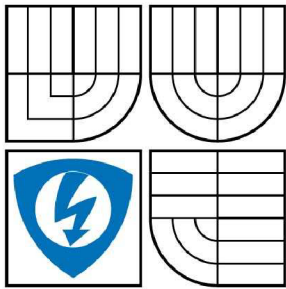


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKACNÍCH
TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ



FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

OBJEKTÍVNI METODY PRO HODNOCENÍ KVALITY AUDIOSIGNÁLŮ

OBJECTIVE METHODS OF EVALUATION OF THE AUDIO SIGNAL QUALITY

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MICHAL GRANÁT

VEDOUcí PRÁCE

SUPERVISOR

Ing. ONDŘEJ RÁŠO

BRNO 2009

ANOTACE

Táto diplomová práca sa zaoberá objektívnym hodnotením kvality audiosignálov. Podrobne seznamuje čitateľa s algoritmy, ktoré sú využívané v jednotlivých metódach. Jednotlivé metódy sú rozebrané z hľadiska výpočetnej náročnosti, schopnosti kvalitne predikovať subjektívnu kvalitu audiosignálov. V prvej a druhej časti práce sa venuje charakteristike základných vlastností zvukov, zavedením a vysvetlením pojmov, s ktorými sa bežne pracuje v ďalšej časti práce. Tretia časť sa venuje teoretickému popisu jednotlivých metód, ich výhodám a nevýhodám a praktickému použitiu. V štvrtéj časti je popísaná praktická implementácia dvoch vybraných metód v grafickom rozhraní programu MATLAB. Tady sú popísané všetky funkcie, ktoré aplikácie ponúkajú. V tejto časti sú podrobne rozebrané výsledky dosiahnuté praktickým testovaním audiosignálov s rôznym charakterom vo vytvorených aplikáciách a možnosti použitia aplikácií v praxi. Zároveň je zhodnotená kvalita dosiahnutých výsledkov.

KLÍČOVÉ SLOVÁ

Audiosignál, objektívne metódy, artikulačný index, šum, bark, kritické pásmo,

ABSTRACT

This diploma thesis deals with the objective methods of audio-signal qualification. In detail have been described algorithms to the reader which are used in individual methods. Individual methods are analyzed from computational intensity, ability to predict precisely subjective quality of audio-signals, point of view. There have been described basic variables by which audio-signals are described in further parts of this thesis. Second part has been dedicated to theoretical description of individual methods – their advantages and disadvantages and practical forms of usage. Third part has been dedicated to implementation of two chosen methods in graphical interface of MATLAB program. All the functions are described in detail which these applications offer. In the last part of the thesis there are in detail described results obtained by practical testing of audio-signals with different character in created applications and evaluated quality of reached results as an outcome.

KEY WORDS

Audio signal, objective method, Articulation Index, noise, bark, critical band

Prohlášení

Prohlašuji, že svojí diplomovou práci na téma OBJEKTÍVNE METÓDY PRO HODNOCENÍ KVALITY AUDIOSIGNÁLŮ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

.....
podpis autora

Obsah

Úvod	8
1 Zvuk	9
1.1 Definícia zvuku.....	9
1.2 Druhy zvukov	9
2 Vlastnosti zvuku.....	10
2.1 Amplitúda	10
2.2 Frekvencia	11
2.3 Výška.....	11
2.3.1 Vnímanie výšky zvuku	11
2.4 Hlasitosť zvuku.....	13
2.4.1 Vnímanie hlasitosti	14
2.4.2 Meranie subjektívnej hlasitosti.....	14
2.5 Farba	16
3 Kvalita audio signálov	17
3.1 Zrozumiteľnosť.....	17
3.2 Subjektívna kvalita a zrozumiteľnosť.....	17
3.3 Subjektívne metódy merania kvality audiosignálov.....	18
3.4 Objektívna kvalita zrozumiteľnosti	18
3.4.1 Artikulačný index.....	20
3.4.2 Meranie založené na LPC	22
3.4.3 Metódy založené na percepčných vlastnostiach	23
3.4.4 Metóda PSQM	25
3.4.5 Metóda PESQ	26
3.4.6 Neintrusívne metódy	27
4 Implementácia.....	30
4.1 Výpočet Artikulačného Indexu.....	30
4.2 Metóda Wighted Spectral Slope	37
4.3 Výpočtová náročnosť	40
4.4 Grafické rozhranie	40
5 Záver	45
Použitá literatúra.....	46
A Prvá Príloha	49
B Druhá Príloha	60

Zoznam tabuliek

Tabuľka 1: Rozdelenie frekvenčného rozsahu do rôznych kmitočtových pásiem	13
Tabuľka 2: Pomerné intenzity zvuku v dB	14
Tabuľka 3: Vybrané zvuky a ich hladiny	15
Tabuľka 4: Stupnica metódy MOS	18
Tabuľka 5: Výsledky testovania AI a.wav	34
Tabuľka 6: Výsledky testovania AI s.wav	34
Tabuľka 7: Výsledky testovania AI test.wav	35
Tabuľka 8: Výsledky testovania AI 7.wav	35
Tabuľka 9: Výsledky testovania AI cos.wav	35
Tabuľka 10: Výsledky testovania AI housle.wav	35
Tabuľka 11: Doporučené hodnoty počtu filtrov	37
Tabuľka 12: Výsledky testovania WSSM a.wav	38
Tabuľka 13: Výsledky testovania WSSM s.wav	38
Tabuľka 14: Výsledky testovania WSSM test.wav	39
Tabuľka 15: Výsledky testovania WSSM 7.wav	39
Tabuľka 16: Výsledky testovania WSSM cosinus.wav	39
Tabuľka 17: Výsledky testovania WSSM housle.wav	39
Tabuľka 18: Porovnanie výpočtovej náročnosti	40

Zoznam obrázkov

Obrázok 1 : Objektívne a subjektívne vlastnosti zvuku	10
Obrázok 2: Kritické frekvenčné pásma	12
Obrázok 3: Závislosť hladiny akustického tlaku od frekvencie pri rôznych hladinách hlasitosti[10].	15
Obrázok 4: Porovnanie chybového signálu s maskovacím prahom	19
Obrázok 5: Porovnanie vnútorných reprezentácií signálu	19
Obrázok 6: Blokovaná schéma PESQ	27
Obrázok 7: Blokovaná schéma 3SQM	28
Obrázok 8: Vývojový diagram algoritmu výpočtu Artikulačného indexu	31
Obrázok 9: Priebeh signálu test.wav „létající prase“	31
Obrázok 10: Spektrogram signálu test.wav	32

Obrázok 11: Biely šum	32
Obrázok 12: Zašumený signál test.wav	33
Obrázok 13: Spektrogram degradovaného signálu test_deg.wav.....	33
Obrázok 14: Hodnota AI v jednotlivých rámcoch signál test.wav	34
Obrázok 15: Bloková schéma metódy WSS.....	37
Obrázok 16: Rozloženie banky barkových filtrov.....	38
Obrázok 17: Aplikácia pre výpočet Artikulačného Indexu	41
Obrázok 18: Aplikácia s implementovanou metódou WSS	44
Obrázok 19: Priebeh signálu a.wav	60
Obrázok 20: Spektrogram signálu a.wav.....	60
Obrázok 21: Priebeh signálu a_deg.wav	61
Obrázok 22: Spektrogram signálu a_deg.wav	61
Obrázok 23: Priebeh signálu 7.wav	62
Obrázok 24: Spektrogram signálu 7.wav	62
Obrázok 25: Priebeh signálu 7_deg.wav	63
Obrázok 26: Spektrogram signálu 7_deg.wav.....	63
Obrázok 27: Priebeh signálu cosinus.wav	64
Obrázok 28: Spektrogram signálu cosinus.wav	64
Obrázok 29: Priebeh signálu cosinus_deg.wav	65
Obrázok 30: Spektrogram signálu cosinus_deg.wav.....	65
Obrázok 31: Priebeh signálu housle.wav	66
Obrázok 32: Spektrogram signálu housle.wav	66
Obrázok 33: Priebeh signálu housle_deg.wav.....	67
Obrázok 34: Spektrogram signálu housle_deg.wav	67
Obrázok 35: Priebeh signálu s.wav	68
Obrázok 36: Spektrogram signálu s.wav	68

Úvod

V súčasnej dobe veľkého technologického pokroku nám produkty vedy a techniky umožňujú uchovávať obrovské množstvá dát rôzneho druhu za pomerne prijateľné náklady.

Text tejto práce sa zaoberá zvukovými dátami. Konkrétne objektívnym hodnotením kvality audiosignálu. Cieľom práce je pochopiť existujúce metódy hodnotenia kvality audiosignálu s ich možnosťami, porovnať ich a naznačiť aplikačné využitie v praxi. Projekt je rozdelený do troch častí:

V prvej a v druhej časti sa venujem charakteristike základných vlastností zvuku (audiosignálu), zavedením a vysvetlením pojmov, s ktorými sa bežne pracuje v ďalšej časti práce.

V tretej časti sú podrobne matematicky popísané a vysvetlené vybrané objektívne metódy hodnotenia kvality audio signálu. Od jednoduchých až po najsofistikovanejšie, najmodernejšie, využívajúce neurónové siete. V tejto časti sa ďalej môžeme dočítať o delení objektívnych metód na intrusívne a neintrusívne podľa toho, či na určenie výslednej kvality potrebujú, alebo nepotrebujú referenčný signál. Pre úplnosť sú tu spomenuté a popísané aj niektoré subjektívne metódy, ktoré s objektívnymi veľmi úzko súvisia a niekedy sa dokonca prelínajú.

Posledná časť popisuje praktickú implementáciu vybraných metód v grafickom rozhraní programu MATLAB. Aplikácie, v ktorých sú implementované vybrané metódy objektívneho hodnotenia kvality audiosignálov, som podrobil rozsiahlemu testovaniu, pričom som sa zameril na schopnosť predikcie kvality signálu ako aj na výpočtovú náročnosť.

1 Zvuk

1.1 Definícia zvuku

Zvuk je každé pozdĺžne mechanické vlnenie v látkovom prostredí, ktoré je schopné vyvolať v ľudskom uchu alebo inom zvukovom prijímači vnem. Frekvencie tohto vlnenia ležia v rozsahu približne od (16-20 Hz až do 20kHz). Za jeho hranicami človek zvuk nevníma. V širšom zmysle môžeme za zvuk považovať aj vlnenie s vyššími a nižšími frekvenciami ako udáva tento rozsah. Zvuk s frekvenciou nižšou ako je 16Hz (tento zvuk počuje napríklad slon) sa nazýva infrazvuk. Zvuk s frekvenciou vyššou ako 20kHz (delfíny a netopiere vnímajú zvuk s frekvenciami až 150kHz) nazývame ultrazvuk.

Zdroj zvukového vlnenia sa nazýva zdroj zvuku a prostredie v ktorom sa šíri je jeho vodič. Vodič zvuku, najčastejšie vzduch, sprostredkováva spojenie medzi zdrojom zvuku a jeho prijímačom(detektorom). V praxi ucho, mikrofón alebo iný snímač. Zvuk sa šíri aj kvapalinami a pevnými telesami. V kvapalinách a plynach sa šíri ako pozdĺžne vlnenie, v pevných látkach (napr. steny domu) sa môže šíriť aj vo vlnách priečných s výchylkou v kolmom smere na smer šírenia.

Zdrojom zvuku môže byť každé chvejúce sa teleso. Okrem telies kmitajúcich vlastnými kmitmi môžu byť zdrojom zvuku aj telesá kmitajúce kmitmi vynútenými. Patria medzi ne reproduktory, sluchátka pre generovanie alebo reprodukciu zvuku.

1.2 Druhy zvukov

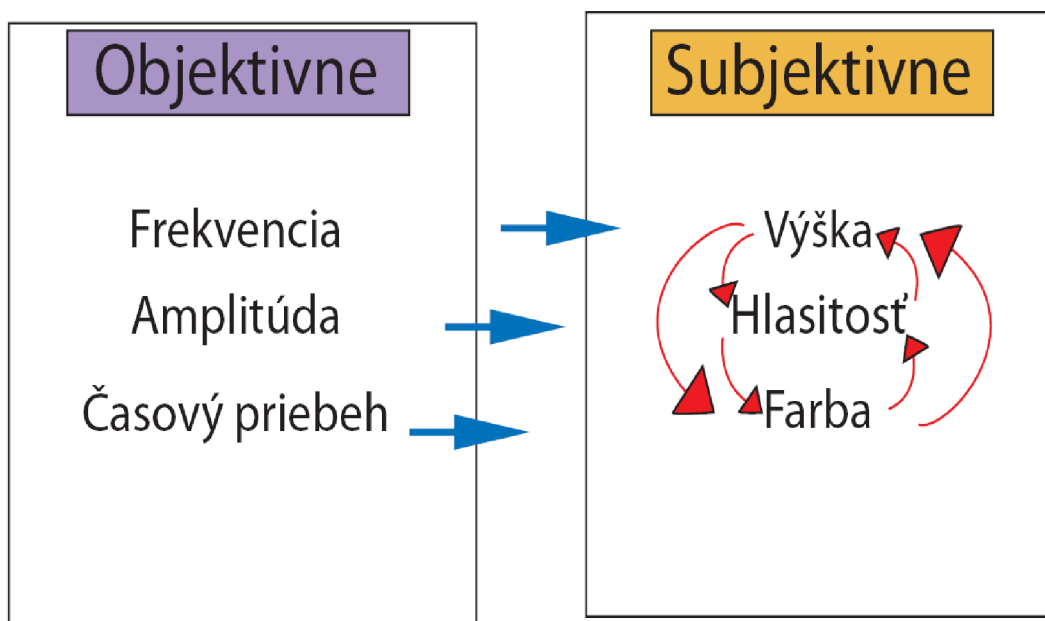
- ❖ *Hudobné (tóny)*
- ❖ *Nehudobné (hluky)*

Základný význam má rozdiel medzi zvukmi s charakterom tónovým a šumovým. Kritériom pre ich rozlíšenie je periodicita vlnenia. Tónové zvuky vynikajú dôsledkom periodického vlnenia. Zvuky šumové majú tvar vlny nepravidelný. Ďalší rozdiel predstavujú zvuky jednoduché a zložené. Jednoduchý je periodický zvuk sínusového charakteru. Každý zvuk, ktorý nemá sínusový charakter je hodnotený ako zložený. Jednoduchých tónov je v reálnom živote pomerne málo (napr. tón vydávaný ladičkou).

2 Vlastnosti zvuku

Zvuk vzniká periodickým stláčaním a rozpínaním pružného prostredia, v ktorom sa šíri. Vo vzduchu dochádza k zmenám atmosférického tlaku. Zvuk sa skladá z vln (kmitov) s nerovnakým tlakom. Tlak vytvárajú vo vzduchu naše hlasivky, hudobné nástroje alebo prírodné sily.

Vlastnosti tohto signálu môžeme deliť na objektívne a subjektívne. Objektívne vlastnosti matematicky popisujú fyzikálne vlastnosti daného zvukového signálu. Subjektívne vlastnosti vníma každý z nás inak. Subjektívne vlastnosti sú navzájom previazané a zmena jednej sa odzrkadlí na zmene druhej. Čo o objektívnych vlastnostiach neplatí. Každá objektívna vlastnosť má svoju obdobu v subjektívnych vlastnostiach (viď Obrázok 1).



Obrázok 1 : Objektívne a subjektívne vlastnosti zvuku

2.1 Amplitúda

Pri grafickom znázornení sínusovej vlny znamená amplitúda výšku. U signálov amplitúda zodpovedá výkonu použiteľnému pri prenose. Telekomunikační technici sa musia vyrovnať so skutočnosťou, že pri prenose dát na určitú vzdialenosť sa znižuje amplitúda signálu a dochádza k jeho zoslabeniu. V komunikačných systémoch udáva výkon alebo amplitúda dve informácie:

1. **Prenosový výkon komunikačného média**
2. **Pomer výkonu šumu k výkonu signálu**, tzv. odstup signál šum – jeho popisu a výpočtom sa podrobne venujem v kapitole 3.4.1

2.2 Frekvencia

Je vyjadrením rýchlosti periodicky prebiehajúceho deja. Udáva koľkokrát kmitne vlna hore a dole počas daného časového intervalu. Vyjadruje počet cyklov za sekundu. Jednotkou je 1Hertz (Hz). Ak hovoríme o zvuku, hovoríme o frekvenciách. Majme človeka s vysokým hlasom. Jeho hlas vytvárajú frekvencie v priemere o niečo vyššie ako je väčšina ľudských hlasov.

2.3 Výška

Výšku jednoduchého tónu udáva frekvencia. U jednoduchého tónu určuje frekvencia absolútnu výšku tónu. U zložených tónov, ktoré obsahujú zložky rôznych frekvencií, je výška daná základnou frekvenciou t.j. najnižšou.

Vzhľadom k tomu, že absolútnu výšku tónu väčšinou nedokážeme priamo určiť sluchom, zavádzame relatívnu výšku tónu. Je daná pomerom frekvencie daného tónu k frekvencií tónu základného. Ako základný tón bol v hudobnej akustike určený tón s frekvenciou 440 Hz. Je označovaný ako komorné a. V technickej praxi sa ako základný tón používa tón s frekvenciou 1 kHz. Takže relatívne určovanie výšky tónu znamená, že máme porovnať či daný tón je vyšší alebo nižší ako základný.

2.3.1 Vnímanie výšky zvuku

Vnímanie výšky zvuku do značnej miery závisí na základnej frekvencií opakovania zloženej vlny. Pri notách s frekvenciou menšou ako 1000 Hz vnímame zmenu len 2-3 Hz. Pri tónoch s frekvenciou okolo 7000 Hz vnímame zmenu až 14 Hz.

Zobrazovanie vnímanej výšky :

- Približne do 800 Hz vnímame rozdiely vo výške tónu lineárne
- Nad 800 Hz vnímame vzťah medzi dvoma výškami a skutočnou frekvenciou logaritmicky

Údaje v Hertzoch nezodpovedajú presne tomu, ako rozdiely vo frekvencií vnímame. Na určenie rozdielu používame **melovu** alebo **Barkovu** stupnicu, ktoré vnímané intervaly objektivizujú:

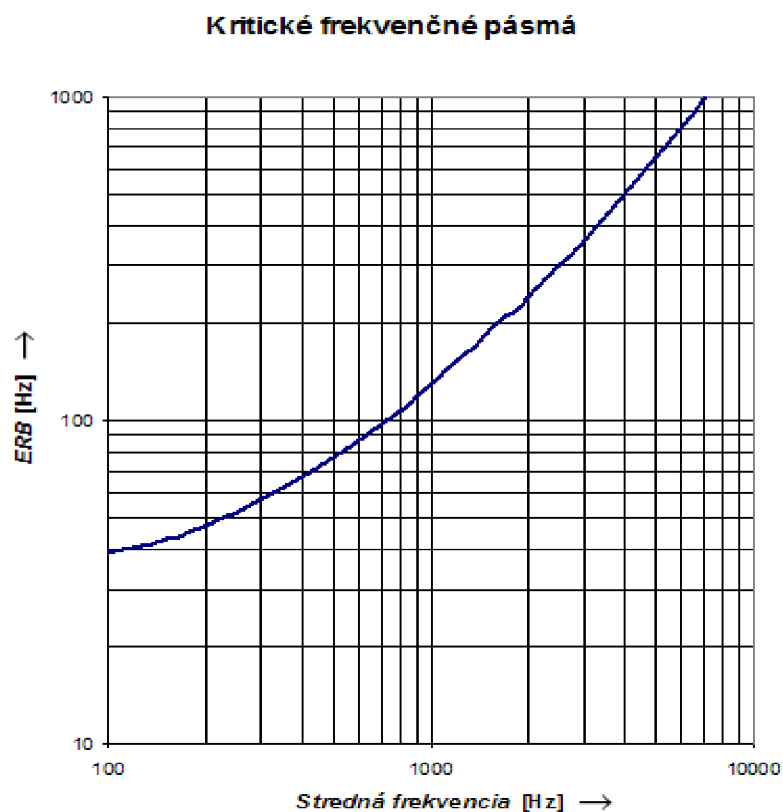
- **Mel**
 - Stupnica od 0 do 2400 melov
 - Dva zvuky, ktoré sú od seba vzdialené rovnaký počet melov sú rovnako vzdialené aj percepčne
 - 200 Hz = 200 mel, 500 Hz = 500 mel, 1000 Hz = 850 mel

- **Bark, kritické pásmo**

Ucho funguje ako frekvenčný analyzátor, každý bod baziliárnej membrány má svoju charakteristickú frekvenciu. Každý bod môžeme chápať ako pásmový filter so strednou frekvenciou, šírkou pásma a sklonom. To viedlo fyzikov k vytvoreniu špeciálnych frekvenčných skupín, tzv. **kritických pásiem počutia**. K vnímaniu hlasitosti prispievajú všetky spektrálne maximá, ale nie všetky rovnakou mierou, čo bolo základnou myšlienkou pre vytvorenie kritických pásiem. Celková hlasitosť, ktorú vnímame, je súčtom amplitúd jednotlivých frekvenčných skupín. Šírka frekvenčnej skupiny (kritického pásma počutia) sa líši v závislosti na frekvencii. Pod 500 Hz je šírka kritického pásma takmer 100 Hz, potom jeho šírka s frekvenciou rastie (tabuľka 1). Graf kritických frekvenčných pásiem je na Obrázok 2.

Jeden Bark zodpovedá šírke jedného kritického pásma. Barkova stupnica začína na 0 Hz (0 Bark) pokračuje v krokoch po 100 Hz. Od 500 Hz. sa šírka kritického pásma stále rozširuje. Okolo strednej frekvencie 1,6 kHz má Bark šírku pásma 240 Hz. Pri 13,5 kHz takmer 3,5 kHz. Veľkosť kritického pásma sa zisťuje v experimentoch s maskovaním signálového tónu šumom.

Cieľom je stanoviť frekvenčnú šírku pásma, pri ktorej by sme už tón v šume nedokázali rozpoznať. K takémuto maskovaniu dochádza v momente, kedy sa šírka pásma šumu rovná šírke pásma sluchového filtru. Táto šírka sa nazýva **kritické pásmo**.



