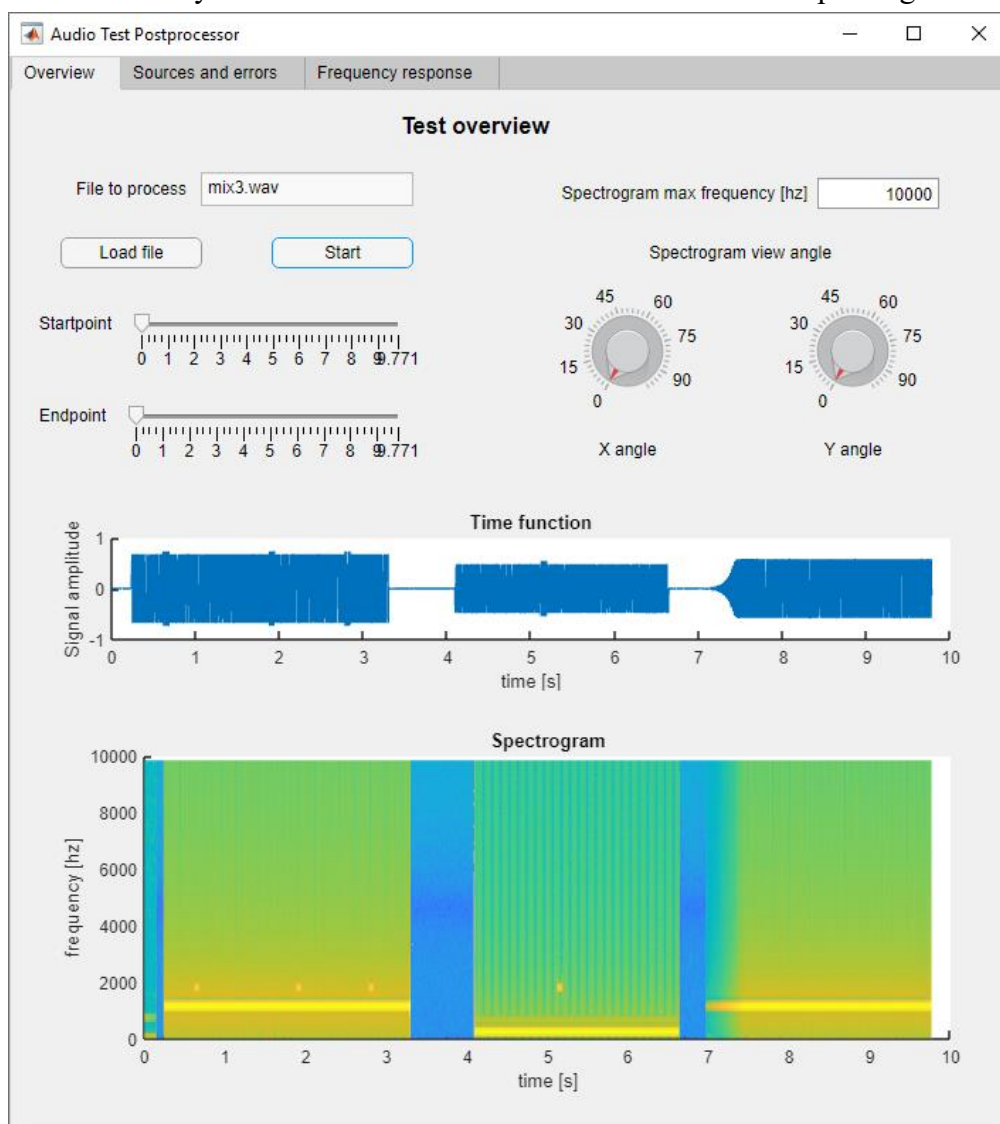
Obrázek 6.19 - Frekvenční odezva testované jednotky získaná metodou Segmentace signálu sine sweep

6.7 Popis aplikace

Jak bylo řečeno v úvodu kapitoly, je výstupem práce aplikace sdružující všechna zmíněná vylepšení a implementované algoritmy do uceleného nástroje určeného pro zpracování zvukových záznamů z testů. Zároveň slouží jako demonstrace funkčnosti implementovaných algoritmů a jejich použitelnosti při reálném testování. V následujících řádcích si popíšeme jednotlivé funkce aplikace a jejich zamýšlené použití. Aplikace samotná i její zdrojový kód jsou k dispozici v datové příloze práce.

6.7.1 Úvodní strana

Po spuštění aplikace se uživatel ocitne na kartě určené k vizuální kontrole záznamu testu. K dispozici je panel pro výběr souboru pomocí dialogového okna. Obraz signálu v časové oblasti i spektrogram. Je také možné navigovat se v grafech pomocí ovládacích prvků, které řídí časový i frekvenční rozsah zobrazení a úhel natočení spektrogramu.



Obrázek 6.20 - Úvodní strana aplikace

6.7.2 Kontrola zdrojů

Po načtení souboru a jeho vizuální kontrole na první kartě má uživatel možnost přejít na kartu „Sources and errors“. Jejíž první spustitelnou funkcí je kontrola aktivních zdrojů v testu Implementována podle algoritmu popsáném v kapitole (5.4). Proces se spustí tlačítkem „Process“.



Obrázek 6.21- Karta aplikace "Sources and errors"

Graf aktivních zdrojů v čase se zobrazí v horní části aplikace. Zároveň proběhne kontrola neznámých zdrojů v testovacím signálu s kontrolou „mezer“ a tichých míst, pokud je mezera mezi aktivitou dvou zdrojů delší, než je povolený limit. Test je vyhodnocen jako

chybový. Zároveň jsou k dispozici statistické informace o době trvání jednotlivých zdrojů a mezer.

6.7.3 Detekce artefaktů

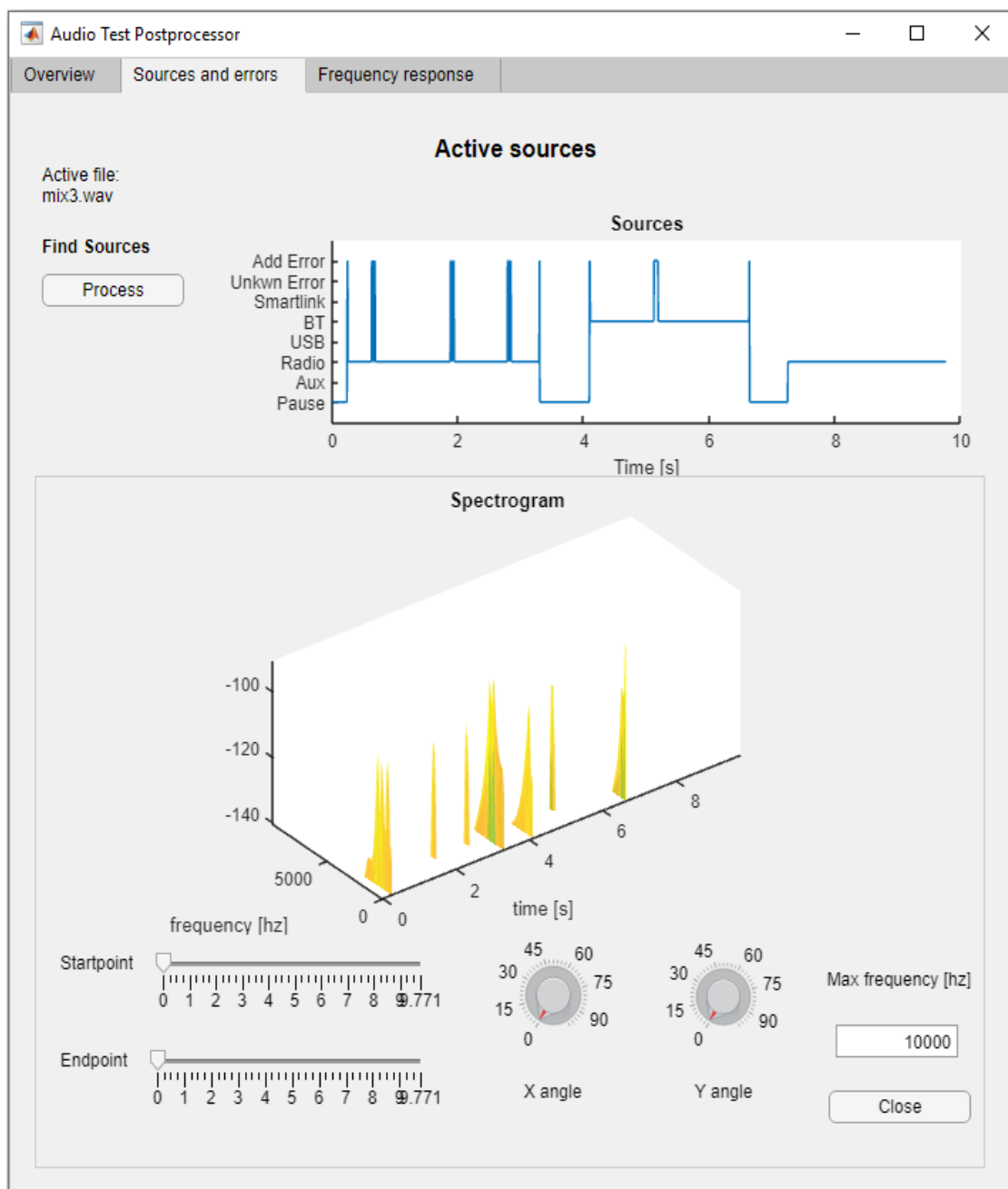
Dalším krokem je kontrola aditivního šumu a rušivých artefaktů v signálu, které by mohli způsobit chyby v testech. Tlačítkem „Check“ se spustí funkce jejíž algoritmus je popsán podrobněji v kapitole (5.5).