Obrázok 2: Kritické frekvenčné pásma

Tabuľka 1: Rozdelenie frekvenčného rozsahu do rôznych kmitočtových pásiem

Kritické pásmo	Stredná frekvencia [Hz]	Šírka pásma [Hz]	Šírka pásma [%]	ERB [Hz]
1.	50	100	200	33
2.	150	100	67	43
3.	250	100	40	52
4.	350	100	29	62
5.	450	110	24	72
6.	570	120	21	84
7.	700	140	20	97
8.	840	150	18	111
9.	1000	160	16	130
10.	1170	190	16	150
11.	1370	210	15	170
12.	1600	240	15	200
13.	1850	280	15	220
14.	2150	320	15	260
15.	2500	380	15	300
16.	2900	450	16	350
17.	3400	550	16	420
18.	4000	700	18	500
19.	4800	900	19	620
20.	5800	1100	19	780
21.	7000	1300	19	990
22.	8500	1800	21	1300
23.	10500	2500	24	1700
24.	13500	3500	26	2400

2.4 Hlasitosť zvuku

Ak k nášmu sluchovému orgánu prichádzajú dva tóny s rovnakou akustickou intenzitou, ale s rôznou frekvenciou, nevnímame ich ako rovnako hlasné. Tento rozdiel v subjektívnom vnímaní hlasitosti tónov je spôsobený rôznou citlivosťou sluchového orgánu k rôznym akustickým frekvenciám. Závislosť približne vystihuje **Weber-Fecherov zákon**:

Ak rastie fyzikálna intenzita tónu aj danej frekvencie geometricky, jeho subjektívny účinok sa zväčšuje len aritmeticky [9].

Najvýznamnejší rozsah je 2-4 kHz, je najdôležitejší pre zrozumiteľnosť reči - na tieto frekvencie je ľudské ucho najcitlivejšie. Smerom k hornej a dolnej medzi citlivosť počutia klesá. Najvyššia informačná hodnota reči je prenášaná v pásme 0,5 – 2 kHz. Tieto

vlastnosti sú často využívané pri prenose reči. Nižšie frekvencie, na ktoré sme citlivejší, sú kódované na viac bitov ako ostatné frekvencie. Dosiachneme tým kompromis medzi kvalitou prenášaného signálu a počtom bitov, ktoré sú na prenos nutné.

2.4.1 Vnímanie hlasitosti

Silu zvuku najlepšie reprezentuje jeho **intenzita I**. Je to energia, ktorá za jednu sekundu dorazí na plochu kolmú k smeru šírenia vlnenia:

$$I_1 = \frac{E}{S * t} [J * m^{-2} * s^{-1}] \quad (2.1)$$

$$I_2 = \frac{P}{S} [W * m^{-2}] \quad (2.2)$$

Intenzita sa rovná mocnine akustického tlaku $I = p^2$. Pre široký rozsah ľudského ucha bola vytvorená logaritmická decibelová stupnica vyjadrujúca pomer dvoch intenzít $10 \log \frac{I_1}{I_0} [dB]$, s príkladmi v Tabuľka 2.

Tabuľka 2: Pomerné intenzity zvuku v dB

I_1/I_0	$\text{Log}(I_1/I_0)$	rozdiel v dB
1000:1	3	30
100:1	2	20
10:1	1	10
1:1	0	0
1:2	-0,3	-3
1:4	-0,6	-6

Referenčná hodnota pre 0dB je 10^{-12}W/m^2 (intenzita tónu 1000Hz je tesne pod prahom počutia). Takémuto vyjadreniu pripadá termín hladina akustického výkonu.

Decibelová stupnica sa používa pre vyjadrenie pomeru intenzity aj akustického tlaku $I = p^2$

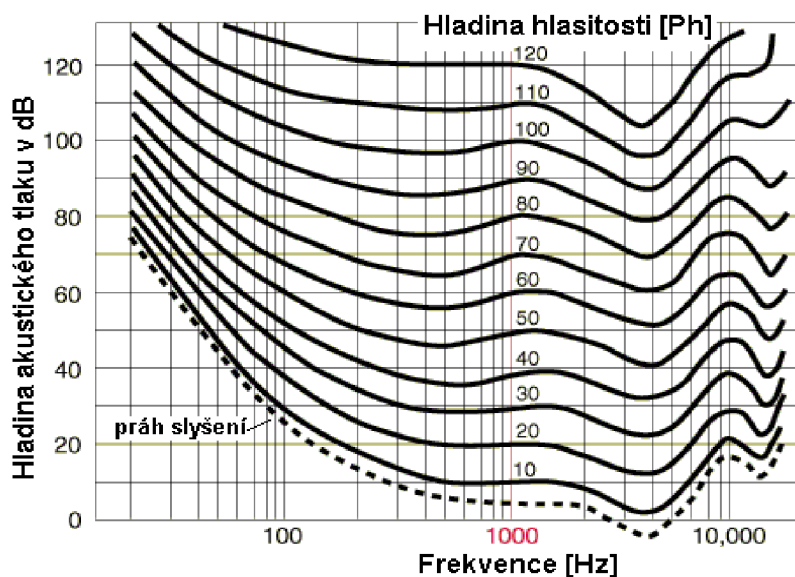
$$B = 10 \log \frac{I_1}{I_0} = 10 \log \left(\frac{p_1}{p_2} \right)^2 [dB] \quad (2.3)$$

Desaťnásobné zvýšenie akustického tlaku zodpovedá stonásobnému zvýšeniu intenzity a +20dB.

2.4.2 Meranie subjektívnej hlasitosti

- Hladina hlasitosti H : jednotkou je 1fón (1Ph), je zhodná s hladinou akustického tlaku vyjadreného v dB iba pre referenčnú frekvenciu 1kHz, berie do úvahy rôznu citlivosť sluchu v celom akustickom pásme. Hladina hlasitosti 1 fón zodpovedá hladine intenzity 1 dB pre frekvenčný tón s frekvenciou 1 kHz. Vypovedá nám

o tom, akú intenzitu musí mať tón s frekvenciou 1000 Hz, aby sme ho vnímali rovnako hlasito. Intenzita tónu o frekvencii 1000 Hz v decibeloch je hladinou hlasitosti daného signálu. Ak signál vnímame rovnako hlasito ako 1000 Hz tón s intenzitou 55 dB, potom $H = 55 \text{ Ph}$. Ľudské ucho počuje najlepšie v strednom pásme okolo 3000 – 4000 Hz. Prah počuteľnosti je niekde okolo 4,2 dB. Prah bolesti sa pre takmer všetky frekvencie pohybuje okolo 120-130 dB (v závislosti na jedincovi). Aby sa nám zvuk javil hlasnejší o 1Ph, musí jeho intenzita vzrásť o 26%, čo je minimálny postrehnutelný rozdiel. Na Obrázok 3 sú znázornené grafické závislosti akustického tlaku od frekvencie pri rôznych hladinách hlasitosti.



Obrázok 3: Závislosť hladiny akustického tlaku od frekvencie pri rôznych hladinách hlasitosti[10].

V **Tabuľka 3** sú uvedené príklady vybraných zvukov.

Tabuľka 3: Vybrané zvuky a ich hladiny

Zvuk	B[dB]	L[son]
Ticho	0	0,06
Tiché dýchanie	10	0,13
Priemerný hluk v byte	40	1
Tichá konverzácia	40-60	1-4
Hlučná ulica	70-90	8-32
Krik	80	16
Rockový koncert	130	512
Tryskové lietadlo	150	2048
Raketový motor	180	16384

Určovanie hlasitosti vo fónoch je z hudobného hľadiska nepraktické, vyjadruje iba hlasitosť pri posluhu jedinej frekvencie. Preto bola zavedená veličina **Hlasitosť** L , ktorá

zodpovedá celkovému podráždeniu sluchového orgánu. Jednotkou hlasitosti je **1 son** – je definovaný ako hlasitosť tónu s frekvenciou 1000Hz a intenzitou 40dB. Pre iné frekvencie sa musí hodnota prepočítať v závislosti na frekvenčnej citlivosti.

Zvýšenie intenzity o 10 dB zodpovedá zdvojnásobeniu hlasitosti v sonoch. Tón ($f = 1000 \text{ Hz}$, $I = 60 \text{ dB}$) vnímame trikrát hlasnejší ako 40 dB čiže $L = 3 \text{ sony}$. Umožňuje, na rozdiel od hladiny hlasitosti, jednoduchšie určiť hlasitosť niekoľkých zvukov. Vystihuje subjektívnu mieru hodnotenia zvuku. Zvuk s hlasitosťou $L_1 = 2 \text{ sony}$ je dvakrát hlasitejší ako zvuk s hlasitosťou $L_2 = 1 \text{ son}$. Jedná sa o experimentálne zavedenú jednotku.

2.5 Farba

Farba zvuku reprezentovala už od svojich prvých definícií subjektívnu predstavu kvality zvuku.

Farba tónu závisí na počte harmonických zložiek, ktoré znejú spoločne s daným základným tónom. Závisí aj na veľkosti amplitúd a šumoch doprevádzajúcich základný tón.

Farba zvuku umožňuje subjektívne rozlíšiť tóny rovnakej výšky, ktoré vydávajú rôzne zdroje, napríklad hudobné nástroje. Hovoríme, že ľudské hlasy majú rôzne zafarbenie.

Podľa Fourierovej teórie môžeme ľubovoľnú periodickú funkciu s periódou T zapísať ako lineárnu kombináciu harmonických funkcií s periódami T/n , kde n je prirodzené číslo. Práve koeficienty v tejto lineárnej kombinácii matematicky popisujú farbu tónu. Frekvencia $f = 1/T$ sa nazýva frekvencia základná, frekvencie zodpovedajúce prirodzeným násobkom tejto základnej frekvencie nazývame vyššími harmonickými frekvenciami. Výsledok analýzy nazývame frekvenčná analýza.

Amplitúda vyšších harmonických tónov je rôzna, ale podstatne menšia ako amplitúda tónu základného. Výsledný zvuk je vplyvom vyšších harmonických tónov pre daný zvuk charakteristický. Čím viac takýchto harmonických zložiek daný tón obsahuje a čím výraznejšie tieto zložky znejú, tým je farba tónu plnšia, sytejšia. V prípade, že z vyšších harmonických tónov sú silné len niektoré, zvuk získava prenikavosť a lesk (napr. zvuk huslí). Tóny obsahujúce len málo vyšších harmonických tónov, znejú duto.

3 Kvalita audio signálov

Algoritmy testovania kvality audio signálov sa obecné delia do dvoch kategórií:

- Subjektívne hodnotenie
- Objektívne hodnotenie

Subjektívne hodnotenia sú založené na porovnávaní pôvodného a spracovaného audio- signálu skupinou poslucháčov, ktorí hodnotia kvalitu a zrozumiteľnosť signálu podľa preddefinovanej stupnice. Objektívne hodnotenia sú založené na matematickom porovnaní pôvodného a spracovaného signálu.

Hlavný rozdiel medzi pojmami kvalita a zrozumiteľnosť je v tom, že pod pojmom kvalita rozumieme „ako“ poslucháč vníma audio signál (ako znie spracovaný signál). Pod pojmom zrozumiteľnosť „čo“ poslucháč počuje (či to čo počuje má aj informačný zmysel). Zrozumiteľnosť predstavuje len jeden z aspektov kvality, veľmi kvalitný audio signál je s veľkou pravdepodobnosťou zrozumiteľný. Opačne to samozrejme môže, ale nemusí platiť. Testy kvality sú používané k vyhodnocovaniu systémov s vysokou zrozumiteľnosťou. Naopak nízka zrozumiteľnosť je predzvesťou nízkej kvality.

3.1 Zrozumiteľnosť

Všeobecne zrozumiteľnosť prijímaného signálu závisí na hladine akustického tlaku a uchu poslucháča, charaktere vnímaného zvukového signálu a na úrovni okolitého hluku. Zrozumiteľnosť vzrastá so zvyšujúcou sa intenzitou na hladine 50 až 60 dB nad prahom zrozumiteľnosti a dosahuje u osôb bez porúch sluchu 100%. Na vysokých hladinách nad prahom dochádza opäť k znižovaniu zrozumiteľnosti. Za prah zrozumiteľnosti sa obvykle považuje hladina kedy počujeme, ale nerozumieme. Zrozumiteľnosť klesá tiež s klesajúcim odstupom užitočného signálu od okolitého hluku (šumu) a so stúpajúcou dobou dozvuku.

3.2 Subjektívna kvalita a zrozumiteľnosť

Subjektívne sa zrozumiteľnosť meria štatistickými metódami. V najjednoduchších prípadoch sa percentuálne vyjadruje počet správne rozpoznávaných prvkov. Podľa povahy prvkov použitých pre vyšetrenie sa odlišuje rozoznateľnosť fonémov, difónov, alofónov, slabík, hlások a nakoniec celková rozoznateľnosť slovná či vetná. Zrozumiteľnosť fonémová je horšia ako hlásková, tá je zreteľne horšia ako slovná a je horšia ako vetná, pretože vo vyšších útvaroch je možné doplniť chýbajúcu časť prvku z kontextu.

Subjektívne meranie kvality môžeme rozdeliť na merania **úžitkové a analytické**. **Úžitkové merania kvality** sa porovnávajú s jednodimenzionálnou škálou. Výsledkom je jedno číslo. **Analytické metódy** sa snažia identifikovať psychoakustické javy určujúce vnímanú kvalitu. Tieto metódy sú orientované skôr na charakterizovanie vnímania daného signálu ako na hodnotenie jeho kvality.

Verbálny popis charakterizuje vlastnosti zvuku výrazmi. Tieto výrazy môžeme zostavovať do protikladných pojmových párov. (napr. silný – slabý, tupý – ostrý, plný - dutý). Tieto páry poskytujú podstatným spôsobom subjektívny vnem signálu. K stanoveniu navzájom ortogonálnych pojmových párov je možné využiť metódy štatistickej analýzy v tomto prípade metódu multidimenzionálneho škálovania.

3.3 Subjektívne metódy merania kvality audiosignálov

Subjektívne metódy merania kvality môžeme rozdeliť do dvoch kategórií:

1. metódy založené na relatívnej preferencii:

Pri hodnotení kvality metódou relatívnej preferencie sa poslucháčom prehrávajú páry úsekov audiosignálu obsahujúce referenčný úsek reči a testovaný úsek signálu. Referenčný úsek je vytvorený degradovaním signálu (napr. aditívnym šumom). Poslucháči potom volia ten úsek, ktorý sa im zdá kvalitnejší.

2. metódy založené na priradení číselnej hodnoty kvalite:

V testoch založených na hodnotení kvality je poslucháčom prehraný úsek signálu a ich úlohou je ohodnotiť jeho kvalitu na stupnici. Najčastejšie sa využíva päťhodnotová stupnica, kde 1 je slabá kvalita a 5 výborná kvalita. Pri tomto hodnotení sa nepoužíva žiadny referenčný signál.

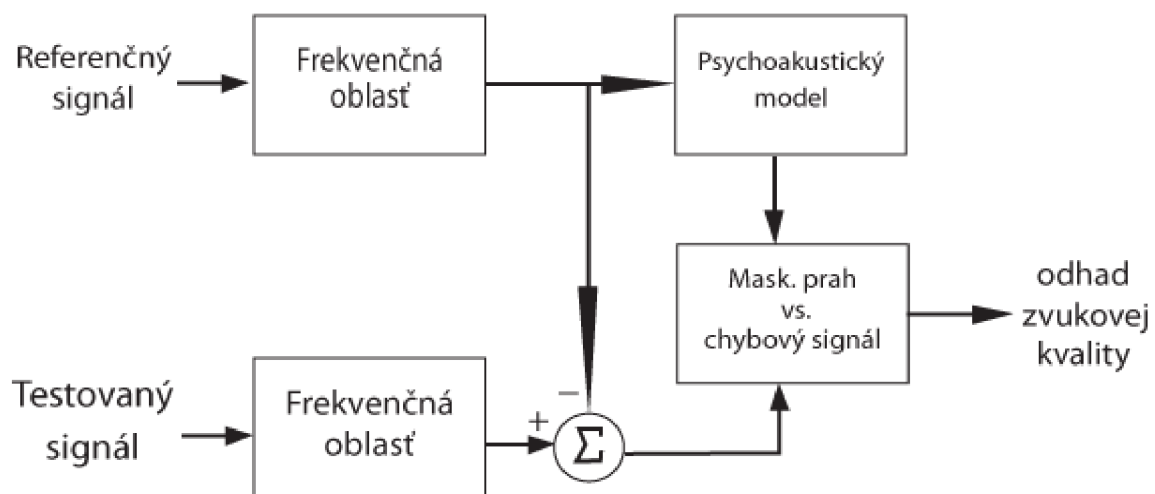
Metóda MOS (Mean Opinion Score) je metódou absolútneho hodnotenia. Poslucháči hodnotia reč na päťstupňovej stupnici, kde je dojmu poslucháča priradená číselná hodnota (Tabuľka 4). Výhodou MOS je, že poslucháči môžu spracovanému signálu ľubovoľne priradiť svoj dojem kvality. Nevýhodou je naopak, že dojem kvality je u poslucháčov veľmi rozdielny.

Tabuľka 4: Stupnica metódy MOS

Hodnotenie	Kvalita signálu	Úroveň skreslenia
1	Nedostatočná(Bad)	veľmi nepríjemné až protivné
2	Zlá(Poor)	Nepríjemné
3	Dostatočná(Fair)	rozoznateľné trochu nepríjemné
4	Dobrá(Good)	rozoznateľné ale nie nepríjemné
5	Vynikajúca(Excelent)	nerozoznateľné

3.4 Objektívna kvalita zrozumiteľnosti

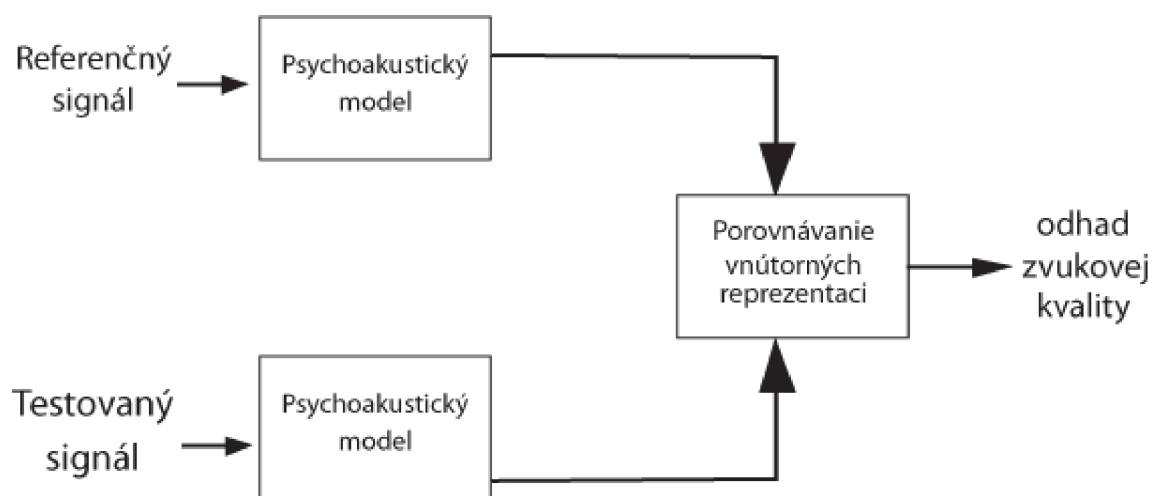
Najväčšou výhodou objektívnych metód je, že nezahŕňajú ľudský faktor a s ním spojené nedostatky, ktorými trpia subjektívne metódy. Objektívne metódy sú vhodné najmä v prípadoch, kedy je nutné vykonať veľký počet meraní, napr. pri nastavovaní jeho optimálnych parametrov. Subjektívne merania sú v takýchto prípadoch časovo veľmi náročné. Objektívne meranie kvality audio signálu sú dobrými prediktormi subjektívnej kvality signálu. Obrázok 4 popisuje objektívnu metódu, ktorá spočíva v určení chybového signálu a jeho porovnaní s vypočítanými maskovacími prahmi.



Obrázok 4: Porovnanie chybového signálu s maskovacím prahom

Pri tejto metóde je nutné brať na vedomie, že stanovenie maskovacích prahov nemusí byť dostatočne presné, metóda neposkytuje informáciu i mieru rušivosti chybového signálu. Dozvieme sa len, či je skreslenie počuť, alebo nie. V prípade, že sú zmeny počuteľné nie je možné odhadnúť ich dopad na celkový vnem kvality.

Lepšie je porovnávať vnútorné reprezentácie oboch vstupov pomocou psychoakustického modelu, ktorý zohľadní prenos zvuku vnútorným a stredným uchom a ďalšie vplyvy, odhadneme pôsobenie každého vstupu na vlastný sluchový receptor (Cotriho orgán). Potom pomocou kognitívneho modelu vygeneruje jediný údaj vypovedajúci o kvalite (Obrázok 5). Hlavnou nevýhodou je vysoká náročnosť zostrojenia psychoakustického modelu a vymodelovania vlastného Cotriho orgánu.



Obrázok 5: Porovnanie vnútorných reprezentácií signálu

3.4.1 Artikulačný index

Artikulačný index je pokusom o kvantifikovanie subjektívneho kritéria akustickej kvality nazývaného zrozumiteľnosť. Objektívne meranie artikulačného indexu vychádza z jeho merania subjektívneho. Zrozumiteľnosť je veľmi dobrá pri AI = 100% a naopak veľmi zlá ak sa blíži k 0%.

Jednou z metód ako merať artikulačný index je vypočítať v každom z L frekvenčných pásiem odstup signál šum (SNR) a spriemerovať merania. Merania SNR v každom frekvenčnom pásme musí byť obmedzené prahom počuteľnosti a prahom bolesti. Hodnota artikulačného AI pre L pásiem je definovaná vzťahom:

$$AI = \frac{1}{L} \sum_{j=1}^L \frac{\min[SNR_j, 30]}{30} \quad (3.1)$$

Kde SNR je obmedzené na 30dB.

Artikulačný index sa podľa normy ANSI S3.5-1997[11] určuje pre 4 rôzne referenčné oblasti signálu, ktoré sa líšia počtom frekvenčných pásiem.

- Kritické pásma (21pásiem)
- Tretinooktávové pásma(18pásiem)
- Dôležité kritické pásma (17pásiem)
- Oktávové pásma (6pásma)

Meranie odstup signál šum – SNR (Sound to noise ratio)

Meranie odstup signálu od šumu je najčastejšie používané pre analógové a digitálne systémy kódovania rečových signálov a v poslednej dobe je čoraz častejšie používané pre hodnotenie algoritmov zvýraznenia reči určených pre širokopásmové skreslenia.

Ak označíme zašumený číslicový audiosignál $y[n]$, jeho ekvivalent bez šumu $s[n]$ a $\hat{s}[n]$ zodpovedajúci spracovávaný signál, môžeme chybový signál vyjadriť ako:

$$\varepsilon[n] = s[n] - \hat{s}[n] \quad (3.2)$$

Jeho energiu:

$$E_\varepsilon = \sum_{n=-\infty}^{\infty} \varepsilon^2[n] = \sum_{n=-\infty}^{\infty} (s[n] - \hat{s}[n])^2 \quad (3.3)$$

Energia obsiahnutá v samotnom audiosignále je:

$$E_s = \sum_{n=-\infty}^{\infty} s^2[n] \quad (3.4)$$

Výsledná hodnota SNR je potom:

$$SNR = 10 \log_{10} \frac{E_s}{E_e} \quad (3.5)$$

Toto meranie sa využíva najmä v situáciách kedy máme k dispozícii degradovaný a nezašumený signál. Hlavnou výhodou merania kvality signálu na základe SNR je jeho matematická nenáročnosť.