Obrázek 6.22 - Detekce artefaktů v aplikaci

V prostředím grafickém poli jsou zobrazeny chyby způsobené artefakty v čase. Současně je informace o aditivní chybě zobrazena v grafu aktivních zdrojů pro identifikace

potenciálně problémového zdroje. Ve spodní statistické části je také zobrazen celkový čas signálu zatíženého chybou. Test je vyhodnocen jako otevřený, protože výsledky ze signálu zatíženého artefakty nemusí být validní. Po stisku tlačítka „Spektrogram“ je uživateli



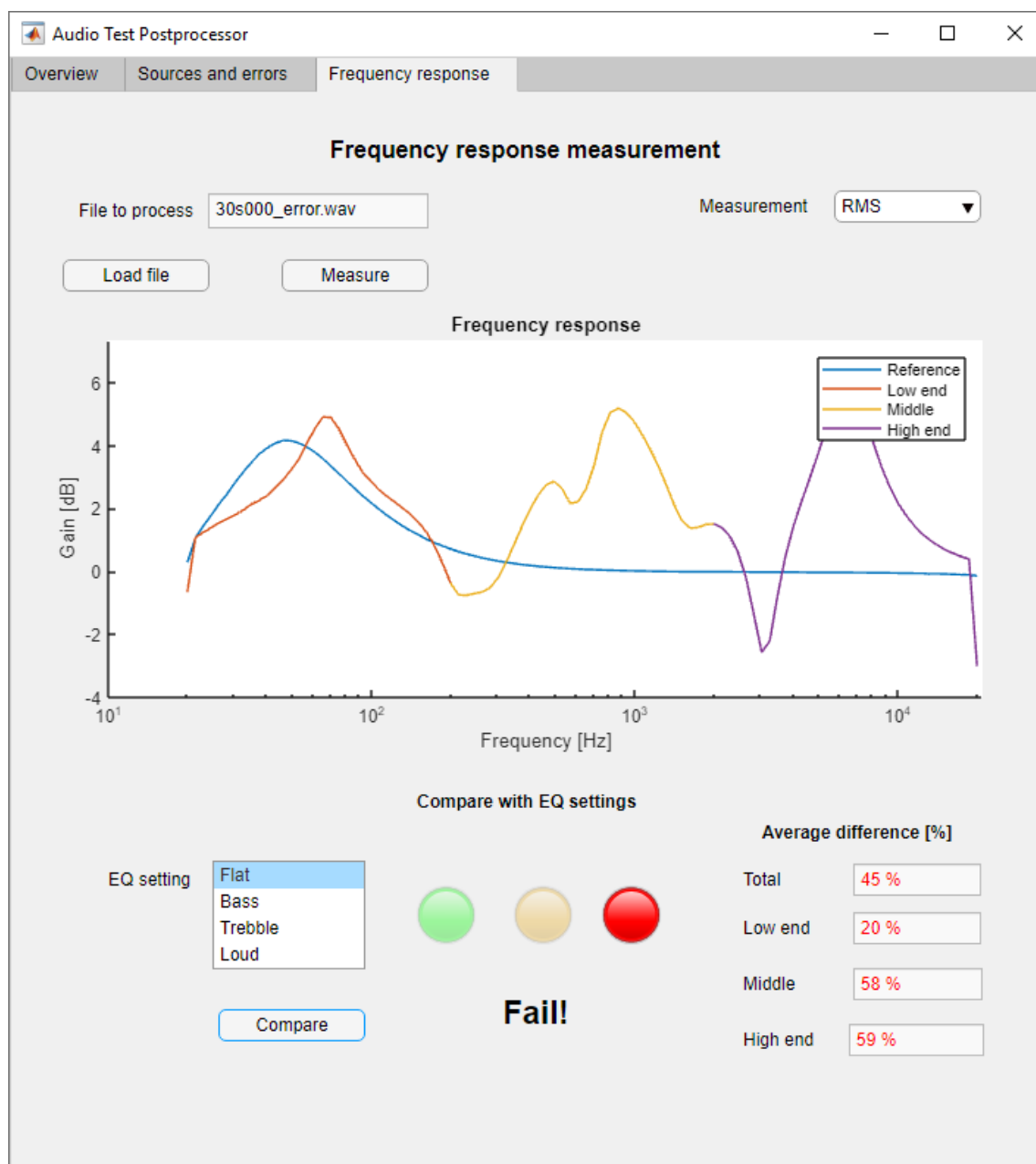
Obrázek 6.23 - Spektrogram chybového signálu v aplikaci

zobrazen spektrogram chybového signálu.

6.7.4 Frekvenční odezva

Problematikou získávání frekvenční odezvy a problémům s ní spojený se zabývá kapitola (5.6). Jako hlavní metodu pro měření frekvenční odezvy v aplikaci se používá Segmentace signálu sine sweep, jejíž je popsán v kapitole (5.6.5) a v aplikaci je

pojmenována jako RMS. Zbylé dvě metody jsou v aplikaci také k dispozici s ukázkovými soubory k otestování. Přepnou na ně lze rozbalovacím menu v pravém horním rohu karty.



Obrázek 6.24 - Výpočet frekvenční odezvy v aplikaci

Plná funkcionlita je však k dispozici pouze pro hlavní metodu z důvodů popsanych v kapitole (5.6).

V levém horním rohu má uživatel možnost vybrat soubor k analýze. Tlačítem „Measure“ se spustí vyhodnocení frekvenční odezvy. Po prohlédnutí průběhu v grafickém poli karty může porovna s očekavanými průběhy. Pro potřeby této demonstrace byly vybrány čtyři typická nastavení. Frekvenční odezva je poté porovnána se zvolenou referencí. Hodnotí se procentuální rozdíl v celém průběhu a zvlášt' ve třech frekvenčních pásmech.

Pokud odchylka překročí stanovenou mez, jako na obrázku (5.24), je kontrola rekvenční odezvy ozančena jako chybná.

7 Závěr

Tématem této diplomové práce byly Audio testy infotainmentových jednotek. Cílem bylo seznámit se s problematikou jejich testování a zaměřit se na zvukový výstup jednotky. Dále se zaměřit na metody detekce artefaktů a určování změn ve frekvenční charakteristice jednotky v závislosti na nastavení ekvalizéru. Výstupem mělo být vlastní řešení použitelné při testování.

V první části práce je čtenář seznámen s pojmem infotainment a jeho krátkou historií. Dále jsou přestaveny běžné funkce dnešních infotainmentových jednotek pro automobilový průmysl. Poté je konkrétně představena jednotka Škoda Swing, které byla předmětem testování v dalších částech práce.

V druhé kapitole jsou čtenáři nejprve poskytnuty teoretické podklady týkající se testování softwaru relevantní k testům infotainmentu. Záběr je postupně konkretizován, z počátku jsou popsány obecné termíny a postupy týkající se testování softwaru. Následně jsou popsány techniky používané při návrhu a exekuci testů infotainmentu.

Třetí část se zabývá popisem testovací laboratoře ve společnosti Digiteq automotive s. r. o a testovacího stavu vyžitého při pokusech v rámci této práce.

Následující kapitola se soustředí sledové parametry a metody zpracování zvuku aktuálně používané k jejich kontrole na zmíněném pracovišti. Následně je navrženo několik vylepšení ke zkvalitnění testování zvukového výstupu. Konkrétně kontrola změn frekvenční odezvy v závislosti na nastavení ekvalizéru, využití zvukových záznamů z testů, kontinuální kontrola aktivních zdrojů a detekce artefaktů. Posledním zmíněným tématem se zabývá i závěr této části, kdy jsou studovány reálné artefakty vyskytující se při testování na konkrétním testovacím stavu. Po zohlednění vyskytujících se artefaktů jsou vybrány dvě metody detekce k dalšímu studiu a sice Wienerova filtrace a Spektrální odečítání.

V páté kapitole jsou postupně představeny všechny metody zpracování zvukového signálu relevantní k této práci. Postupně jsou popsány zmíněné metody detekce artefaktů, Wienerova filtrace a Spektrální odečítání. Následně metody Sine sweep, Segmentace signálu sine sweep a MLS zabývající se určováním frekvenční odezvy lineárních systémů.

Šestá část popisuje vybrané algoritmy a jejich implementaci. Z počátku kapitoly je představeno požitě vývojové prostředí Matlab App Designer, následně způsob získávání dat pro testy a posléze jsou popsány jednotlivé implementované algoritmy. Jako první je

popsán způsob získávání vizuálních informací o zvukovém záznamu testu. Poté algoritmus pro kontinuální kontrolu zdrojů, u které bylo s výhodou využito kontroly nad typem přehrávaných testovacích signálů. Pro detekci artefaktů byl po zvážení vstupních podmínek testování a typických vlastností obou zvažovaných metod vybrán mírně upravený algoritmus Spektrálního odečítání, ve kterém je opět využito znalosti charakteru testovacích signálů a výstupu funkce pro určování aktivních zdrojů v čase.

V následující části šesté kapitoly je popsán postup při určování frekvenční odezvy jednotky. Z počátku jsou popsána omezení při měření různými vzorkovacími frekvencemi jednotlivých zařízení. Tyto komplikace vedly k neuspokojivým výsledkům zejména u metody MLS, kde navzdory implementaci metody na kompenzaci rozdílu vzorkovacích frekvencí ve frekvenční oblasti, byly výsledky značně nepřesné a zatížené šumem. Mezi možné důvody mohl patřit necelý rozdíl mezi vzorkovacími frekvencemi nebo způsob získávání nahraných testovacích signálů a jsou podrobněji popsány v dané kapitole. U metody Sine sweep se zmíněné problémy neprojeví tak výrazně, ale výsledky stále nebyly dostačující pro použití při vyhodnocování změn nastavení ekvalizéru. Z toho důvodu byla implementována třetí metoda s názvem Segmentování signálu sine sweep. Ta prokázala robustnost vůči řešeným problémům i různým délkám testovacích signálů. Výsledky takto získané dosahují dostatečné přesnosti pro kontrolu nastavení ekvalizéru.

V poslední kapitole práce je představena aplikace sdružující jednotlivé implementované algoritmy do uceleného nástroje vhodného pro použití při testování a poskytujícího všechny zadané doplňující funkce. Uživatel může také ověřit funkci implementovaných algoritmů na reálných datech.