Energia audio signálu je veľmi časovo premenlivá. Pokiaľ predpokladáme, že skreslenie je širokopásmové s malým kolísaním energie, potom sa bude SNR líšiť rámc od rámc. Značne presnejšie meranie získame, ak budeme SNR merať v krátkych časových intervaloch a výsledky spriemerujeme. Toto meranie sa nazýva Segmental SNR (SNR_{seg}), matematicky je popísané rovnicou:

$$SNR_{seg} = \frac{1}{M} \sum_{j=0}^{M-1} 10 \log_{10} \left[\frac{\sum_{n=m_j-N+1}^{m_j} s^2[n]}{\sum_{n=m_j-N+1}^{m_j} (s[n] - \hat{s}[n])^2} \right] \quad (3.6)$$

Kde m_0, m_1, \dots, m_{M-1} sú koncové časy M rámcov, každý o dĺžke N . Pre každý rámc (typicky 15 – 30 ms) je vypočítané SNR a výsledok je získaný spriemerovaním výsledkov všetkých segmentov.

V niektorých prípadoch vznikne (pokiaľ je niektorý rámc prázdny) ticho, čo znamená veľký záporný výsledok. Tomuto problému sa dá vyhnúť detekovaním tichých pasáží a ich vyradeniu z výpočtu SNR. Iným riešením je nastavenie dolnej medznej hodnoty. Signály s hodnotou SNR vyššou ako 35dB sú poslucháčom vnímané ako temer rovnaké. Preto sa výsledky obmedzujú aj zhora (bežne 30-35dB).

Ďalšou modifikáciou výpočtu SNR je metóda Frequency Weighted Segmental SNR (SNR_{fw-seg}), ktorá zisťuje segmentovú hodnotu SNR vo frekvenčných pásmach zodpovedajúcich kritickým pásmam. Na každé pásmo je aplikovaná séria šumovo závislých percepčných váhových koeficientov w_{jk0} , vďaka čomu zistená hodnota SNR viac zodpovedá poslucháčovmu vnemu kvality. Matematicky ho popisuje rovnica:

$$SNR_{fw-seg} = \frac{1}{M} \sum_{j=0}^{M-1} 10 \log_{10} \left[\frac{\sum_{k=1}^K w_{j,k} 10 \log_{10} (E_{s,k}[m_j] / E_{e,k}[m_j])}{\sum_{k=1}^K w_{j,k}} \right] \quad (3.7)$$

Kde M je počet rámcov, K počet frekvenčných pásiem, $E_{s,k}[m_j]$ krátkodobú energiu v k -tom frekvenčnom pásme rámcu bez šumu a $E_{e,k}[m_j]$ zodpovedá množstvu šumu $\varepsilon[n]$.

3.4.2 Meranie založené na LPC

Veľká skupina objektívnych metód merania kvality využíva rozdiel autoregresívnych (AR) modelov čistého a zvýrazneného signálu.

✓ **Autoregresívne modelovanie signálov:**

Najznámejšia z metód modelovania signálu je pomocou autoregresie (AR). Táto metóda obsahuje výpočet rádu a koeficientov autoregresívneho filtra aplikovaného na vstupný signál tak, aby došlo k minimalizácii predpokladanej náhodnej chyby časového priebehu podľa špecifického kritéria. Nech $x(n)$ je časový signál vzorkovaný v intervaloch ΔT . P je rád a $a(i)$, $i=1..P$ koeficienty autoregresie predpokladaného procesu. Potom:

$$x(n) = \sum_{i=1}^P a(i)x(n-i) + e(n), P \leq n \leq N \quad (3.8)$$

z čoho

$$e(n) = x(n) - \sum_{i=1}^P a(i)x(n-i) \quad (3.9)$$

je jeden krok predpokladanej chyby

Rád P a koeficienty $a(i)$, $i=1..P$ sú také, aby daná funkcia bola minimalizovaná. Koeficienty $a(i)$ môžu byť opakovane testované metódou najmenších štvorcov odchýlok alebo najväčšou pravdepodobnosťou. Z tohto modelu môžeme vyjadriť spektrum signálu ako.

$$X(f) = \frac{1}{1 - \sum_{i=1}^P a(i)e^{j2\pi f \Delta T}} \quad (3.10)$$

Vlastnosti autoregresívnej techniky modelovania (AR) z pohľadu časovo premennej spektrálnej analýzy sú:

- Výpočet nie je skreslený efektmi postranných lalokov
- Pre štatisticky spoľahlivý odhad musí byť počet vzorkou značne vyšší ako rád modelu P
- Vysoká rozlišovacia schopnosť metódy AR je značne závislá na prítomnosti šumu. Keď sa pomer signál / šum znižuje, je potrebné k získaniu rovnakého rozlíšenia vyšší rád modelu. Ale príliš vysoký rád spôsobuje chyby vo výpočtovom spektre.
- AR – modely nie sú príliš vhodné pre identifikáciu jednoduchých (sínusových) spektrálnych zložiek. V oblasti modelovania sú výsledky takýchto jednoduchých zložiek póly na jednotkovej kružnici a vedú k stanoveniu chybných rovníc [6].

Ďalšie metódy:

1. metóda logaritmickej podobnosti (Log-likelihood Ratio LLR)

táto metóda definuje činiteľ vzdialenosti ako

$$d_{LLR}(\mathbf{a}_x, \bar{\mathbf{a}}_x) = \log \frac{\bar{\mathbf{a}}_x^T \mathbf{R}_x \bar{\mathbf{a}}_x}{\mathbf{a}_x^T \mathbf{R}_x \mathbf{a}_x} \quad (3.11)$$

kde $\mathbf{a}_x^T = [1, -\alpha_x(1), -\alpha_x(2), \dots, -\alpha_x(p)]$ sú LPC koeficienty čistého signálu, $\bar{\mathbf{a}}_x^T = [1, -\alpha_{\hat{x}}(1), -\alpha_{\hat{x}}(2), \dots, -\alpha_{\hat{x}}(p)]$ koeficienty zvýrazneného signálu a \mathbf{R}_x je autokorelačná (Toeplitzova matica) pôvodného signálu. Vo frekvenčnej oblasti by táto rovnica mala tvar:

$$d_{\text{LLR}}(\mathbf{a}_x, \bar{\mathbf{a}}_x) = \log \left(1 + \int_{-\pi}^{\pi} \left| \frac{A_x(\omega) - \bar{A}_x(\omega)}{A_x(\omega)} \right|^2 d\omega \right) \quad (3.12)$$

Kde $A_x(\omega)$ a $\bar{A}_x(\omega)$ sú spektrá LPC koeficientov čistého a zvýrazneného signálu.

2. metóda Long-Area-Ratio Measure (LAR)

Metóda LAR je tiež založená na rozdielnosti LPC koeficientov medzi originálnym a spracovaným signálom. Parametre metódy long-area-ratio sú postupne získané z LPC koeficientov pôvodného $r_{\Phi}(j)$ a spracovaného $r_d(j)$ signálu vo frekvenčnom pásme j .

Činiteľ vzdialenosti d_{LAR} je definovaný:

$$d_{\text{LAR}} = \left| \frac{1}{M} \sum_{i=1}^M \left[\log \frac{1 + r_{\Phi}(j)}{1 - r_{\Phi}(j)} - \log \frac{1 + \hat{r}_d(j)}{1 - \hat{r}_d(j)} \right] \right|^{\frac{1}{2}} \quad (3.13)$$

[15]

3.4.3 Metódy založené na percepčných vlastnostiach

Predchádzajúce metódy sú relatívne ľahko implementovateľné, veľmi rozšírené. Ich hlavným nedostatkom je fakt, že neberú ohľad na percepčné vlastnosti sluchu. Výskumu metód, ktoré zohľadňujú vnímanie reči ľudským sluchom je venovaná veľká pozornosť. Tieto metódy sa opierajú najmä o to, že vnímanie reči, hudby alebo obecných audio signálov ľudským sluchom nie je lineárne. Tieto metódy zohľadňujú psychoakustické vnímanie.

a. Metóda Weighted Spectral Slope

Psychoakustické pokusy ukázali, že pri hodnotení samohlások, na ktorých boli prevádzané rôzne spektrálne manipulácie, hodnotia poslucháči ako najviac skreslené tie, čo sa líšia vo frekvenčných formantoch. Preto boli navrhnuté merania založené na váhových rozdieloch medzi strmou spektrálnej funkcie v každom pásme. Tento typ merania najviac potláča rozdiely v pozíciách formantov a ignoruje ďalšie rozdiely.

Najprv je v každom frekvenčnom pásme vypočítaná strmota. Označme spektrum kritických pásiem pôvodného signálu $E_s(f)$ a spektrum kritických pásiem zvýrazneného signálu $E_{\hat{s}}(f)$, strmoti jednotlivých pásiem budú dané rovnicou:

$$\begin{aligned}\Delta E_s(f) &= E_s(f+1) - E_s(f) \\ \Delta E_{\hat{s}}(f) &= E_{\hat{s}}(f+1) - E_{\hat{s}}(f),\end{aligned}\tag{3.14}$$

Kde $\Delta E_s(f)$ a $\Delta E_{\hat{s}}(f)$ označujú strmosti f -teho pásma. Vypočítané strmosti sú váhované podľa toho, či sa nachádzajú v blízkosti lokálneho maxima, alebo minima spektra a či sa prípadne jedná o najväčšie maximum spektra. Celková váhová konštanta je daná rovnicou:

$$W(f) = \frac{W_s(f) + W_{\hat{s}}(f)}{2}\tag{3.15}$$

Kde

$$W_s(f) = \frac{20}{20 + E_{s,\max} - E_s(f)} \frac{1}{1 + P_s(f) - E_s(f)}\tag{3.16}$$

$$W_{\hat{s}}(f) = \frac{20}{20 + E_{\hat{s},\max} - E_{\hat{s}}(f)} \frac{1}{1 + P_{\hat{s}}(f) - E_{\hat{s}}(f)}\tag{3.17}$$

Lokálne maximum $P(f)$ hľadáme smerom nahor od aktuálneho pásma ak $\Delta E_s(f) > 0$, smerom nadol ak $\Delta E_s(f) < 0$

Celková hodnota WSSM sa vypočíta pomocou rovnice:

$$WSSM = \frac{1}{N} \sum_{k=1}^N \left[\frac{\sum_{f=1}^{24} W(f) [\Delta E_s(f) - \Delta E_{\hat{s}}(f)]^2}{\sum_{f=1}^{24} W(f)} \right]\tag{3.18}$$

Kde $W(f)$ je váhový koeficient f -teho pásma a N je počet rámcov na ktoré je signál rozdelený. Výsledná hodnota WSSM je spočítaná ako priemer všetkých hodnôt získaných zo všetkých rámcov spektra. Čím je hodnota WSSM nižšia tým je signál zrozumiteľnejší čiže kvalitnejší, menej degradovaný. Naopak so stúpajúcou hodnotou WSSM sa degradácia signálu výraznejšie prejavuje na kvalite signálu. Metóda WSS má väčšiu koreláciu so subjektívnym hodnotením kvality ako metódy založené na LPC a výrazne lepšiu koreláciu ako metódy ktorých základom je meranie odstupe signál šum.

b. Meranie skreslenia v kritických pásmach

Na základe psychoakustických poznatkov o vnímaní úzkopásmových šumov sluchom je navrhnutých niekoľko objektívnych meraní kvality audiosignálu využívajúce model Počutia.

Ten zohľadňuje:

1. frekvenčné rozlíšenie nie je uniformné
2. citlivosť sluchu je frekvenčne závislá

3. hlasitosť je na intenzite signálu závislá nelineárne

Po vypočítaní spektrálnych koeficientov a výkonového spektra sa pokračuje nasledujúcimi transformáciami:

- filtrácia signálu v kritických pásmach: výpočet výkonu v kritickom pásme sčítaním zložiek výkonového spektra signálu v danom kritickom pásme. Získavame tzv. barkové spektrum signálu v danom kritickom pásme $B(b)$
- Preemfáza podľa kriviek rovnakej hlasitosti : prevod intenzity zvukového signálu v dB na fóny pomocou filtru s prenosovou funkciou $H(z) = \left(\frac{2,6 + z^{-1}}{1,6 + z^{-1}} \right)$ alebo pomocou vypočítaných kriviek rovnakej hlasitosti. Po konverzií získame nové spektrum $P(b)$ vo fónoch
- Kompenzácia subjektívnej hlasitosti: prevod fónov na sóny

$$S(b) = \begin{cases} 2^{(P(b)-40)/10} & \text{pre } P(b) \geq 40 \\ \left(\frac{P(b)}{40} \right)^{2,642} & \text{pre } P(b) < 40, \end{cases} \quad (3.19)$$

Kde $S(b)$ nám udáva spektrum hlasitosti v sónoch

Hodnota BSD pre rámec k je definovaná ako rozdiel medzi spektrami hlasitosti:

$$BSD(k) = \sum_{i=1}^K [S_x(b) - S_y(b)]^2 \quad (3.20)$$

Kde K je počet kritických pásiem a $S_x(b)$ a $S_y(b)$ sú spektrá hlasitosti čistého a spracovávaného signálu. Stredná hodnota BSD je spočítaná ako priemer BSD všetkých rámcov.

Vylepšením metódy BSD vznikla metóda Modified Bark Spectrum Distostion(MBSD), ktorá iba zavádza do výpočtu výslednej hodnoty vzdialenosť váhovacej funkcie $W(b)$.

$$BSD(k) = \sum_{i=1}^K W(b) [S_x(b) - S_y(b)]^2 \quad (3.21)$$

$W(b)$ nadobúda hodnoty 0 alebo 1 podľa toho či signál v danom kritickom pásme je maskovaný alebo nie, resp. či prekračuje hodnotu vypočítanej maskovacej krivky alebo nie[7].

3.4.4 Metóda PSQM

Metóda PSQM (Perceptual Speech Quality Measurment). Podrobne popísaná v doporúčení ITU-T P.861. Algoritmus na výpočet kvality prenosu rečového signálu na základe modelu ľudského vnemu. Skladá sa z dvoch častí:

1. Meranie kvality hlasových kodekov
2. Meranie kvality prenosu hlasu v celom telekomunikačnom reťazci.

Princíp spočíva v porovnaní amplitúd výkonových spektier v navzájom si odpovedajúcich úsekoch pôvodného a preneseného signálu. Dochádza k synchronizácií, eliminuje sa tak možné oneskorenie paketu, vzniknuté pri prenose sieťou. Časové zarovnanie používané touto metódou je však iba veľmi hrubé a nevhodné pre merania na zašumených linkách. Rečový signál je delený na rámce s dĺžkou 16ms s 50% prekrytím susedných rámcov. Rozlišujú sa úseky aktívneho hovoru (speech periods) a úseky ticha (silent periods). Každá skupina je hodnotená samostatne pri posudzovaní vplyvu na šum. Na záver je toto hodnotenie kombinované do výsledného parametru. Ak výsledný parameter nadobúda hodnotu 0 vyjadruje dokonalú zhodu medzi vstupným a výstupným signálom, hovoríme o ideálnej zrozumiteľnosti. Vyššie hodnotenie reprezentuje stúpajúcu intenzitu skreslenia.

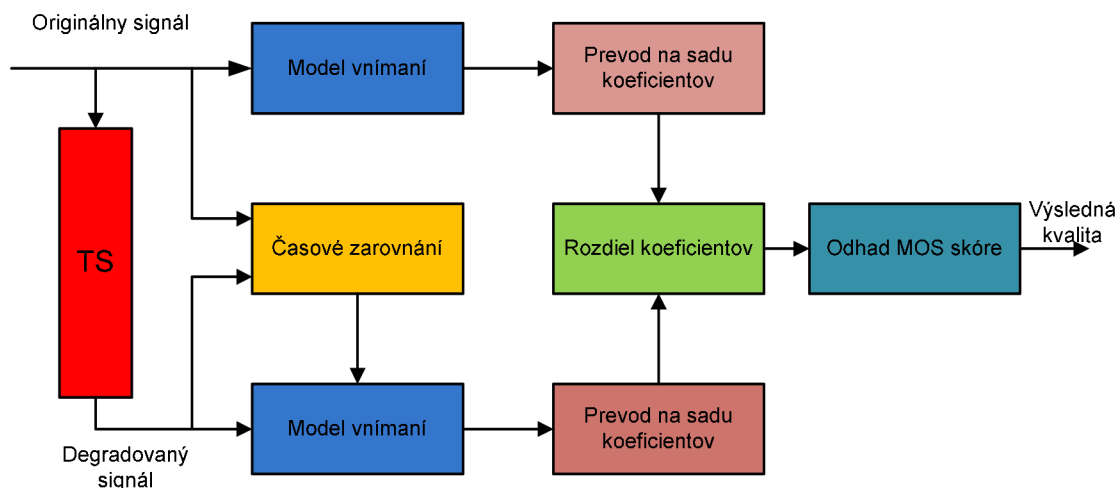
3.4.5 Metóda PESQ

Metóda PESQ (Perceptual Evaluation of Speech Quality) patrí medzi intrusívne objektívne metódy. Podrobne je popísaná v doporčení ITU-T P.862 [13]. Základná bloková schéma metódy je na Obrázok 6.

Jeden z problémov, ktoré prináša IP telefónia spočíva v skutočnosti, že IP prenosy sú paketovo orientované, čiže prenášajú sa skupiny bitov, ktoré môžu do cieľového miesta putovať odlišnými cestami s rozdielnym oneskorením. To spôsobuje premenlivé oneskorenie rekonštruovaných úsekov signálu (čo môžeme do určitej miery kompenzovať, ukladaním dát do pomocného registra na prijímacej strane, veľkosť tohto registra však nemôžeme ľubovoľne zväčšovať).

Žiadny z popísaných algoritmov nám neumožňuje meranie kvality týchto hlasových prenosov, lebo aj malé zmeny v oneskorení počas prenosu jedného hlasového vzorku vedú k zlyhaniu časovej synchronizácie. Okrem tohto problému, pri tomto type dátových prenosov vzrastá počet krátkodobých výpadkov, ktoré uvedené algoritmy tiež nehodnotia v súlade s ľudským vnímaním – toto hodnotenie je veľmi ťažko algoritimizovateľné a je značne individuálne. Tieto nedostatky viedli k navrhnutiu nových algoritmov. Britská skupina British Telecom navrhla algoritmus PAMS (Perceptual Analysis Measurement System). Tento algoritmus je používaný metódou PESQ.

Pôvodný a prenesený signál sú najprv časovo a amplitúdovo korelované. Táto korelácia je prevádzaná v každom bloku osobitne, nie na celom zázname naraz ako tomu bolo pri predošlých algoritmoch. Tým je možné zachytenie premenlivého oneskorenia. Oba signály sú ďalej spracovávané algoritmom, ktorého výsledkom je dvojrozmerné časovo-frekvenčné pole hodnôt, ktoré približne zodpovedá ľudskému vnímaniu daného signálu. Rozdiel sebe odpovedajúcich hodnôt týchto dvoch súborov tvorí tzv. posluchovú odchýlku, ktorá je použitá pre odhad hodnotiacich parametrov.



Obrázok 6: Bloková schéma PESQ

MOS skóre nadobúda pri subjektívnom testovaní hodnoty od 1 (najhoršia kvalita) po 5 (najlepšia kvalita), vid' tabuľku 4. Hodnoty MOS pre PESQ podľa ITU-T P.862 sú v rozsahu -0,5 až 4,5. Aby bolo možné porovnať PESQ so subjektívnym hodnotením, musíme zjednotiť ich rozsah, to zaisťuje odporúčenie ITU-T P.862.1. Toto doporučenie definuje funkciu pre prepočet vypočítaných PESQ parametrov na MOS skóre (3.22). Inverzný prevod popisuje rovnica (3.23).

$$y = 0.999 + \frac{4.999 - 0.999}{1 + e^{(-1.4945 \cdot x + 4.6607)}} \quad (3.22)$$

$$x = \frac{4.6607 - \ln\left(\frac{4.999 - y}{y - 0.999}\right)}{1.4945} \quad (3.23)$$

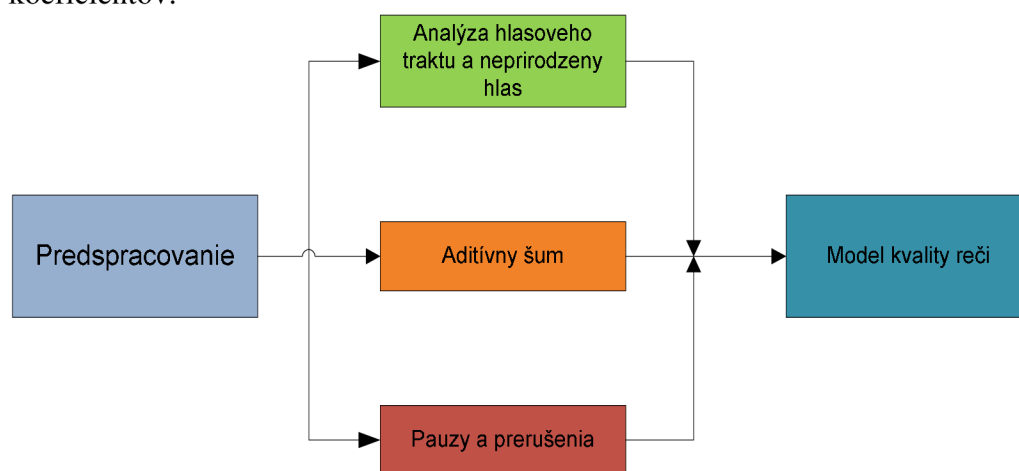
3.4.6 Neintrusívne metódy

Podstatou neintrusívnych metód je predikovanie kvality degradovaného signálu bez znalosti pôvodného (referenčného) signálu. Všetky doteraz popísané metódy a algoritmy porovnávajú vlastnosti signálu vypočítané z pôvodného a degradovaného signálu. Signál môže byť degradovaný vplyvom echa, stratou paketov prípadne iným rušením, ktoré sú spôsobené prenosom sieťou. Prípadne iným externým rušením. Neintrusívne metódy sú založené na pasívnom monitorovaní prebiehajúceho spojenia. Môžu byť implementované v rámci monitorovacích či dohľadových systémov.

❖ Metóda 3SQM

Podobne ako metóda PESQ sa aj táto metóda snaží odhadnúť MOS skóre, ktorým by bol signál ohodnotený pri subjektívnom teste. Na rozdiel od PESQ nepotrebuje 3SQM pre odhad kvality signálu originálny signál, výslednú

kvalitu odhaduje iba zo signálu degradovaného. Bloková schéma tejto metódy je na Obrázok 7. V bloku predspracovanie je hlas upravený pomocou IRS filtru, ďalej je vyhľadaná hraničná úroveň medzi užitočným signálom a šumom, signál je normovaný a nakoniec označený tzv. pitch markermi. Nasledujúce bloky (Analýza hlasového traktu, Aditívny šum, pauzy a prerušenia) vypočítajú 39 parametrov signálu. Podľa tzv. kľúčových parametrov sa určí, ktoré zo 6 skreslení (nízke kľudové SNR, pauzy a prerušenia, nízke segmentové SNR, neprirodený hlas – robotizácia, neprirodený mužský – ženský hlas) je dominantný. Pre výpočet výsledného MOS skóre je vždy použitých 11 rovnakých koeficientov zo spomínaných 39 a ďalej na základe určenia dominantného skreslenia je vybraná skupina 12-tich ďalších koeficientov. Niektoré sa môžu zhodovať s 11-timi stálymi koeficientmi. Celkovo je teda výsledné MOS skóre počítané lineárnou kombináciou maximálne 23 koeficientov.



Obrázok 7: Bloková schéma 3SQM

❖ **Doporučenie P.561 a P.562**

Podrobne popísané v [15]. Toto doporučenie je typickým príkladom neintrusívneho merania kvality. Štandard P.561(INMD) obsahuje zoznam parametrov, ktoré je treba vyhodnocovať na prenesenej verzii signálu (napr. úroveň signálu, SNR, rečová aktivita, echo) a P.562(CCI – Call Clarity Index) obsahuje návod ako hodnoty týchto parametrov skombinujeme do jediného výsledného parametru.

❖ **Doporučenie P.563(P.SEAM)**

P.563 je najnovším z „rodiny“ štandardov ITU-T v oblasti neintrusívneho merania kvality prenášaného rečového signálu. Zpracováva prenesenú vzorku pomocou 3 odlišných algoritmov a kombinuje ich výsledné hodnotenie do jedného parametru. Podrobný popis vnútornej štruktúry zatiaľ nie je zverejnený. Koeficient korelácie s objektívnymi posluchoвыми testami sa pohybuje u bežných typov skreslenia v rozmedzí 0.78-0.86.

❖ **PsyVoIP**

Algoritmus PsyVoIP je proprietárna metóda vyvinutá firmou Psytechnic Ltd. pre odhad kvality prenosu hlasu v IP prenosoch. Nepracuje s hlasovými vzorkami, ale iba s parametrami prenosu ako sú oneskorenie a jeho zmeny, typ kodeku či strata paketov. Všetky tieto parametre sú vyhodnocované neurónovou sieťou, ktorá nám odhadne výsledné kvality na stupnici MOS. Jedná sa o veľmi progresívny algoritmus, ktorý je schopný dosiahnuť vysokú koreláciu s výsledkami posluchovéch testov. Bohužiaľ firma Psytechnic Ltd. Am na tento algoritmus nepovolila copyright a doposiaľ bližší popis algoritmu nezverejnila.

4 Implementácia

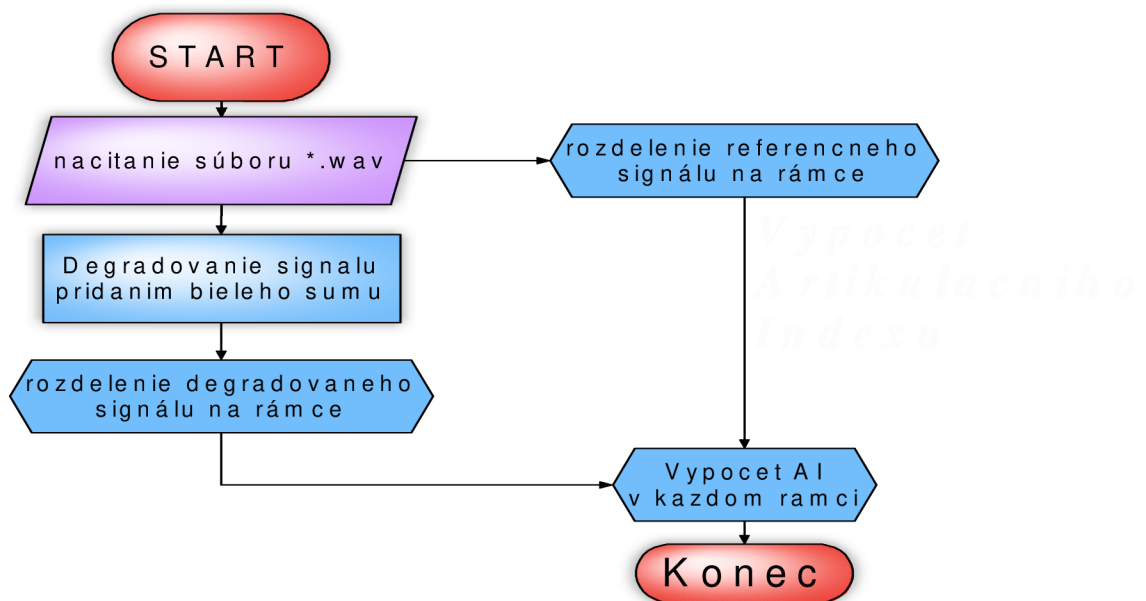
Vybrané metódy som prakticky realizoval v prostredí MATLAB (MATrix LABoratory). MATLAB je robustný nástroj, v ktorom môžeme simulovať prakticky všetko čo je možné popísať matematicky. Ponúka viac ako 1000 funkcií.

Hlavné klady MATLAB-u

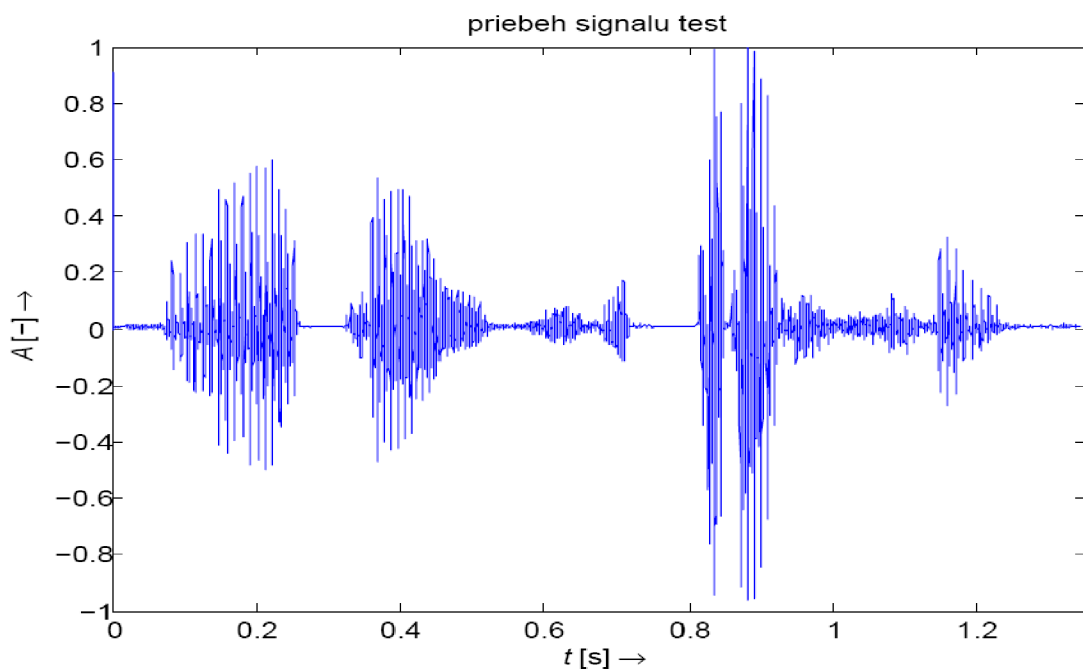
- + jazyk 4. generácie používajúci kľúčové slová
- + intuitívna ponuka, rýchly výber, okamžitá odozva
- + všetky vlastnosti ovplyvniteľné a nastaviteľné nielen v aplikáciách, ale aj v prostredí
- + bohatá nápoveda v HTML a pdf formáte
- + import a export dát v bežných formátoch
- + interaktívny nástroj na tvorbu užívateľských rozhraní
- + spolupráca s externými aplikáciami a jazykmi ako C, C++, Java, COM, .NET, Fortran

4.1 Výpočet Artikulačného Indexu

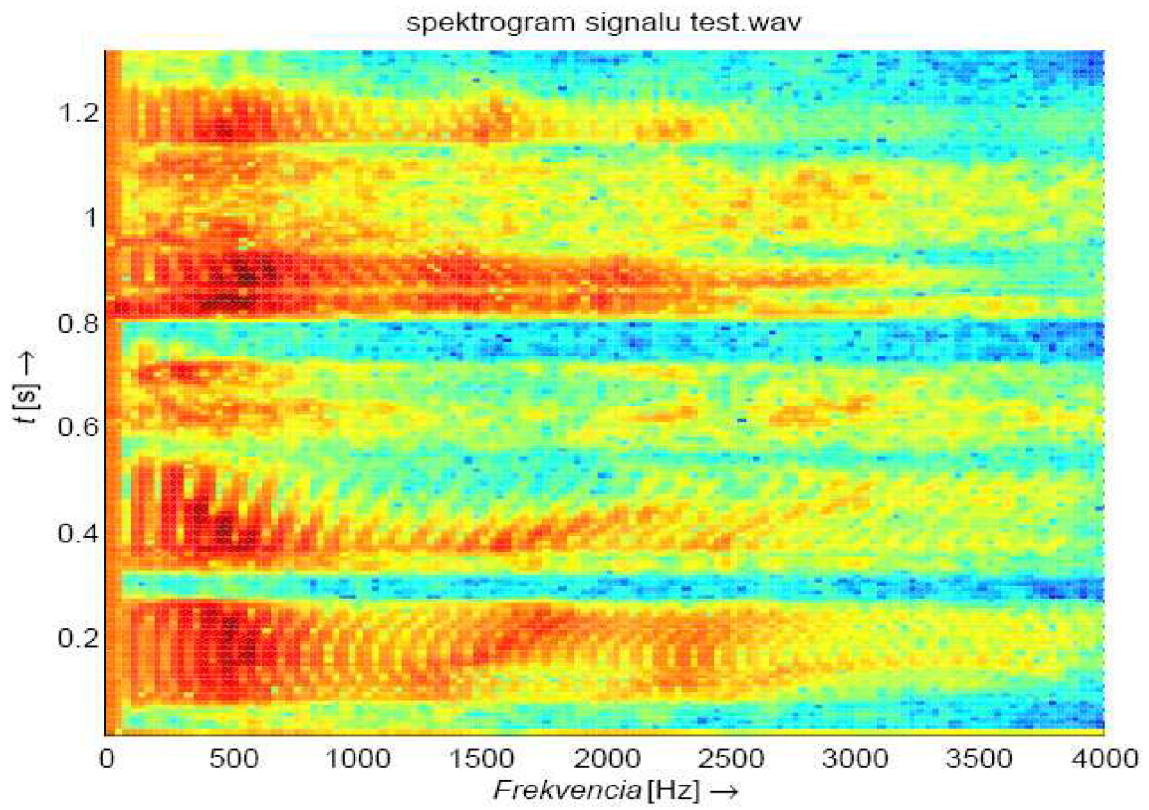
V prostredí MATLAB som vytvoril funkciu (zdrojový kód v prílohe) vývojový diagram (Obrázok 8), ktorým načítam súbor s príponou .wav. Načítaný signál je degradovaný bielym šumom. Degradovaný aj referenčný signál sa rozdelí na segmenty (rámce) pomocou funkcie segmentace (umožní nám nastaviť rôznu dĺžku okna a prekrytia). V každom segmente spočíta SNR (matematický popis v kapitole 3.4.1) Na základe vypočítaného segmental SNR (3.6) určíme artikulačný index. Získame maticu, ktorej každý prvok udáva hodnotu artikulačného indexu v danom segmente. Tieto hodnoty sa spriemerujú, čím dostávame jednu hodnotu Artikulačného indexu pre daný signál, ktorá vypovedá o jeho kvalite. Čím väčšia hodnota AI, tým kvalitnejší je skúmaný signál. Spriemerovanie hodnôt AI so sebou však vnáša aj značnú chybovosť do celkového výsledku. Najmä v oblastiach ticha, kde je minimálny šum, úplne pokrýva signál. Algoritmus automaticky takýto segment vyhodnotí ako nezrozumiteľný, čiže s veľmi nízkou hodnotou AI, tá potom do celkového výsledku vnáša relatívne veľkú chybu. Aj keď na užitočný signál nemá temer žiadny vplyv. Preto lepšou vypovedajúcou hodnotou je matica hodnôt AI pre každý segment jedna hodnota AI (príklad na obrázku 14). Pre lepšie porovnanie som vzorky podrobil aj subjektívnemu testovaniu. Každá degradovaná nahrávka bola pustená 3 krát skupine 10 poslucháčov. Medzi poslucháčmi boli zastúpené všetky vekové kategórie, 4 ženy a 6 mužov (jednalo sa o amatérskych poslucháčov, ich výsledky slúžia len pre túto prácu). Každý poslucháč ohodnotil nahrávku hodnotou so stupnice MOS (viď tabuľka 4). Čiže pre každú nahrávku som získal 10 hodnôt, tie som spriemeroval a získal výslednú hodnotu MOS pre danú nahrávku. Výsledky testov sú uvedené v tabuľkách 5 až 10. Obrázky na nasledujúcich stranách nám približujú priebeh testovaných signálov. Priebeh jednotlivých signálov ma nemalý vplyv na celkový výsledok.



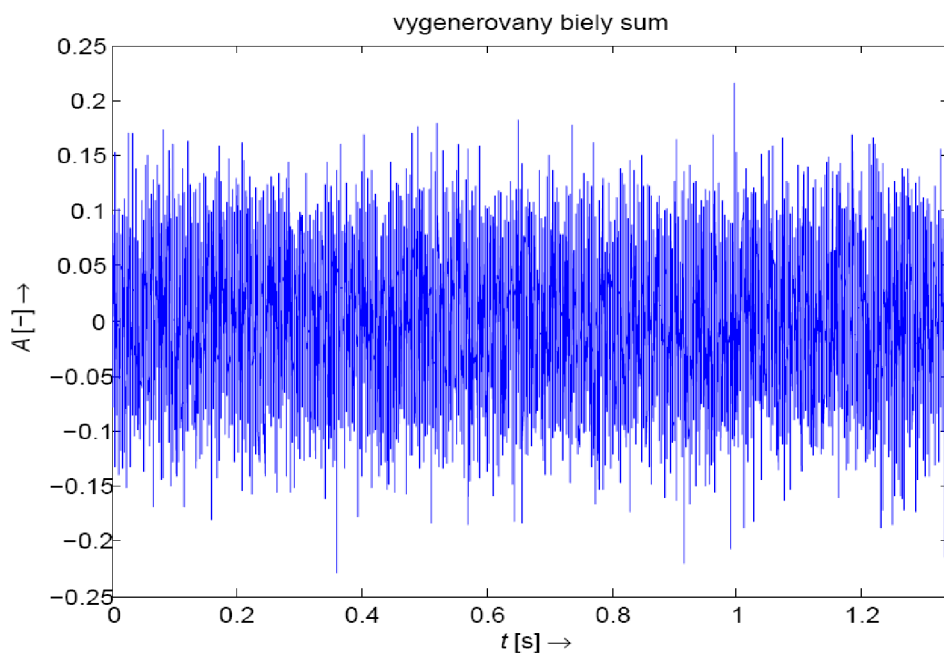
Obrázok 8: Vývojový diagram algoritmu výpočtu Artikulačného indexu



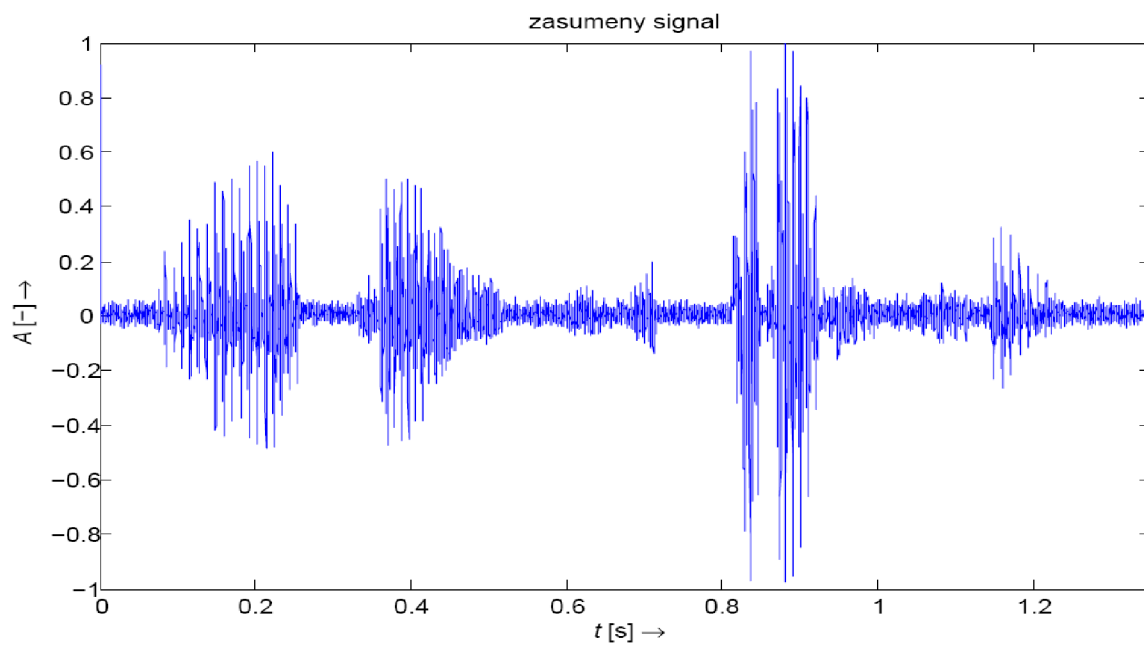
Obrázok 9: Priebeh signálu test.wav „létající prase“



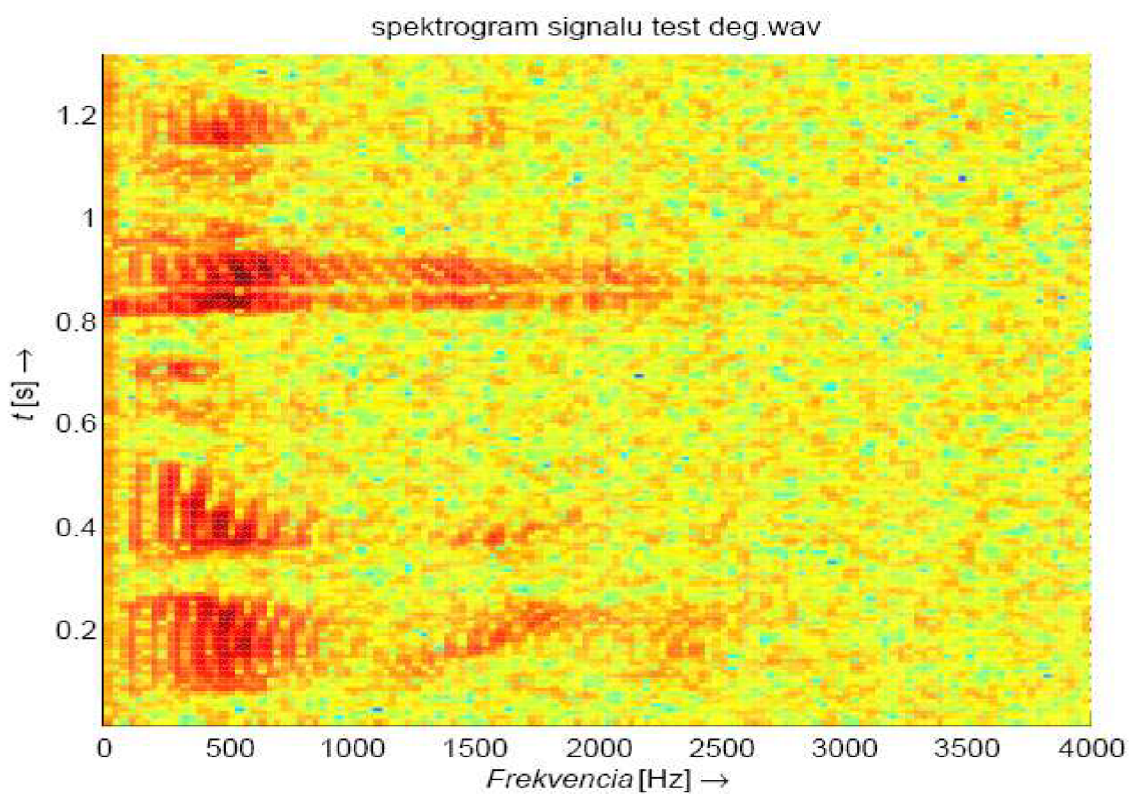
Obrázok 10: Spektrogram signálu test.wav



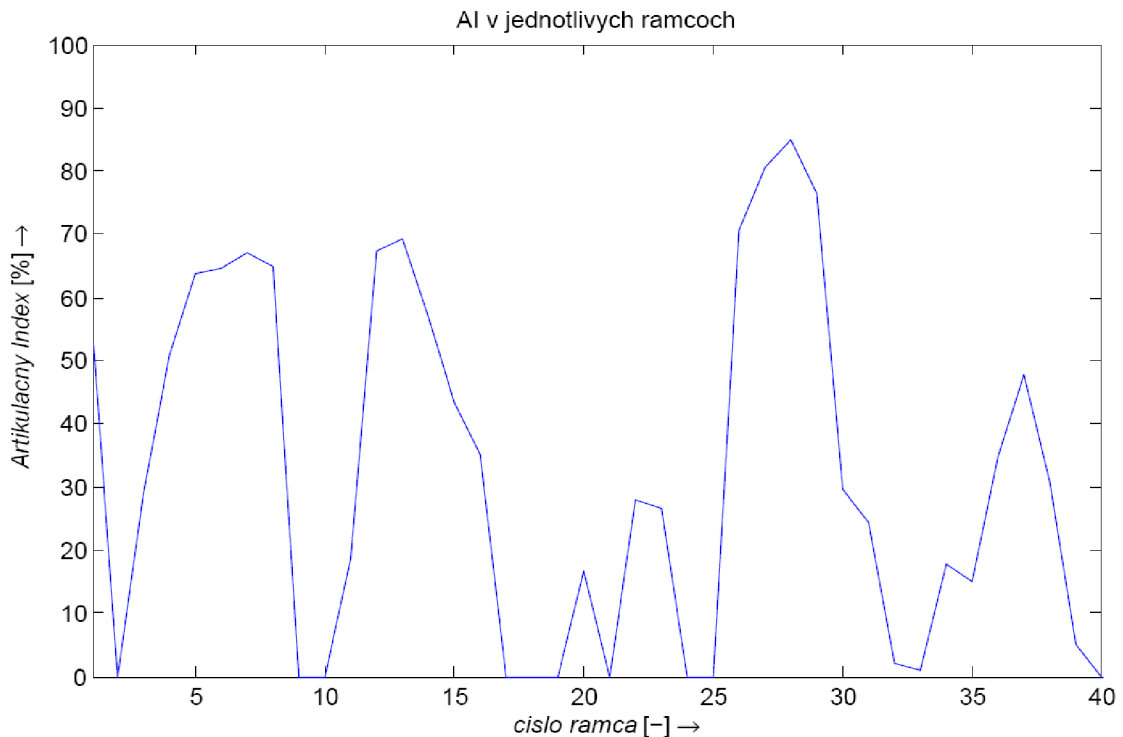
Obrázok 11: Biely šum



Obrázok 12: Zašumený signál test.wav



Obrázok 13: Spektrogram degradovaného signálu test_deg.wav



Obrázok 14: Hodnota AI v jednotlivých rámcoch signál test.wav

Veľká kolísavosť AI (Obrázok 14) v jednotlivých segmentoch testovaných rečových signálov je daná povahou referenčného signálu, v niektorých segmentoch je referenčný signál šumom prekrytý, čiže počujeme len šum, čo zodpovedá AI = 0%. A naopak v niektorých segmentoch je šum voči referenčnému signálu len nepatrný (vysoké kladné hodnoty AI až 80%). U harmonického signálu sa naopak hodnoty AI v jednotlivých segmentoch menia len minimálne.