Všechny funkce výsledné aplikace byly ověřeny na reálném pracovišti se skutečnými daty z testování. Zmíněná aplikace byla také představena několika pracovníkům testovacího centra, kteří potvrdili reálnou využitelnost představených algoritmů.

8 Zdroje

- [1] Huang, Y., et al.: Development of an automated testing system for vehicle infotainment system. *Int. J. Adv. Manuf. Technol.* vol. 51. pp. 233 - 246. Nov. 2010. doi: 10.1007/s00170-010-2626-2
- [2] INFOTAINMENT: Co to je a jak se pozná, Rozumět mediím, [online], 22.5.2019. dostupný z: <http://web.archive.org/web/20070715082016/http://www.rozumetmediim.cz/slovnicek/slovnicek/infotainment.pdf>
- [3] BENGLER, Klaus, Heiner BUBB, Ingo TOTZKE, Josef SCHUMANN a Frank FLEMISCH. *Automotive. Information Ergonomics* [online]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, s. 99 DOI: 10.1007/978-3-642-25841-1_4. ISBN 978-3-642-25840-4. Dostupné z: http://link.springer.com/10.1007/978-3-642-25841-1_4
- [4] Vehicle audio: History: 1920s-1940s. *Wikipedia EN*, 2002 [online], 22.5.2019. Dostupné z: https://en.wikipedia.org/wiki/Vehicle_audio#1920s-1940s
- [5] Škoda Petr: Testování použitelnosti pro technologie Smart integration v automobilovém průmyslu. Diplomová práce, [online], 22.5.2019. Dostupné z: <https://theses.cz/id/6dtnhn/STAG85852.pdf>
- [6] Chromec. Jakub: Řízení vývoje elektroniky vozů Škoda s ohledem na management nákladů, Bakalářská práce ČVUT FEL, 2015
- [7] GRAHAM, Dorothy; VAN VEENENDAAL, Erik; EVANS, Isabel. *Foundations of software testing: ISTQB certification*. Cengage Learning EMEA, 2008
- [8] Lingea, Školní slovník současné češtiny. V Brně, 2012. ISBN 978-80-87471-59-3.
- [9] BEIZER, Boris. *Software testing techniques*. 2nd ed. New York: Van Nostrand Reinhold, c1990. ISBN 0442206720.
- [10] BLACK, Rex. *Managing the testing process: practical tools and techniques for managing software and hardware testing*. 3rd ed. Indianapolis, MN: Wiley, c2009. ISBN 978-0470404157.
- [11] DAVID, F.N. *Probability Theory for Statistical Methods*. [s.l.]: Cambridge University Press, 1949. S. 28. (anglicky)

- [12] Testování komponent Digiteq automotive s. r. o, [online], 22.5.2019. Dostupné z: <https://www.digiteqautomotive.com/cs/test-centrum>
- [13] Vision Builder AI, National Instruments, [online], 22.5.2019. Dostupné z: <http://www.ni.com/cs-cz/shop/electronic-test-instrumentation/application-software-for-electronic-test-and-instrumentation-category/what-is-vision-builder-for-automated-inspection.html>
- [14] Vector CANoe, [online], 22.5.2019. Dostupné z: <https://www.vector.com/int/en/products/products-a-z/software/canoe/>
- [15] MGB grabber Digiteq automotive, [online], 22.5.2019. Dostupné z: <https://www.digiteqautomotive.com/cs/produkt/mgb>
- [16] Vector Interfaces, [online], 22.5.2019. Dostupné z: <https://www.vector.com/int/en/products/products-a-z/hardware/network-interfaces>
- [17] RDS service generator Götting, [online], 22.5.2019. Dostupné z: <https://www.goetting-agv.com/components/81300>
- [18] Bluepirat mini, Telmotive AG, [online], 22.5.2019. Dostupné z: <https://www.telemotive.de/en/products/hardware/blue-pirat-mini/>
- [19] VAS 6154, Škoda e-shop, [online], 22.5.2019. Dostupné z: http://equipment.skoda-auto.cz/webapp/wcs/stores/servlet/ProductDisplayViewDV?catalogId=10051&DM_PersistentCookieCreated=true&errorViewName=ProductDisplayErrorView&parent_category_rn=10010&top_category=10010&categoryId=10010&productId=52003&storeId=10001
- [20] Produkty: USB switcher, [online], 22.5.2019. Digiteq Automotive Dostupné z: <https://www.digiteqautomotive.com/cs/produkty>
- [21] NI 9902, National Instruments, [online], 22.5.2019. Dostupné z: <http://www.ni.com/documentation/en/c-series-voltage-input-module/latest/9201/block-diagram/>
- [22] ZUP 20-20, TDK-Lambda, [online], 22.5.2019. Dostupné z: <https://uk.tdk-lambda.com/KB/ZUP-Datasheet.pdf>
- [23] HERODES, Jakub. Metody potlačení síťového rušení v signálech EKG [online]. Brno, 2015. Dostupné z: <http://hdl.handle.net/11012/40698>.