Tabuľka 5: Výsledky testovania AI a.wav

Testovaný signál:	a.wav	fs=16kHz, wind=256, overlen=0, Nbits=16											
Odstup signálu od šumu [dB]	60	40	35	30	27	25	22	20	17	14	10	8	
Celkový AI[%]	99,56	86,40	72,73	56,73	47,87	41,22	32,35	26,72	18,54	11,19	2,94	0,42	
MOS [-]	5	5	5	4	4	4	3	3	2	2	1	1	
Čas výpočtu [s]	0,51	0,46	0,45	0,45	0,44	0,65	0,49	0,46	0,89	0,47	0,44	0,44	

Tabuľka 6: Výsledky testovania AI s.wav

Testovaný signál:	s.wav	fs=16kHz, wind=256, overlen=64, Nbits=16											
Odstup signálu od šumu [dB]	80	60	55	50	45	40	35	30	25	20	15	10	
Celkový AI[%]	65,31	29,98	20,36	11,83	4,33	0,27	0,00	0,00	0,00	0,00	0,00	0,00	
MOS [-]	4	4	4	3	3	2	2	2	1	1	1	1	
Čas výpočtu [s]	0,5	0,49	0,49	0,49	0,47	0,52	0,53	0,52	0,49	0,47	0,51	0,53	

Tabuľka 7: Výsledky testovania AI test.wav

Testovaný signál:	test.wav	fs=8kHz, wind=256. overlen=64, Nbits=16								
<i>Odstup signálu od šumu [dB]</i>	70	60	50	45	40	35	30	25	20	15
<i>Celkový AI[%]</i>	99,12	89,49	61,53	57,96	43,15	30,24	19,54	10,76	4,84	1,27
MOS [-]	5	4	4	4	3	3	2	2	1	1
Čas výpočtu [s]	0,44	0,44	0,45	0,48	0,47	0,49	0,46	0,43	0,44	0,46

Tabuľka 8: Výsledky testovania AI 7.wav

Testovaný signál:	7.wav	fs=8kHz, wind=256. overlen=64, Nbits=16								
<i>Odstup signálu od šumu [dB]</i>	80	75	70	60	55	50	45	42	40	37
<i>Celkový AI[%]</i>	85,18	75,24	61,53	33,14	19,94	10,38	3,43	1,12	0,43	0,00
MOS [-]	5	5	4	4	4	3	2	2	2	1
Čas výpočtu [s]	0,46	0,43	0,41	0,39	0,47	0,50	0,41	0,45	0,49	0,48

Tabuľka 9: Výsledky testovania AI cos.wav

Testovaný signál:	cos.wav*	fs=8kHz, wind=256. overlen=64, Nbits=16, *-harmonický signál								
<i>Odstup signálu od šumu [dB]</i>	50	40	35	30	25	20	15	10	8	3
<i>Celkový AI[%]</i>	100,00	100,00	100,00	89,83	73,25	56,60	39,94	23,51	16,70	0,79
MOS [-]	5	5	3	2	2	1	1	1	1	1
Čas výpočtu [s]	0,49	0,44	0,51	0,44	0,48	0,53	0,46	0,43	0,42	0,46

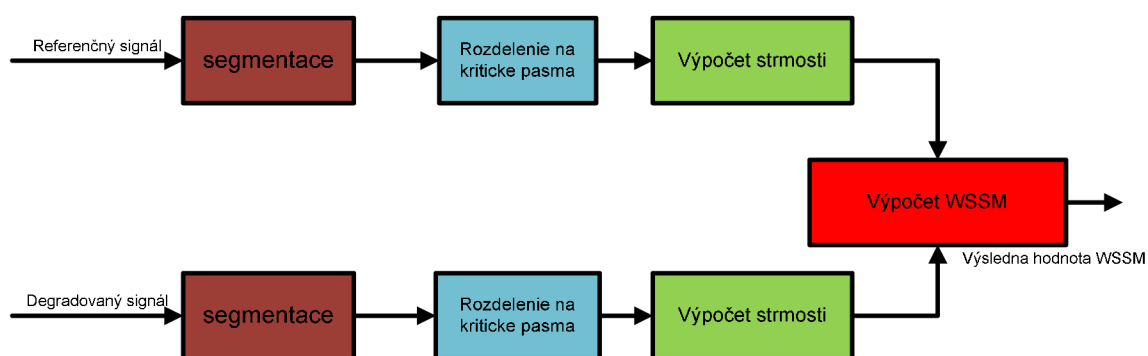
Tabuľka 10: Výsledky testovania AI housle.wav

Testovaný signál:	housle.wav	fs=8kHz, wind=256. overlen=64, Nbits=16								
<i>Odstup signálu od šumu [dB]</i>	60	50	45	40	35	30	27	25	20	15
<i>Celkový AI[%]</i>	95,38	82,87	70,7	55,25	40,06	25,85	18,36	13,91	5,05	0,73
MOS [-]	5	5	4	3	2	2	1	1	1	1
Čas výpočtu [s]	21,42	20,82	20,00	20,21	20,56	20,39	20,28	21,34	20,44	20,22

Zo získaných výsledkov môžeme usúdiť, že korelácia medzi subjektívnym hodnotením a výpočtom AI zo signálu, nie je vysoká. Príčin je viacero. Ako je už vyššie zmienené spriemerovať AI z jednotlivých segmentov, aby sme dostali jednu hodnotu, ktorá vypovedá o kvalite skúmaného signálu, nie je príliš šťastné. Vplyv chyby vnášanej do celkového AI spriemerovaním by sa dal zmenšiť vloženíím váhového koeficientu pred výpočet SNR v každom segmente. Váhový koeficient by nadobúdala hodnotu 1 ak by v danom segmente bola reč a 0 ak by v danom segmente bola pauza. Toto by bolo možné realizovať funkciou VAD (Voice Activity Detector). Tu vyvstáva asi najväčší problém s určením prahu čo je ešte užitočný signál. Tento problém je o to zložitejší pri signále, ktorý používame ako referenčný a má na pozadí akýkoľvek minimálny šum. V takomto prípade sa detekuje aktivita reči veľmi obtiažne. Hlavnou príčinou veľkej korelácie je, že pri výpočte AI nie sú zohľadnené žiadne psychoakustické vlastnosti ľudského sluchu. Charakter testovaného signálu tiež ovplyvňuje vzájomnú koreláciu. Skôr však vplyva na objektívne posudzovanie kvality. Napríklad nízke hodnoty MOS pri relatívne ešte vysokých hodnotách AI pri nahrávke s.wav (obrázok 15 v prílohách) prisudzujem práve tomu, že spoluhláska patrí medzi trené spoluhlásky tzv. frikatíva. Trené spoluhlásky vznikajú za pomoci úžiny (konstrikce), ktorá sa stavia do prúdu vzduchu, čím vzniká šum. To spôsobuje ich šumový charakter. Preto samostatné s sa nám javí ako šum. Nízke hodnoty MOS pri testovaní harmonického signálu pripisujem takisto vlastnostiam ľudského sluchu. Človek pri počúvaní automaticky hľadá v počutej nahrávke určitý zmysel. Keďže harmonický signál v nás nevyvoláva žiadny vnem, poslucháčom sa veľmi ťažko hodnotilo jeho znehodnotenie.

4.2 Metóda Wighted Spectral Slope

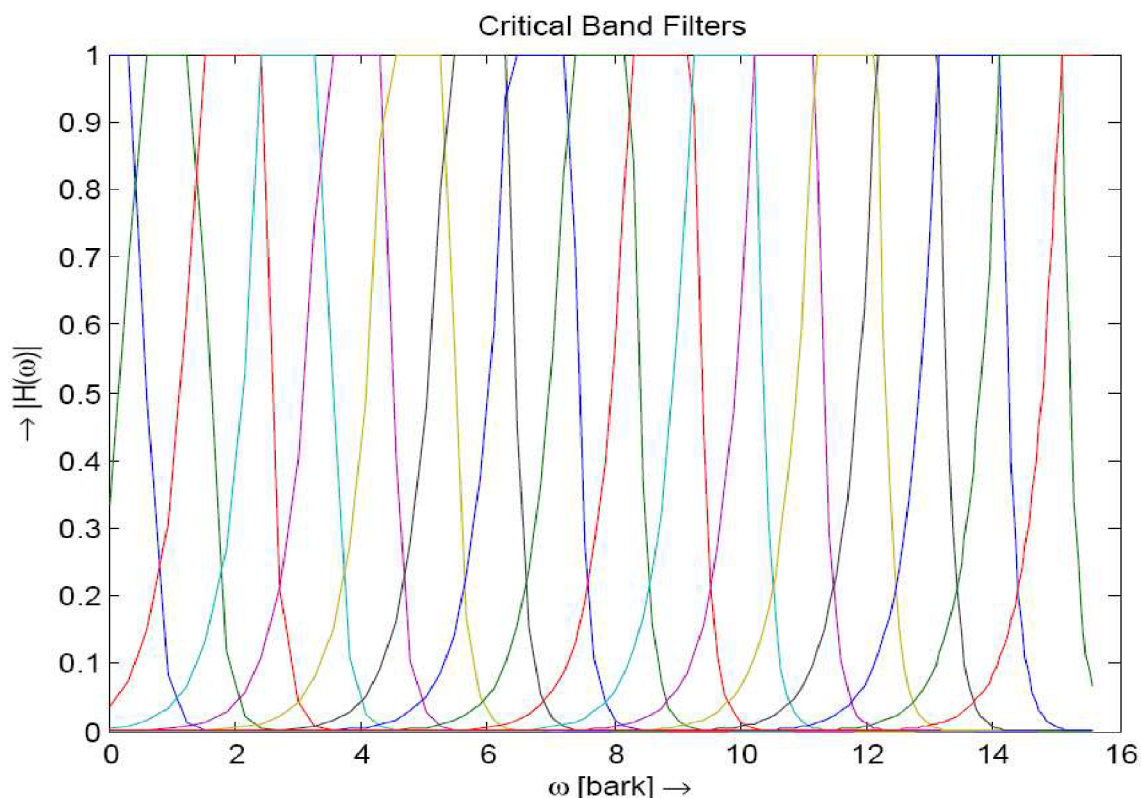
V prostredí MATLAB som vytvoril funkciu pre výpočet hodnoty WSSM podrobný matematický a teoretický popis je v kapitole (3.4.3 a) bloková schéma je na Obrázok 15. Čím nižšia je výsledná hodnota WSSM, tým je skúmaný signál podobnejší referenčnému, čiže menej degradovaný. Nízka hodnota WSSM vypovedá o vysokej kvalite skúmaného signálu. Naopak vysoká hodnota WSSM nám napovedá, že signál bude silne deformovaný. Metóda WSS rozdelí signál na segmenty (podobne ako pri výpočte AI funkciou segmentace, opäť je možné voliť dĺžku okna v nasledujúcich krokoch však bude prevádzaná FFT preto by segmenty mali mať dĺžku 2^N (kde N je celé číslo, vyplýva z vlastností FFT). V ďalšom kroku sa každý segment rozdelí na kritické barkové pásma pomocou funkcie `fft2barkmx` [16]. Funkcia používa tieto vstupné parametre: Počet vzorkov FFT – `nfft`, `fs`- vzorkovania frekvencia skúmaného signálu. Počet filtrov použitých v banke- `nfilters`. Počet pásmových priepustí v banke je závislý na vzorkovacej frekvencii a spracovávanom pásme (Tabuľka 11). Príklad rozloženia banky barkových filtrov typu pásmová priepusť pre šírku pásma spracovávaného signálu od 0 do 15,75 barkov (0 – 4 kHz, `fs` = 8kHz) je na Obrázok 16. Minimálnu a maximálnu frekvenciu spracovávaného pásma doporučuje Tabuľka 11.



Obrázok 15: Bloková schéma metódy WSS

Tabuľka 11: Doporučené hodnoty počtu filtrov

vzorkovacia frekvencia <i>fvz</i> [kHz]	spracované pásmo [kHz]	spracované pásmo [bark]	počet filtrov
8	0 - 4	0 - 15,57	17
11	0 - 5,5	0 - 17,47	19
16	0 - 8	0 - 19,71	21
22	0 - 11	0 - 21,26	23
44	0 - 22	0 - 25,77	27



Obrázok 16: Rozloženie banky barkových filtrov

V každom frekvenčnom pásme je vypočítaná strmosť (3.14). Vypočítané strmosti sú váhované podľa toho či sa nachádzajú v blízkosti lokálneho maxima, alebo minima spektra. Výpočet váhovacej konštanty je daný rovnicou (3.15). Výsledná hodnota WSSM sa potom spočíta podľa rovnice (3.18).

Funkčnosť algoritmu som otestoval na zhodných testovacích nahrávkach ako som testoval algoritmus pre výpočet AI. Výsledky testovania sú zhrnuté v tabuľkách 12 až 17.

Tabuľka 12: Výsledky testovania WSSM a.wav

Testovaný signál:	a.wav	fs=16kHz, wind=256, overlen=0, Nbits=16										
Odstup signálu od šumu [dB]	60	40	35	30	27	25	22	20	17	14	10	8
dWSSM[-]	0,00	0,11	0,38	1,46	3,68	5,76	10,14	17,33	38,63	79,36	275	493
MOS [-]	5	5	5	4	4	4	3	3	2	2	1	1
Čas výpočtu [s]	0,92	0,91	0,90	0,92	0,91	0,92	0,49	0,46	0,89	0,47	0,44	0,44

Tabuľka 13: Výsledky testovania WSSM s.wav

Testovaný signál:	s.wav	fs=16kHz, wind=256, overlen=0, Nbits=16										
Odstup signálu od šumu [dB]	80	60	55	50	45	40	35	30	25	20	15	10
dWSSM[-]	0,00	0,00	0,01	0,02	0,07	0,21	0,65	2,23	6,84	24,45	83,57	276,91
MOS [-]	4	4	4	3	3	2	2	2	1	1	1	1
Čas výpočtu [s]	1,34	1,35	1,32	1,32	1,33	1,3	1,34	1,35	1,31	1,32	1,31	1,33

Tabuľka 14: Výsledky testovania WSSM test.wav

Testovaný signál:	test.wav	fs=8kHz, wind=256. overlen=0, Nbits=16										
Odstup signálu od šumu [dB]	70	60	50	45	40	35	30	25	20	15	10	8
dWSSM[-]	0,00	0,00	0,02	0,05	0,15	0,60	1,71	6,37	20,62	88,47	277,28	482,55
MOS [-]	5	4	4	4	3	3	2	2	1	1	1	1
Čas výpočtu [s]	1,10	1,09	1,06	1,08	1,02	1,34	1,37	1,34	1,36	1,36	1,34	1,36

Tabuľka 15: Výsledky testovania WSSM 7.wav

Testovaný signál:	7.wav	fs=8kHz, wind=256. overlen=0, Nbits=16										
Odstup signálu od šumu [dB]	80	75	70	60	55	50	45	42	40	37	30	25
dWSSM[-]	0,00	0,00	0,00	0,00	0,01	0,02	0,02	0,14	0,21	0,46	2,82	8,25
MOS [-]	5	5	4	4	4	3	2	2	2	1	1	1
Čas výpočtu [s]	1,15	0,60	0,6	0,74	0,73	0,74	0,74	0,72	0,74	0,73	0,72	0,59

Tabuľka 16: Výsledky testovania WSSM cosinus.wav

Testovaný signál:	cosinus.wav*	fs=8kHz, wind=256. overlen=64, Nbits=16, *-harmonický signál									
Odstup signálu od šumu [dB]	50	40	35	30	25	20	15	10	8	3	
dWSSM[-]	0,02	0,23	0,81	2,47	8,12	26,75	91,94	307,11	463,54	1675,10	
MOS [-]	5	5	3	2	2	1	1	1	1	1	
Čas výpočtu [s]	0,49	0,61	0,64	0,78	0,77	0,77	0,77	0,79	0,76	0,77	

Tabuľka 17: Výsledky testovania WSSM housle.wav

Testovaný signál:	housle.wav	fs=8kHz, wind=256. overlen=0, Nbits=16									
Odstup signálu od šumu [dB]	60	50	45	40	35	30	27	25	20	15	
dWSSM[-]	0,00	0,02	0,06	0,21	0,74	2,61	5,56	9,51	34,01	114,03	
MOS [-]	5	5	4	3	2	2	1	1	1	1	
Čas výpočtu [s]	27,75	34,21	33,51	33,61	33,57	33,71	33,59	33,55	33,65	33,60	

Celková hodnota WSSM predikuje subjektívnu kvalitu hodnoteného signálu presnejšie ako výpočet AI. Hlavnou príčinou je vyššia náročnosť metódy WSS, ktorá vo svojom tele zohľadňuje aj psychoakustické vlastnosti sluchu. Ani táto metóda však nie je úplne dokonalá. Ako jej najväčšiu slabinu vidím v nemožnosti jednoznačného prevodu hodnoty WSSM na subjektívnu škálu MOS. Z výsledkov, ktoré som získal, môžeme jednoznačne

povedať len, že veľmi malé hodnoty WSSM výborne predikujú vysokú kvalitu skúmaného signálu. Naopak veľmi vysoké hodnoty WSSM nám napovedajú, že signál bude silne degradovaný, pri hodnotách blížiacich sa ku 100 je veľký problém rozumieť užitočnému signálu, pretože pri týchto hodnotách je značne maskovaný šumom.

4.3 Výpočtová náročnosť

Pri vytvorených funkciách pre zistenie objektívnej kvality skúmaného signálu som sa zamerlal okrem schopnosti predikcie subjektívnej kvality (výsledky v kapitolách 4.1 a 4.2), na výpočtovú náročnosť jednotlivých algoritmov. Ako hodnotiacu veličinu som zvolil čas, za ktorý prebehne celý algoritmus. Ten som v MATLABe sledoval pomocou príkazov Tic (spúšťa stopky po spustení algoritmu) a Toc (ukončuje meranie času na konci algoritmu). Výsledky sú uvedené v **Tabuľka 18**. Aby boli výsledné hodnoty relevantné museli byť zabezpečené pre všetky testovania identické podmienky. Všetky funkcie boli vytvorené v prostredí MATLAB verzia 7.6.0.324(R2008a) z 10. februára 2008, ktorý bol spúšťaný z operačného systému Microsoft Windows XP Professional, Service Pack 3. Na procesore AMD Sempron 2500+, 1,41GHz a s operačnou pamäťou 1,25 GB. Na pozadí neboli spustené žiadne iné programy.

Tabuľka 18: Porovnanie výpočtovej náročnosti

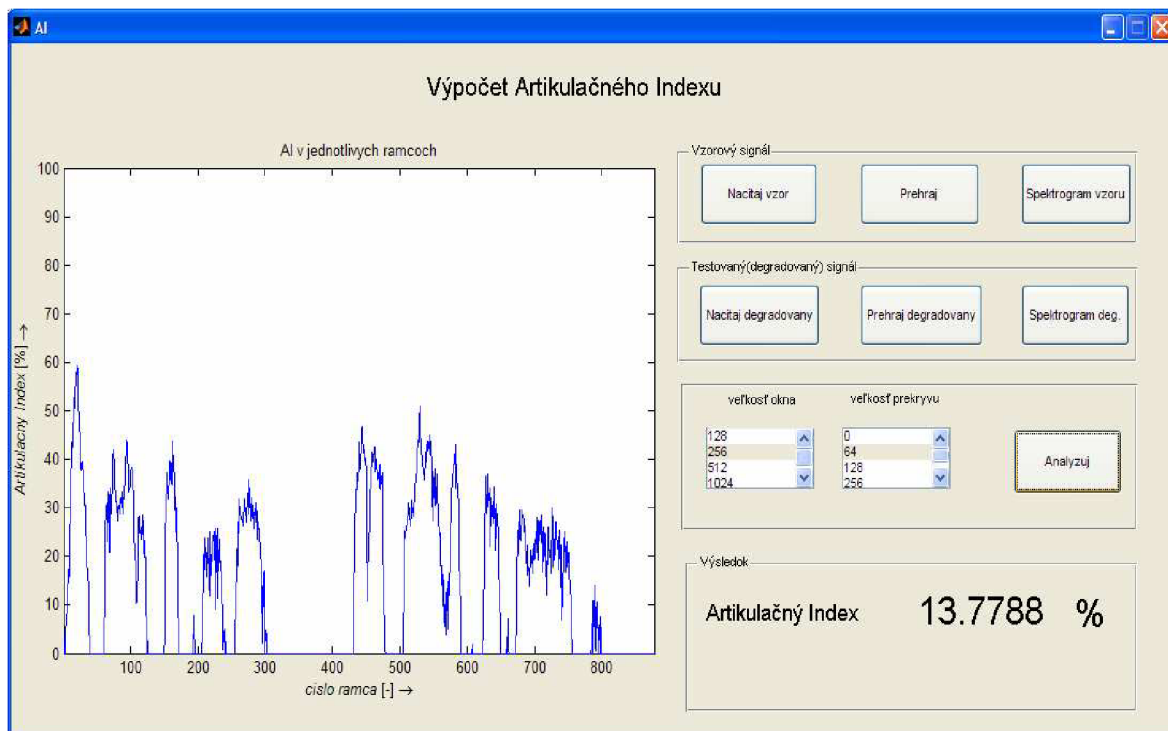
testovaný signál	priemerný čas výpočtu [s]	
	AI	WSS
a.wav	0,51	0,90
s.wav	0,50	1,33
7.wav	0,45	1,24
test.wav	0,46	0,74
cosinus.wav	0,47	0,72
housle.wav	20,57	33,08

Ako môžeme vidieť v tabuľke 18, výpočtovo je náročnejší algoritmus WSS, pri jeho použití je dĺžka výpočtu približne dvojnásobná oproti AI. Tým sme si len potvrdili teoretické znalosti popísané v kapitole 3.4.1 a 3.4.3, ktoré nám naznačujú, že výpočet AI je výpočtovo veľmi jednoduchý. Algoritmus porovnáva energie referenčného a degradovaného signálu. Algoritmus WSS vo svojom vnútri prevádza delenie signálu do kritických (barkových pásiem), výpočet strmosti v jednotlivých pásmach, výpočet váhovej konštanty a na záver samotný výpočet vzdialenosti $dWSSM$. Algoritmus WSS je síce výpočtovo náročnejší, ale je lepším prediktorom subjektívnej kvality, najmä kôli banke filtrov, ktorá delí signál do barkových (kritických) pásiem počutia.

4.4 Grafické rozhranie

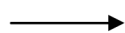
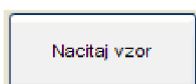
V prostredí GUIDE som vytvoril pre každú metódu užívateľské grafické rozhranie. Grafické rozhranie výrazne zjednodušuje užívateľovi manipuláciu s aplikáciou, ktorá je oveľa „priateľskejšia“ a intuitívnejšia ako ovládanie funkcií priamo z príkazového riadku

MATLABu, umožňuje použiť aplikáciu aj záujemcom, ktorí nie sú rodenými programátormi. Vzhľad aplikácie je na obrázkoch 17 a 18.

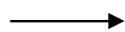
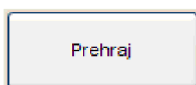


Obrázok 17: Aplikácia pre výpočet Artikulačného Indexu

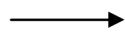
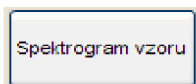
Na obrázku 17 je vzhľad aplikácie pre výpočet Artikulačného indexu. Aplikácia je rozdelená na 5 sekcií. V sekcii **vzorový signál** sa nachádzajú 3 tlačidlá. Kliknúť znamená:



Otvoriť dialógové okno pre **načítanie súboru s príponou *.wav** a jeho priebeh sa zobrazí v ľavom okne

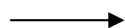
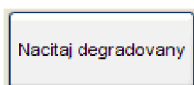


Prehrať načítaný signál

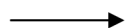
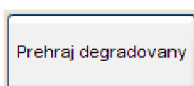


Zobraziť spektrogram načítaného signálu v okne

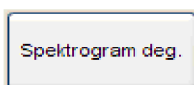
V sekcii **testovaný (degradovaný) signál** sú rovnako 3 tlačidlá. Kliknúť znamená:



Otvoriť dialógové okno pre **načítanie degradovaného vzorového súboru s príponou *.wav**

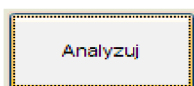
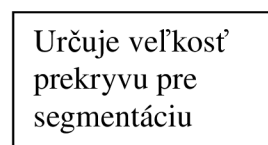
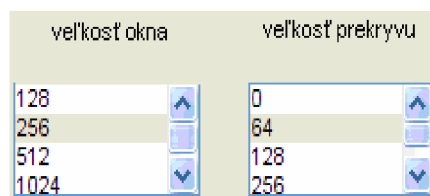
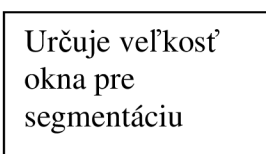


Prehrať degradovanú (testovanú) nahrávku, po prehraní oboch si môžeme urobiť vlastnú subjektívnu mienku o kvalite daného signálu



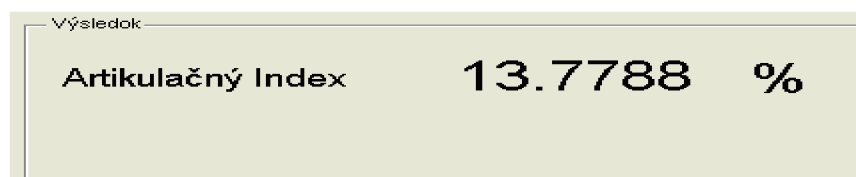
Zobraziť spektrogram testovaného signálu

V sekcii **Analyzuj** sa nachádzajú dva listboxy a tlačidlo Analyzuj:

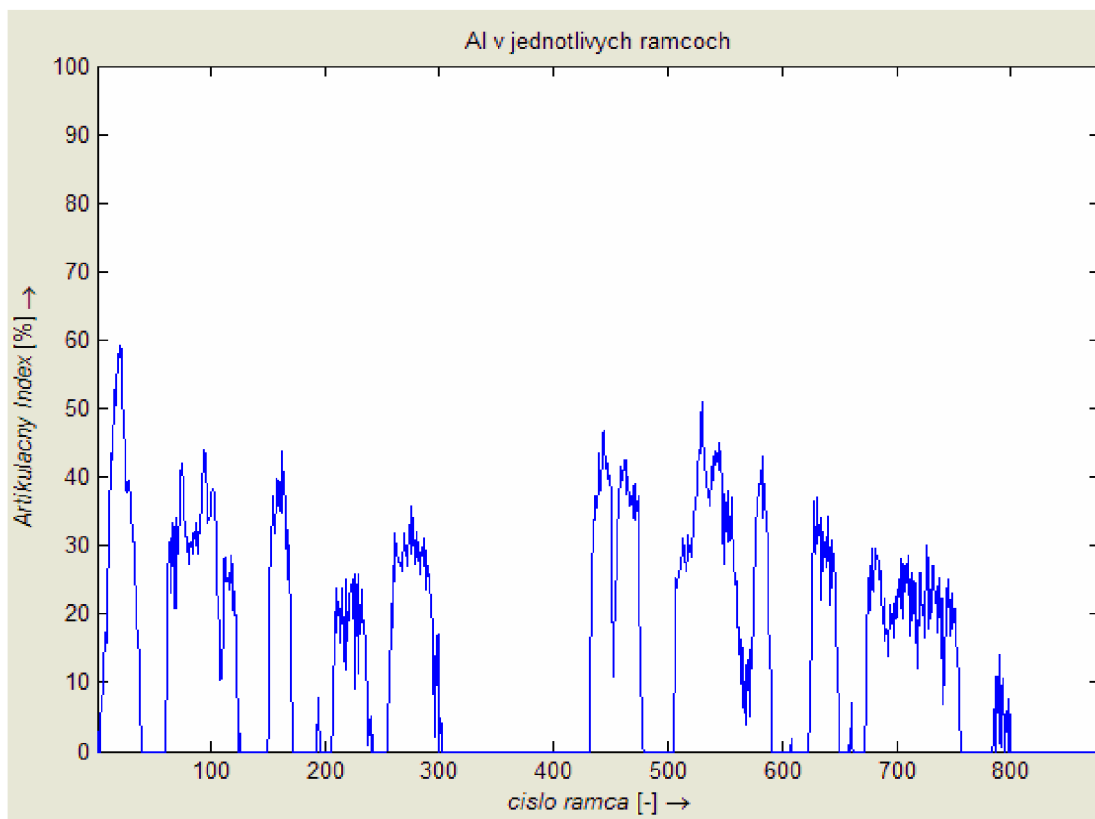


vyvolanie funkcie A_Index, ktorá vypočíta Artikulačný index v jednotlivých segmentoch a zobrazí sa v okne pre graf

Celkový Artikulačný Index vypočítaný ako priemer cez všetky segmenty sa zobrazí v sekcii **výsledok**.



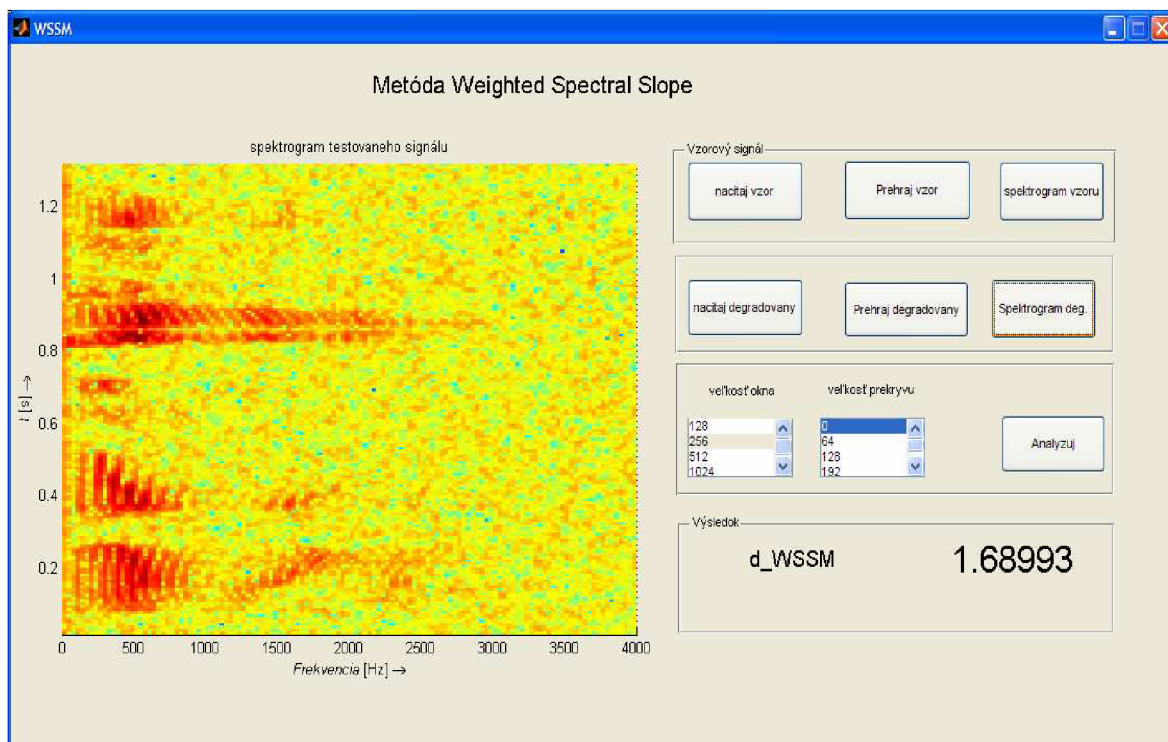
V ľavej sekcii, **okne pre vykreslenie grafu** sa dynamicky menia vykresľované objekty. Po kliknutí na tlačidlo **načítaj vzor** sa zobrazí priebeh načítaného signálu. Po kliknutí na **načítaj testovaný** sa zobrazí priebeh testovaného signálu. Obdobne po kliknutí na **spektrogram** sa zobrazí spektrogram zvoleného signálu. Po kliknutí na **analyzuj** sa zobrazí graf veľkosti AI v jednotlivých segmentoch.



okno pre vykresľovanie grafu

Aplikáciu Výpočet Artikulačného indexu je v praxi možné využiť napríklad pre zistenie kvality nahrávky (hudba, reč) po kompresii. Ako vzor by bol signál (nahrávka, reč) pred kompresiou. A ako testovaný signál by sme použili komprimovanú nahrávku. Výsledný Artikulačný Index by nám predikoval kvalitu komprimovanej nahrávky. **Čím väčšia hodnota AI, tým menšia je strata informácie v nahrávke.**

Obrázok 18 na nasledujúcej strane zobrazuje aplikáciu **pre výpočet d_WSSM**, s implementovanou metódou **weighted spectral slope**. Dizajn aj funkcia všetkých tlačidiel je veľmi obdobný ako v aplikácii pre výpočet Artikulačného indexu. Tlačidlo **analyzuj** volá funkciu **WSSM_func**, ktorej výstupom je výsledná hodnota d_WSSM zobrazená v sekcii výsledok. **Hodnota d_WSSM nám predikuje subjektívnu kvalitu, čím menšia je hodnota d_WSSM, tým menej je skúmaný signál poškodený voči referenčnému.**



Obrázok 18: Aplikácia s implementovanou metódou WSS

Aplikáciu môžeme použiť pre zistenie kvality nahrávky, ktorá prešla určitým stratovým kódovaním. Alebo pre zistenie kvality nahrávky, ktorá bola prenášaná sieťou. V oboch prípadoch je ale nutné mať pôvodnú (referenčnú) nahrávku. Metóda WSSM vlastne porovnáva vzdialenosť medzi váhovanými strmostami jednotlivých barkových pásiem referenčného a testovaného signálu.

5 Záver

V tejto práci som sa zaoberal akustickým signálom, konkrétne hodnotením jeho kvality. Cieľom práce bolo oboznámenie sa a pochopenie vybraných objektívnych metód hodnotenia kvality audiosignálu a ich praktická aplikácia v prostredí programu MATLAB.

Kapitola „Kvalita audiosignálov“ je venovaná podrobnému popisu objektívnych metód hodnotenia kvality audiosignálov od jednoduchších až po najsofistikovanejšie a najzložitejšie. Hlavnou úlohou objektívnych metód je kvalitná predikcia subjektívnej kvality hodnoteného signálu.

V praktickej časti (kapitola 4) som v grafickom prostredí programu MATLAB implementoval dve vybrané metódy a podrobil ich rozsiahlemu testovaniu, výsledky som analyzoval v kapitole 4.1 a 4.2. Prvá aplikácia počíta hodnotu Artikulačného Indexu na základe rozdielov energií pôvodného a degradovaného signálu. Medzi hlavné prednosti patrí relatívna jednoduchosť, nízka výpočtová náročnosť. Naopak medzi najväčšie slabiny patrí nejednoznačnosť výsledku. Celková hodnota AI je počítaná ako priemer Artikulačných indexov v jednotlivých segmentoch. Čo môže vnášať do výslednej hodnoty AI skreslenie. Lepšiu vypovedajúcu hodnotu má graf závislosti hodnoty AI v jednotlivých segmentoch. Ďalší a podstatnejším nedostatkom je, že vo výpočte AI nie je zahrnutá žiadna vlastnosť psychoakustického sluchového vnemu. Pre otestovanie kvality daného signálu touto metódou musíme mať k dispozícii aj referenčný (čiže čistý) signál, čo môžeme považovať tiež za nevýhodu.

Do druhej aplikácie som implementoval metódu WSS. Výstupom aplikácie je hodnota rozdielu váhovaných strmosti kritických pásiem testovaného a referenčného signálu. Čím menšia je táto hodnota, tým je testovaný menej porušený. Výsledky získané touto metódou dosahujú lepšiu koreláciu oproti metóde založenej na výpočte AI. Prisudzujem to najmä skutočnosti, že metóda WSS obsahuje banku filtrov, ktorá každý segment rozdelí na barkove kritické pásma, čo je podstatou psychoakustického vnímania. Za slabinu tejto metódy môžeme rovnako ako pri predchádzajúcej metóde považovať nutnosť znalosti referenčného signálu. Z hľadiska výpočtovej náročnosti je táto metóda zhruba dvakrát náročnejšia ako výpočet AI. Podrobnejšiemu popisu výpočtovej náročnosti sa venuje kapitola 4.3. Obe aplikácie boli vytvorené v grafickom rozhraní programu MATLAB – GUIDE. Podrobný popis aplikácií, funkcie jednotlivých tlačidiel sú zhrnuté v kapitole 4.4.

Výsledky mojej práce potvrdili teoretické poznatky o vybraných metódach. Ani jedna z nich však nepatrí medzi úplne najmodernejšie štandardy hodnotenia kvality audiosignálov, ktoré sa používajú pre určovanie QoS v IP Telefónii (PESQ, PSQM). Alebo sú zabudované priamo v smerovačoch a podľa parametrov danej siete sú schopné predikovať kvalitu prenášaného signálu. Podrobné popisy algoritmov týchto štandardov nie sú však úplne prístupné.

Použitá literatura

- [1] MELKA, Alois. Základy experimentální psychoakustiky. 1. vyd. Praha : Akademie múzických umění v Praze, 2005. 328 s. ISBN 80-7331-043-0.
- [2] SYROVÝ, Václav. Hudební akustika. Praha : Základy experimentální psychoakustiky, 2003. 427 s. ISBN 80-7331-901-2.
- [3] VÍCH, R., SMÉKAL, Z.: Číslicové filtry. Academia, Praha 2000. ISBN 80-200-0761-X
- [4] MAITHANI, Sunita, RAWAT, J. S. Noise as a Tool for Spoken Language Identification. 2008 [cit. 2008-12-08]. Dostupný z WWW: <<http://www.aclweb.org/anthology-new/I/I08/I08-2114.pdf>>
- [5] VONDRÁŠEK, Martin, POLLÁK, Petr. Methods for Speech SNR estimation. *RADIOENGINEERING* [online]. 2005, vol. 14, no. 1 [cit. 2008-12-08]. Dostupný z WWW: <http://noel.feld.cvut.cz/speechlab/publications/037_radioengineering05.pdf>
- [6] KUBĚNA, Rostislav. Analýza signálů pomocí autoregresního modelování. *Sborník XXI* [online]. 1998 [cit. 2008-11-30]. Dostupný z WWW: <<http://www.fs.vsb.cz/akce/1998/asr98/Sbornik/kubena/kubena.htm>>
- [7] HAMZA, Özer, AVCIBAS, İsmail, SANKUR, Bülent. Steganalysis of Audio Based on Audio Quality Metrics. *TÜBİTAK* [online]. 2004 [cit. 2008-12-07]. Dostupný z WWW: <www.busim.ee.boun.edu.tr/~sankur/SankurFolder/Audio_Steganalysis_16.doc>
- [8] GRUNDLEHNER, Bernard, LECOCQ, Johan, RADU, Balan. PERFORMANCE ASSESSMENT METHOD FOR SPEECH ENHANCEMENT SYSTEMS. *PAPERS* [online]. 2005 [cit. 2008-12-08]. Dostupný z WWW: <http://www.math.umd.edu/~rvbalan/PAPERS/MyPapers/sps2005_speech_final.pdf>
- [9] *Zvukové vlnenie: MF WEB* [online]. 1998 [cit. 2008-12-08]. Dostupný z WWW: <<http://mfweb.wz.cz/fyzika/153.htm>>
- [10] SKARNITZL, Radek. Psychoakustika. *Akustika* [online]. 2008 [cit. 2008-12-08]. Dostupný z WWW: <http://fu.ff.cuni.cz/vyuka/akustika/3_psychoakustika.pdf>
- [11] ANSI S3.5-1997: ANSI S3-5-1997 Methods for calculation of the speech intelligibility index. 1997 [cit. 2008-12-08]. Dostupný z WWW: <<http://hse.bme.ym.edu.tw/chinese/modules/filemanager/files/projects/SSB-Project20040500101%2007MeetingMinutes/20040607-0612/ansi%20s3-5-1997.doc>>
- [12] *Akustika, vznik a šíření zvuku, frekvenční analýza a syntéza, sluchový vjem zvukového signálu* [cit. 2008-12-08]. Dostupný z WWW: <http://homen.vsb.cz/~ber30/texty/varhany/anatomie/pistaly_akustika.htm>
- [13] ITU-T P.862 : Perceptual evaluation of speech quality (PESQ): 2001, no. 2 [cit. 2009-05-12]. Dostupný z WWW: <http://www.itu.int/rec/T-REC-P.862-200102-I/en>

[14] DOC. ING. HOLUB Jan, PH.D.. *IT Point : TEORIE A PRAXE IP TELEFONIE* [online]. 9 Praha, 2009 [cit. 2009-05-12]. Český. Dostupný z WWW: <<http://www.itpoint.cz/voip/mereni-a-hodnoceni-QoS-v-IP-telefonii.asp?voip=9>>.

[15] ITU-T P.862 : In-service non-intrusive measurement. 2002, no. 7 [cit. 2009-05-13]. Dostupný z WWW: <<http://www.itu.int/rec/T-REC-P.561-200207-I/en>>.

[16] ELLIS, Dan. *PLP and RASTA (and MFCC, and inversion) in Matlab : using melfcc.m and invmelfcc.m* [online]. 07/2006. 2006 , 03/07/2006 [cit. 1985-05-21]. English. Dostupný z WWW: <<http://labrosa.ee.columbia.edu/matlab/rastamat/>>.