- [24] REICHL, Jaroslav. Frekvenční modulace [online], 22.5.2019. Dostupné z: <http://fyzika.jreichl.com/main.article/view/1390-frekvencni-modulace>
- [25] MOLDŘÍKOVÁ, Zuzana. METODY POTLAČENÍ ŠUMU PRO ROZPOZNÁVAČE ŘEČI [online]. Brno, 2014, Dostupné z: <https://core.ac.uk/download/pdf/30299537.pdf?fbclid=IwAR2bfTdKmokD6EcgcZr4uLrNSgO6cqR9mHE1wRBbbV2jcxpqLtLbBnP2q1k>
- [26] PELIKÁN, Martin. IDENTIFIKACE ŘEČOVÉ AKTIVITY V RUŠENÉM ŘEČOVÉM SIGNÁLU Brno, 2013, [online], 22.5.2019. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=67686&fbclid=IwAR2BLadv29nWZoGAdpXzN4NTqIBv6oDKEgTCwIARoQrcgPzNDB-Xdy0-SCU
- [27] LOIZOU, P.C. Speech enhancement: theory and practice. 2nd ed. Boca Raton, Fla.: CRC Press, 2013, 689 s. ISBN 978-1-4665-0421-9.
- [28] Znojemský, Karel, “Popis metody spektrálního odečítání“, FEL ČVUT, [online], 22.5.2019. Dostupné z: <http://noel.feld.cvut.cz/vyu/prj-czs-asi/znojemsky/metoda.html>
- [29] MAFI, Mehdi, MARTIN, Harold, A Comprehensive Survey on Impulse and Gaussian Denoising Filters for Digital Images,” Signal Processing, vol. 157, pp. 236-260, 2019
- [30] JAN, Jiří. Číslíková filtrace, analýza a restaurace signálů. 2. upr. a rozš. vyd. Brno: VUTIUM, 2002. ISBN 80-214-2911-9.
- [31] Kadlec, František: Zpracování akustických signálů. 1 vydání, Vydavatelství ČVUT, 2002
- [32] P. Kovitz, “Two Maximal Length Sequence Devices for Measuring Room Acoustics Parameters,” presented at the AES 11th International Conference: Test & Measurement (1992 May), conference paper 11-034.
- [33] Novak, Antonin; Rund, Frantisek; Honzik, Petr: Impulse Response Measurements using MLS Technique on Nonsynchronous Devices, JAES Volume 64 Issue 12 pp. 978-987; December 2016
- [34] Muller, Swen; Massarani, Paulo: Transfer-Function Measurement with Sweeps, JAES Volume 49 Issue 6 pp. 443-471; June 2001

[35] Matlab App Designer, Mathworks, [online], 22.5.2019. Dostupné z:
<https://www.mathworks.com/products/matlab/app-designer.html>

[36] Audacity, Audacityteam, [online], 22.5.2019. Dostupné z:
<https://www.mathworks.com/products/matlab/app-designer.html>

Obsah datové přílohy

- Aplikace „Audio Test Postprocesor“ s ukázkou implementovaných algoritmů
- Zdrojový kód aplikace ve formátu „.pdf“
- Skript pro generaci MLS signálu „MLS.m“
- Skript pro generaci Sine sweep signálu „SineSweep.m“
- Testovací data
- Práce ve formátu PDF