Zoznam skratiek

MOS	Mean Opinion Score	Priemerná známka hodnotenia
SNR	Signal to Noise Ratio	Odstup signál šum
SNR _{seg}	Segmental Sound to Noise Ratio	Segmentálny odstup signál šum
SNR _{fw-seg}	Frequency Weighted Segmental Sound to Noise Ratio	
AR		Autoregresívne
LPC	Linear Predictive Coding	
LLR	Log-likelihood Ratio	logaritmická podobnosť
LAR	Log-Area-Ratio	
WSS	Weighted Spectral Slope	
WSSM	Weighted Spectral Slope measure	
BSD	Bark Spectral Disortion	
MBDS	Modified Bark Spectral Disortion	
AI	Articulation Index	Artikulačný index
FFT	Fast Fourier Transform	rýchla fourierova transformácia
F _s	frequency sample	vzorkovacia frekvencia
PSQM	Perceptual speech quality measure	
QoS	Quality of Services	
CCI	Call Clarity Index	

A Prvá Príloha

V prílohe sa nachádza zdrojový kód spúšťacích m-filov pre grafické aplikácie

AI.m

```
function varargout = AI(varargin)
% AI M-file for AI.fig
%   AI, by itself, creates a new AI or raises the existing
%   singleton*.
%
%   H = AI returns the handle to a new AI or the handle to
%   the existing singleton*.
%
%   AI('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in AI.M with the given input arguments.
%
%   AI('Property','Value',...) creates a new AI or raises the
%   existing singleton*. Starting from the left, property value pairs
are
%   applied to the GUI before AI_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property
application
%   stop. All inputs are passed to AI_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only
one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help AI

% Last Modified by GUIDE v2.5 22-May-2009 21:15:05

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @AI_OpeningFcn, ...
                  'gui_OutputFcn',  @AI_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before AI is made visible.
function AI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
```



```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to AI (see VARARGIN)

% Choose default command line output for AI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes AI wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = AI_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1. %% NACITAJ vzor
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
%%
% otvorenie dialogoveho okna pre nacitanie suboru
[fname, fpath] = uigetfile('*.wav', ['Zadaj cestu k suboru s PSD prjmaneho
signalu...']);
subor=strcat(fpath, fname);
save subor;
%%
% "zazni" plochu pre vykreslenie priebehu a buttony 2, 4 a 7
set(handles.axes1, 'Visible', 'on');
set(handles.pushbutton2, 'Visible', 'on');
set(handles.pushbutton4, 'Visible', 'on');
set(handles.pushbutton7, 'Visible', 'on');
set(handles.uipanel2, 'Visible', 'on');
set(handles.uipanel3, 'Visible', 'on');
set(handles.uipanel4, 'Visible', 'on');

y=wavread(subor);
    grid on;
    plot(y);

    title('priebeh signalu');
    xlabel('\it dlzka signalu} [ vzorky
] \rightarrow', 'FontName', 'Arial');
    ylabel('\it A} [-] \rightarrow', 'FontName', 'Arial');
    set(get(gcf, 'CurrentAxes'), 'FontName', 'Arial')

% --- Executes on button press in pushbutton2. %% PREHRAJ referencny
function pushbutton2_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
load subor;
[data fs]=wavread(subor);
sound(data, fs);

% --- Executes on button press in pushbutton3. %% ANALYZUJ
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.text2, 'Visible', 'on');
set(handles.text3, 'Visible', 'on');
load subor;
load subor1;
% global X
% global fs
[data, fs] = wavread(subor);
[data_d, fs] = wavread(subor1);
%save data
w=get(handles.listbox1, 'Value');
if w==1 wind=128; end
if w==2 wind=256; end
if w==3 wind=512; end
if w==4 wind=1024; end

o=get(handles.listbox2, 'Value');
if o==1 overlen=0; end
if o==2 overlen=64; end
if o==3 overlen=128; end
if o==4 overlen=256; end

Y=segmentace(data, wind, overlen);
[M N]=size(Y);

[AI AI_seg]=A_index(data, fs, wind, overlen, data_d);
set(handles.text1, 'String', AI);
grid on;
plot(AI_seg*100);
axis([1 N 0 100]);
title('AI v jednotlivych ramcoch');
xlabel('{\it cislo ramca} [-] \rightarrow');
ylabel('{\it Artikulacny Index} [%] \rightarrow');

% --- Executes on selection change in listbox1.
function listbox1_Callback(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject, 'String') returns listbox1 contents as
cell array
%         contents{get(hObject, 'Value')} returns selected item from
listbox1

```

```

% --- Executes during object creation, after setting all properties.
function listbox1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: listbox controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in listbox2.
function listbox2_Callback(hObject, eventdata, handles)
% hObject    handle to listbox2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns listbox2 contents as
cell array
%         contents{get(hObject,'Value')} returns selected item from
listbox2

% --- Executes during object creation, after setting all properties.
function listbox2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listbox2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: listbox controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

end

```
% --- Executes on button press in pushbutton4. %% NACITAJ DEGRAD
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[fname,fpath] = uigetfile('*.wav',['Zadaj cestu k suboru s PSD prjimaneho
signalu...']);
subor1=strcat(fpath,fname);
save subor1;

%%%
%"zazni"
set(handles.pushbutton5,'Visible','on');
set(handles.pushbutton3,'Visible','on');
set(handles.pushbutton8,'Visible','on');
set(handles.listbox1,'Visible','on');
set(handles.text4,'Visible','on');
set(handles.listbox2,'Visible','on');
set(handles.text5,'Visible','on');

data_d=wavread(subor1);
    grid on;
    plot(data_d);

    title('priebeh signalu');
    xlabel('\it dlzka signalu} [ vorky
]\rightarrow','FontName','Arial');
    ylabel('\it A} [-] \rightarrow','FontName','Arial');
    set(get(gcf,'CurrentAxes'),'FontName','Arial')

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
load subor1;
[data_d fs]=wavread(subor1);
sound(data_d, fs);

% --- Executes on button press in pushbutton8. %% SPEKTROGRAM
DEGRADOVANEHO
function pushbutton8_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
load subor1
[data_d fs]=wavread(subor1)
spectrogram(data_d,256,192,256, fs);
title('spektrogram testovaneho signálu');
ylabel('\it t} [s] \rightarrow');
xlabel('\it Frekvencia} [Hz] \rightarrow');
```

```

% --- Executes on button press in pushbutton7. %% SPEKTROGRAM VZORU
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton7 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
load subor;
[data fs]=wavread(subor)
spectrogram(data,256,192,256,fs);
title('spektrogram vzoru');
ylabel('\it t} [s] \rightarrow');
xlabel('\it Frekvencia} [Hz] \rightarrow');

```

WSS.m

```

function varargout = WSSM(varargin)
% WSSM M-file for WSSM.fig
%   WSSM, by itself, creates a new WSSM or raises the existing
%   singleton*.
%
%   H = WSSM returns the handle to a new WSSM or the handle to
%   the existing singleton*.
%
%   WSSM('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in WSSM.M with the given input arguments.
%
%   WSSM('Property','Value',...) creates a new WSSM or raises the
%   existing singleton*. Starting from the left, property value pairs
are
%   applied to the GUI before WSSM_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property
application
%   stop. All inputs are passed to WSSM_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only
one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help WSSM

% Last Modified by GUIDE v2.5 22-May-2009 22:06:20

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @WSSM_OpeningFcn, ...
                  'gui_OutputFcn',  @WSSM_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else

```

```

        gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before WSSM is made visible.
function WSSM_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to WSSM (see VARARGIN)

% Choose default command line output for WSSM
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes WSSM wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = WSSM_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1. %% NACITAJ VZOR
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%%
% otvorenie dialogoveho okna pre nacitanie suboru
[fname,fpath] = uigetfile('*.*wav', ['Zadaj cestu k suboru s PSD prjmaneho
signalu...']);
subor=strcat(fpath,fname);
save subor;
%%
% "zazni" plochu pre vykreslenie priebehu a buttony 2, 4 a 6
set(handles.axes1, 'Visible', 'on');
set(handles.pushbutton2, 'Visible', 'on');
set(handles.pushbutton4, 'Visible', 'on');
set(handles.pushbutton6, 'Visible', 'on');

y=wavread(subor);
    grid on;
    plot(y);

    title('priebeh signalu');
    xlabel('{\it dlzka signalu} [ vzorky
]\rightarrow', 'FontName', 'Arial');

```

```

ylabel({'\it A} [-] \rightarrow' , 'FontName', 'Arial');
set(get(gcf, 'CurrentAxes'), 'FontName', 'Arial')

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
load subor;
[data fs]=wavread(subor);
sound(data, fs);

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.text3, 'Visible', 'on');
set(handles.text1, 'Visible', 'on');
load subor;
load subor1;
% global X
% global fs
[data, fs] = wavread(subor);
[data_d, fs] = wavread(subor1);
%save data
w=get(handles.listbox1, 'Value');
if w==1 wind=128; end
if w==2 wind=256; end
if w==3 wind=512; end
if w==4 wind=1024; end

o=get(handles.listbox2, 'Value');
if o==1 overlen=0; end
if o==2 overlen=64; end
if o==3 overlen=128; end
if o==4 overlen=256; end

if fs==8000 nfilts = 17; end
if fs==11000 nfilts = 19; end
if fs==16000 nfilts = 21; end
if fs==22000 nfilts = 23; end
if fs==44100 nfilts = 27; end

width = 1;
minfreq = 0;
maxfreq = fs/2;

WSSM = WSSM_gui(data, fs, wind, overlen, nfilts, width,
minfreq, maxfreq, data_d)

set(handles.text1, 'String', WSSM);

```

```

% --- Executes on selection change in listbox1.
function listbox1_Callback(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns listbox1 contents as
cell array
%         contents{get(hObject,'Value')} returns selected item from
listbox1

% --- Executes during object creation, after setting all properties.
function listbox1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: listbox controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in listbox2.
function listbox2_Callback(hObject, eventdata, handles)
% hObject    handle to listbox2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns listbox2 contents as
cell array
%         contents{get(hObject,'Value')} returns selected item from
listbox2

% --- Executes during object creation, after setting all properties.
function listbox2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listbox2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: listbox controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)

```



```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
% str2double(get(hObject,'String')) returns contents of edit1 as a
double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
[fname,fpath] = uigetfile('*.wav',['Zadaj cestu k suboru s PSD prjimaneho
signalu...']);
subor1=strcat(fpath,fname);
save subor1;

%%%
%"zazni"
set(handles.pushbutton5,'Visible','on');
set(handles.pushbutton7,'Visible','on');
set(handles.pushbutton3,'Visible','on');
set(handles.listbox1,'Visible','on');
set(handles.text4,'Visible','on');
set(handles.listbox2,'Visible','on');
set(handles.text5,'Visible','on');

data_d=wavread(subor1);
grid on;
plot(data_d);

title('priebeh signalu');
xlabel('\it dlzka signalu} [ vorky
]\rightarrow','FontName','Arial');
ylabel('\it A} [-] \rightarrow','FontName','Arial');
set(get(gcf,'CurrentAxes'),'FontName','Arial')

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton5 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
load subor1;
[data_d fs]=wavread(subor1);
sound(data_d, fs);

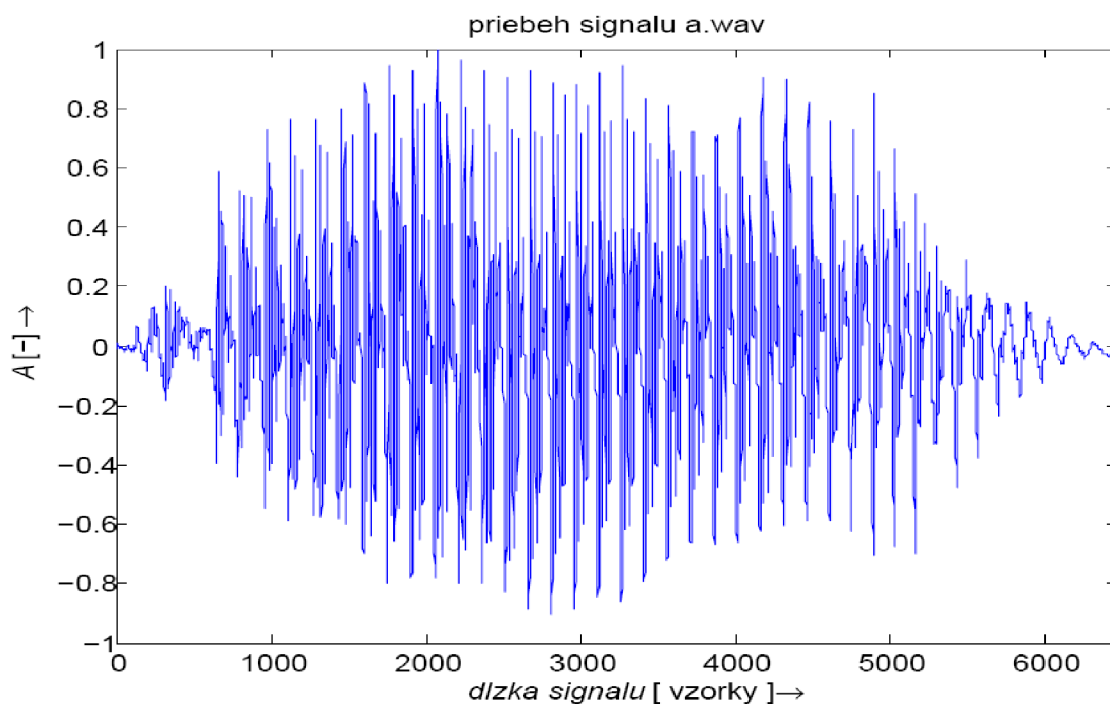
% --- Executes on button press in pushbutton6. %% SPEKTROGRAM VZORU
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
load subor;
[data fs]=wavread(subor);
spectrogram(data,256,192,256, fs);
title('spektrogram vzoru');
ylabel('{\it t} [s] \rightarrow');
xlabel('{\it Frekvencia} [Hz] \rightarrow');

% --- Executes on button press in pushbutton7. %% SPEKTROGRAM
DEGRADOVANEHO
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton7 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
load subor1
[data_d fs]=wavread(subor1);
spectrogram(data_d,256,192,256, fs);
title('spektrogram testovaneho signálu');
ylabel('{\it t} [s] \rightarrow');
xlabel('{\it Frekvencia} [Hz] \rightarrow');

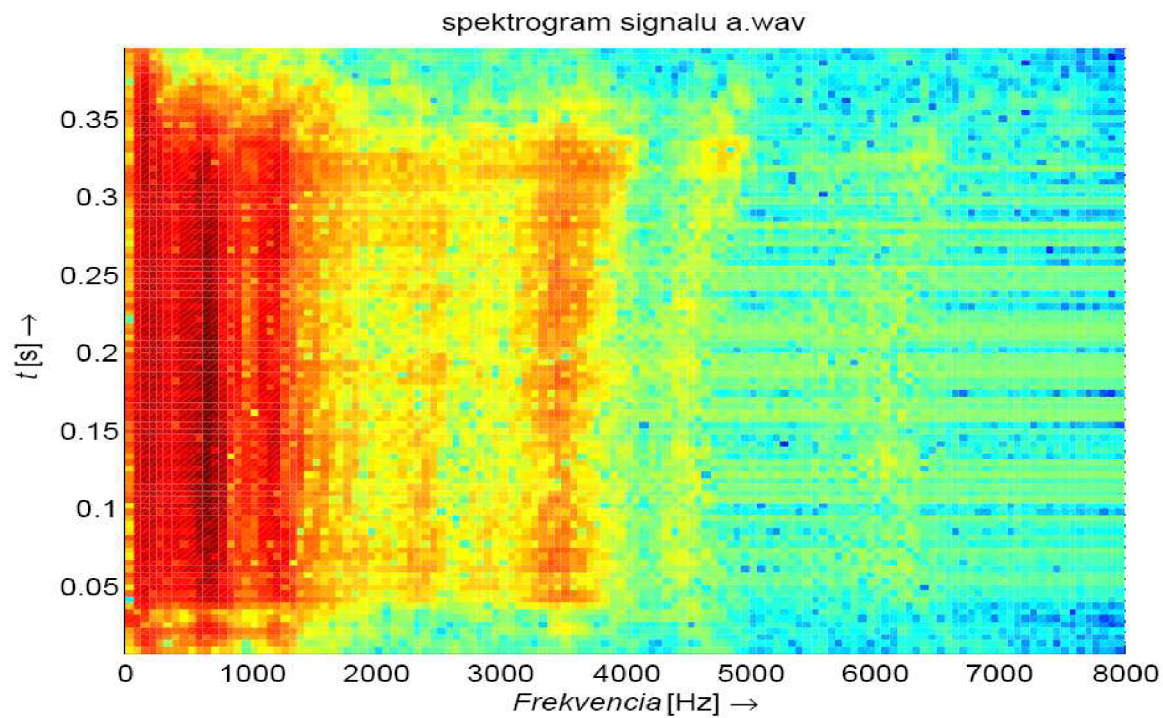
```

B Druhá Príloha

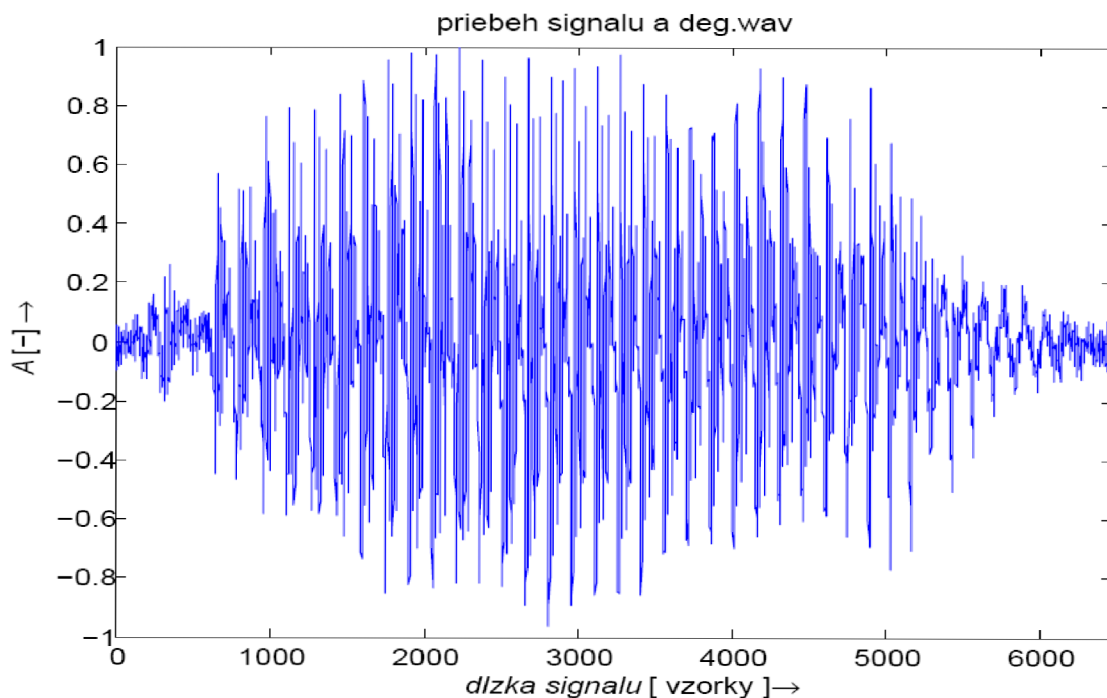
V druhej prílohe sú priebehy a spektrogramy všetkých testovaných signálov



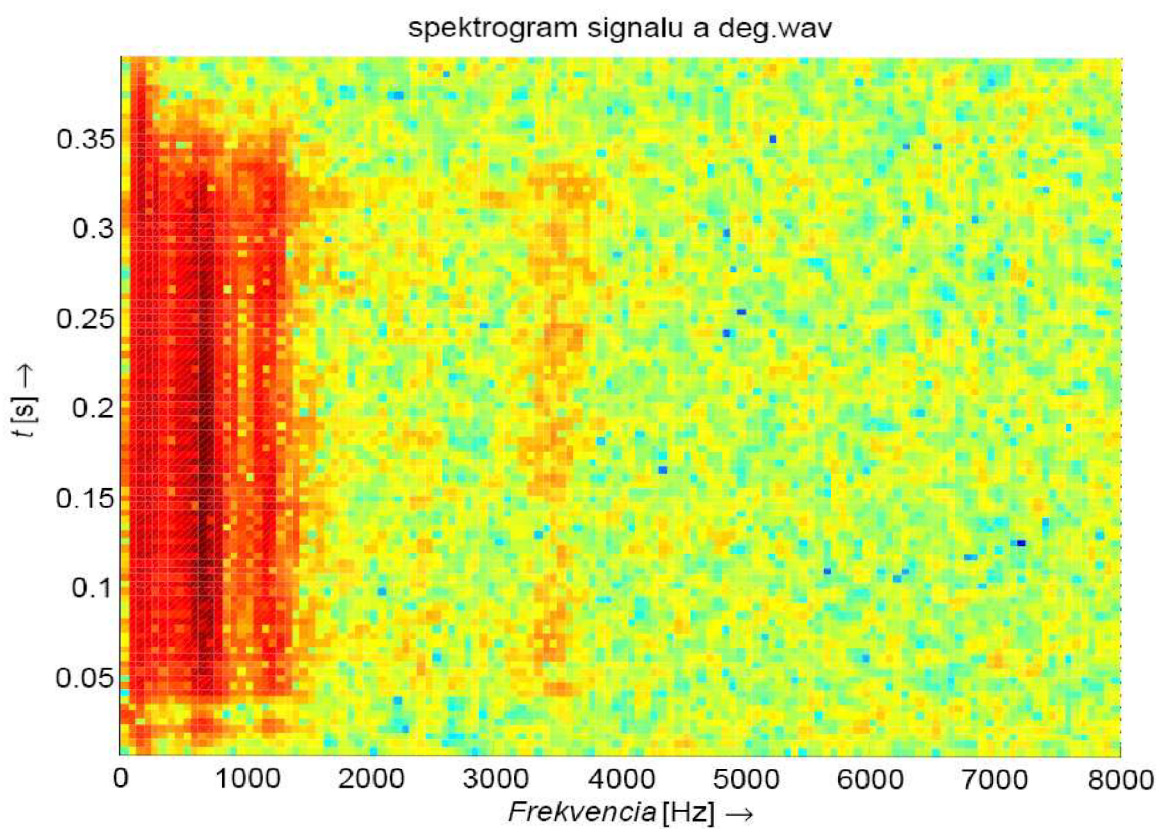
Obrázok 19: Priebeh signálu a.wav



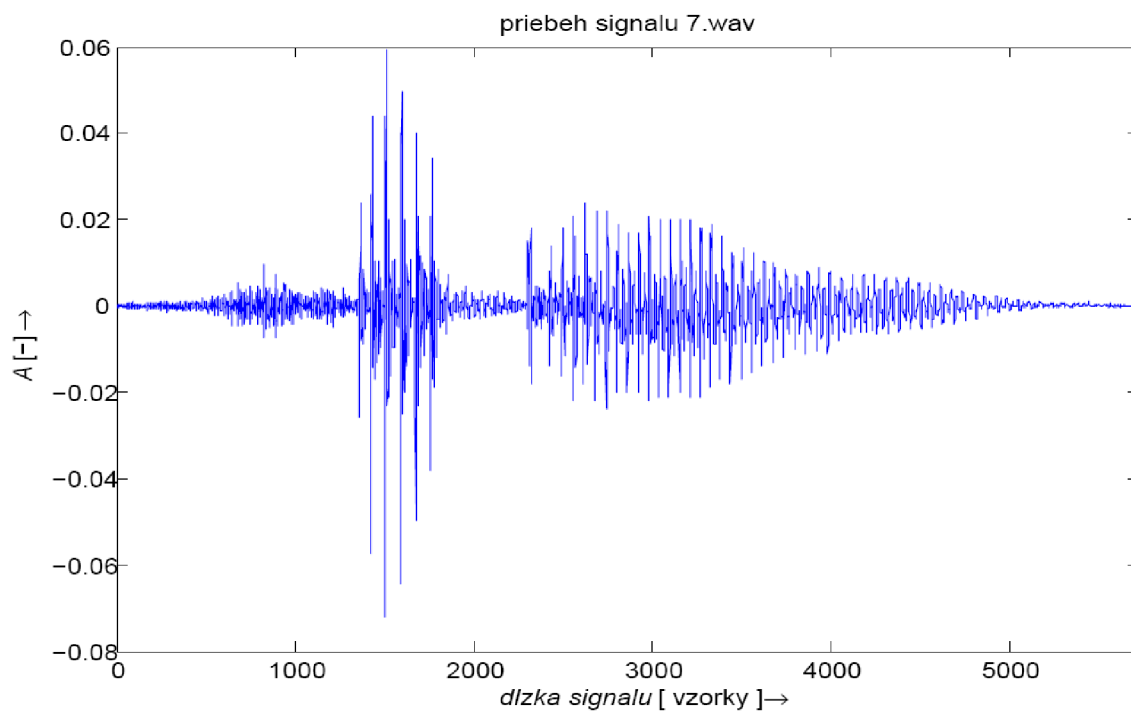
Obrázok 20: Spektrogram signálu a.wav



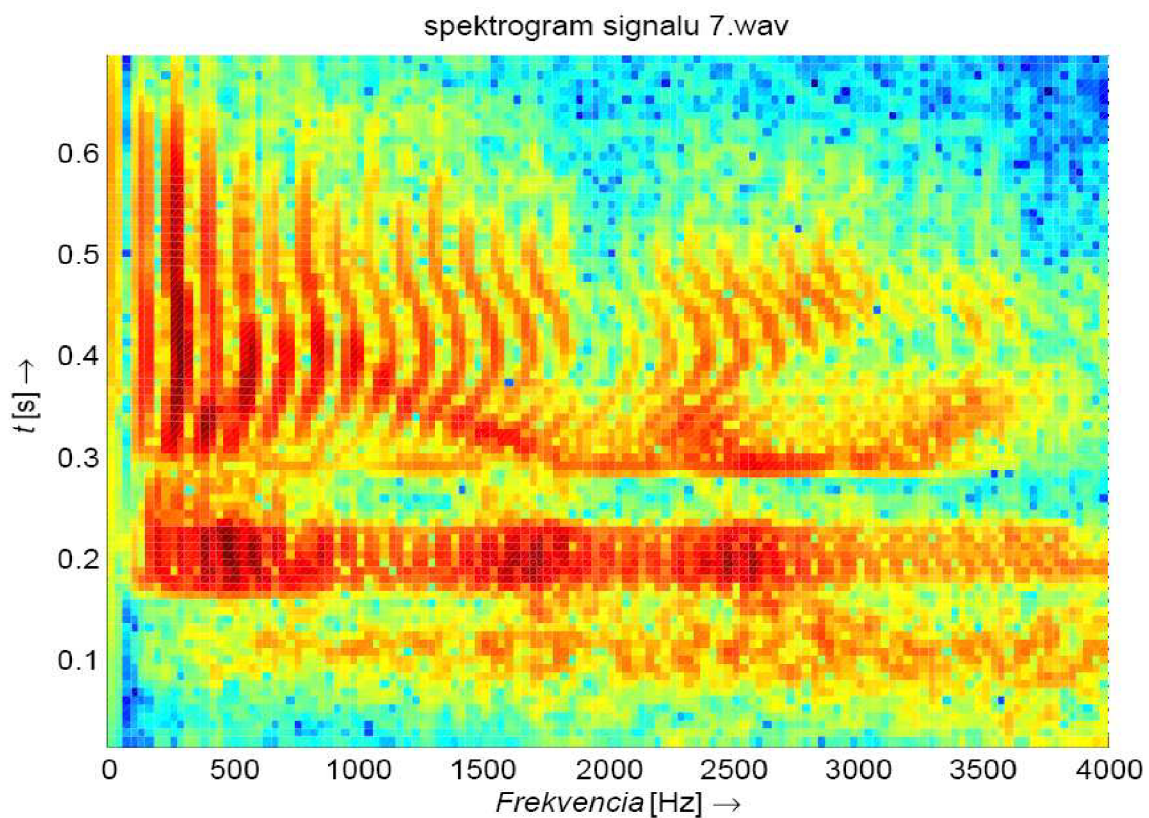
Obrázok 21: Priebek signálu a_deg.wav



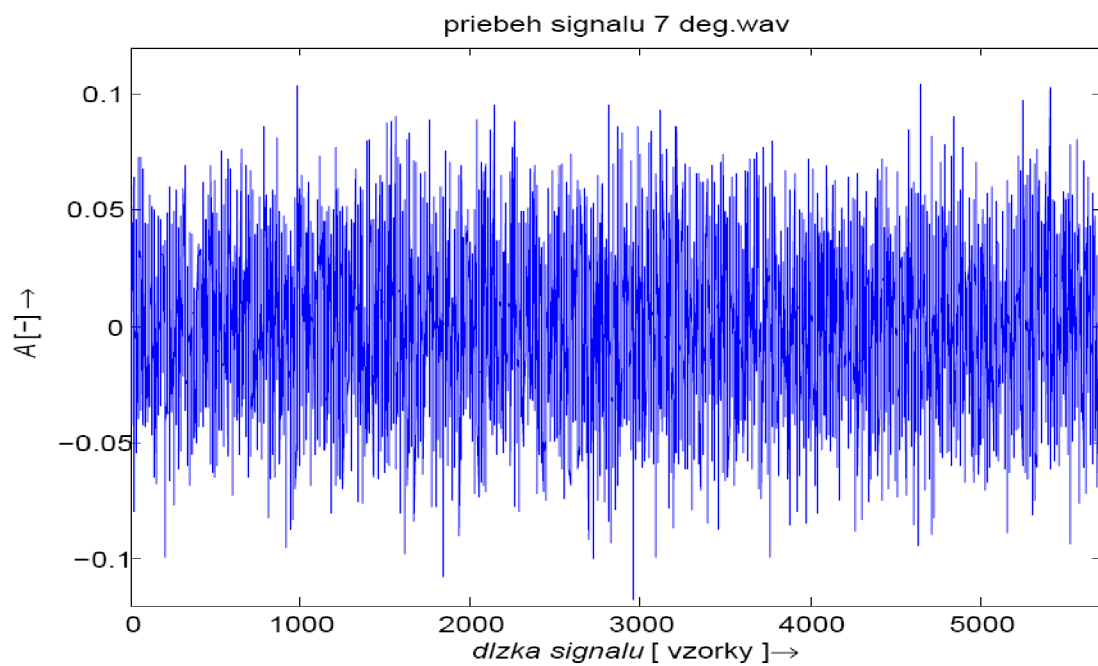
Obrázok 22: Spektrogram signálu a_deg.wav



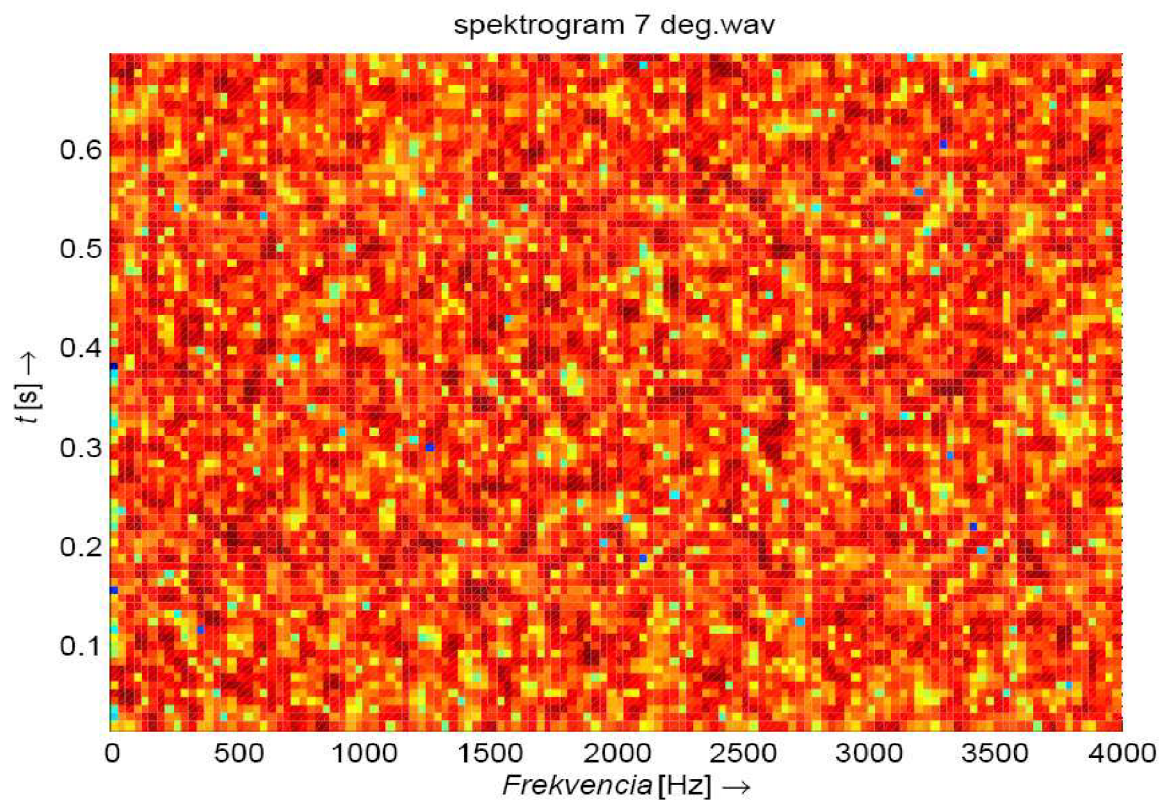
Obrázok 23: Priebeh signálu 7.wav



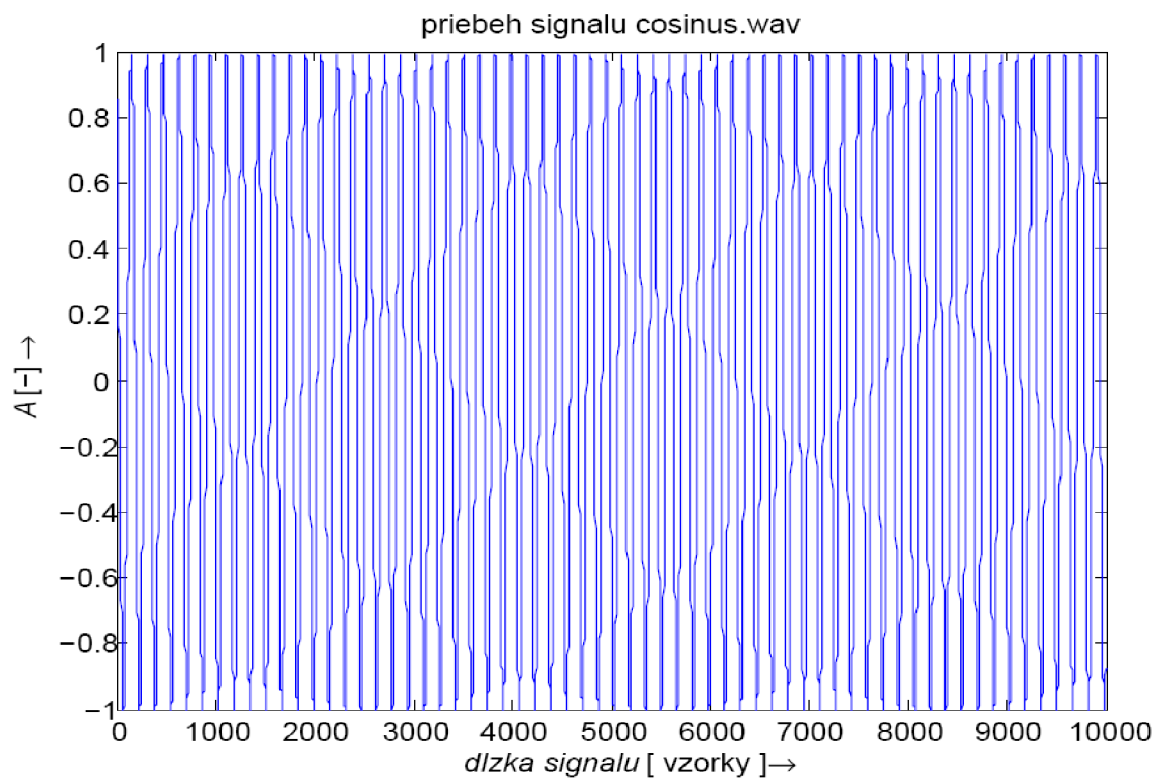
Obrázok 24: Spektrogram signálu 7.wav



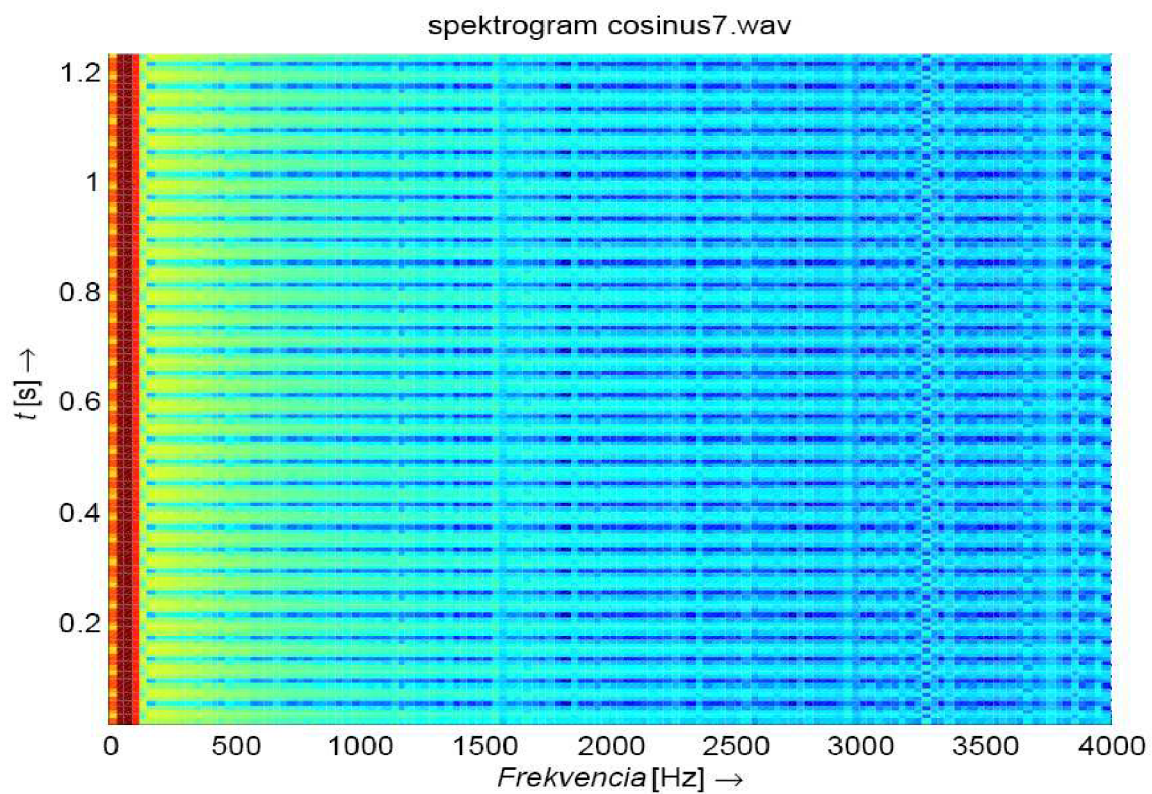
Obrázok 25: Priebek signálu 7_deg.wav



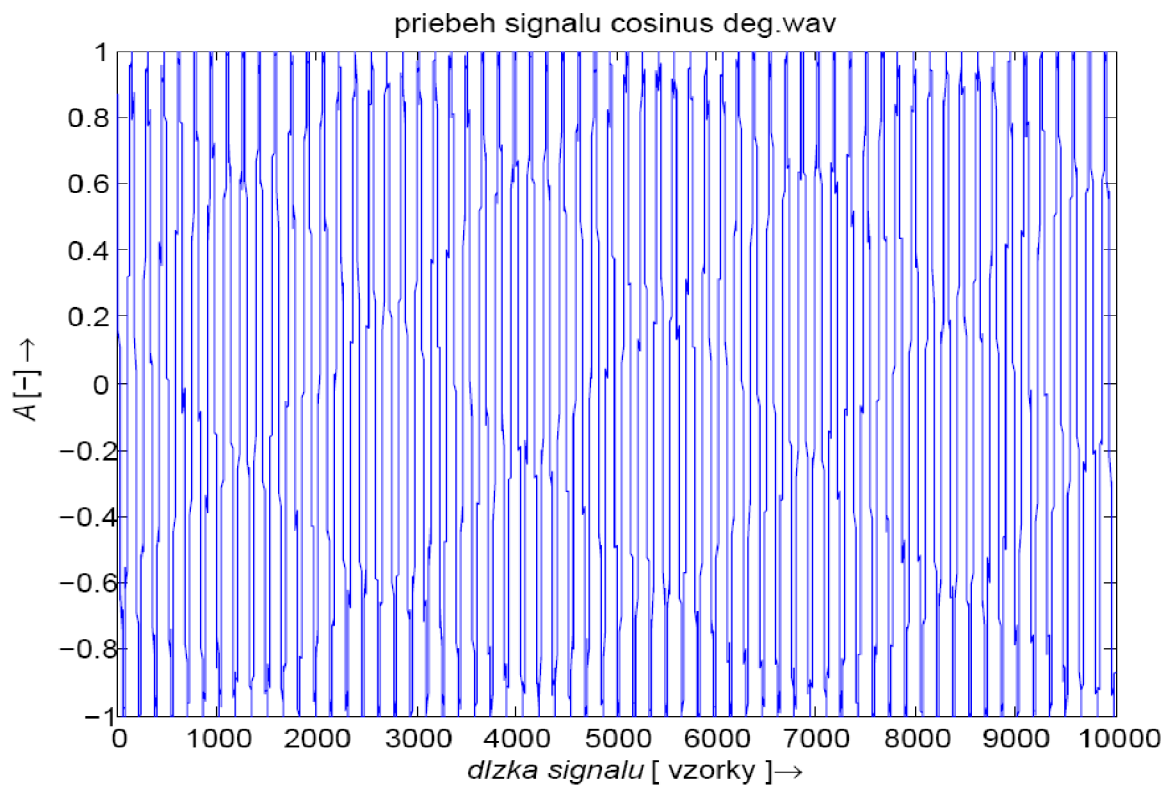
Obrázok 26: Spektrogram signálu 7_deg.wav



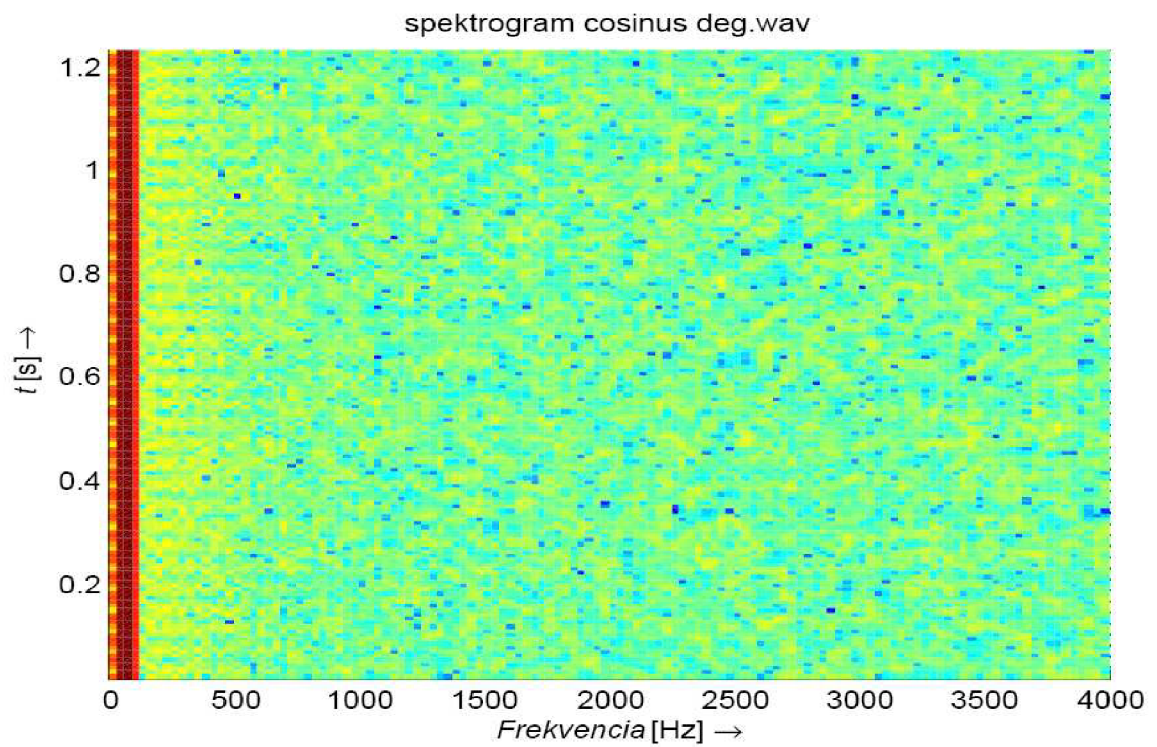
Obrázok 27: Priebeh signálu cosinus.wav



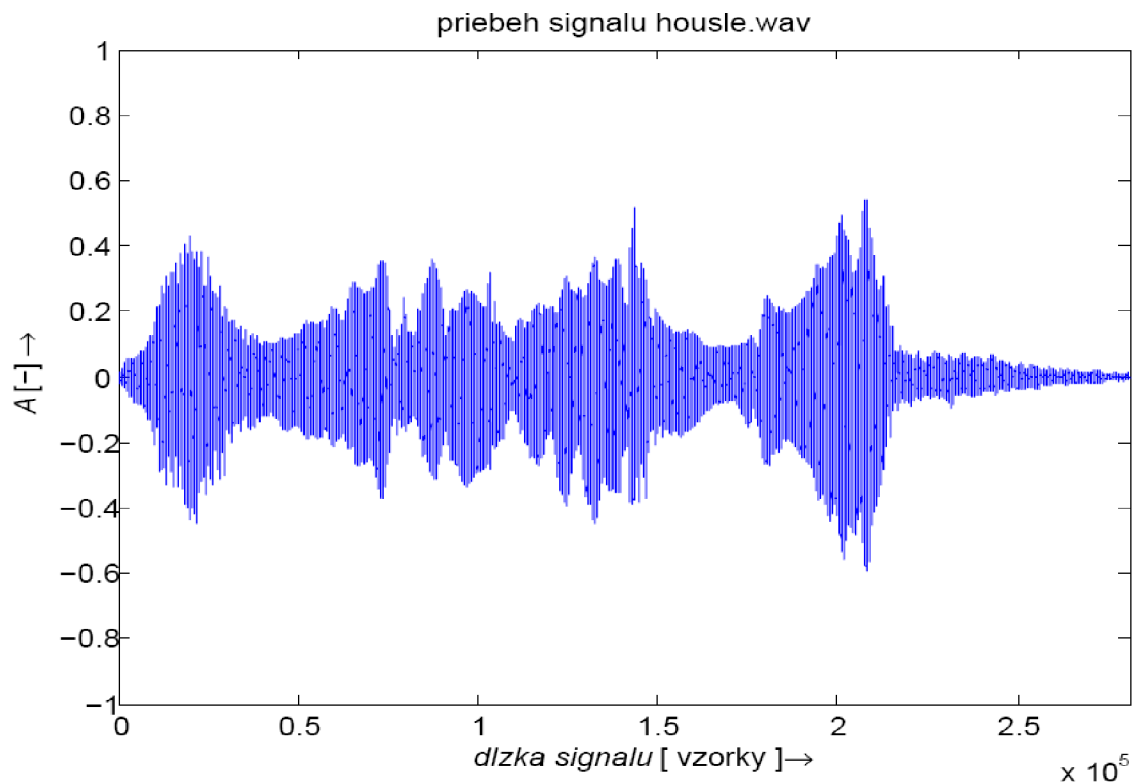
Obrázok 28: Spektrogram signálu cosinus.wav



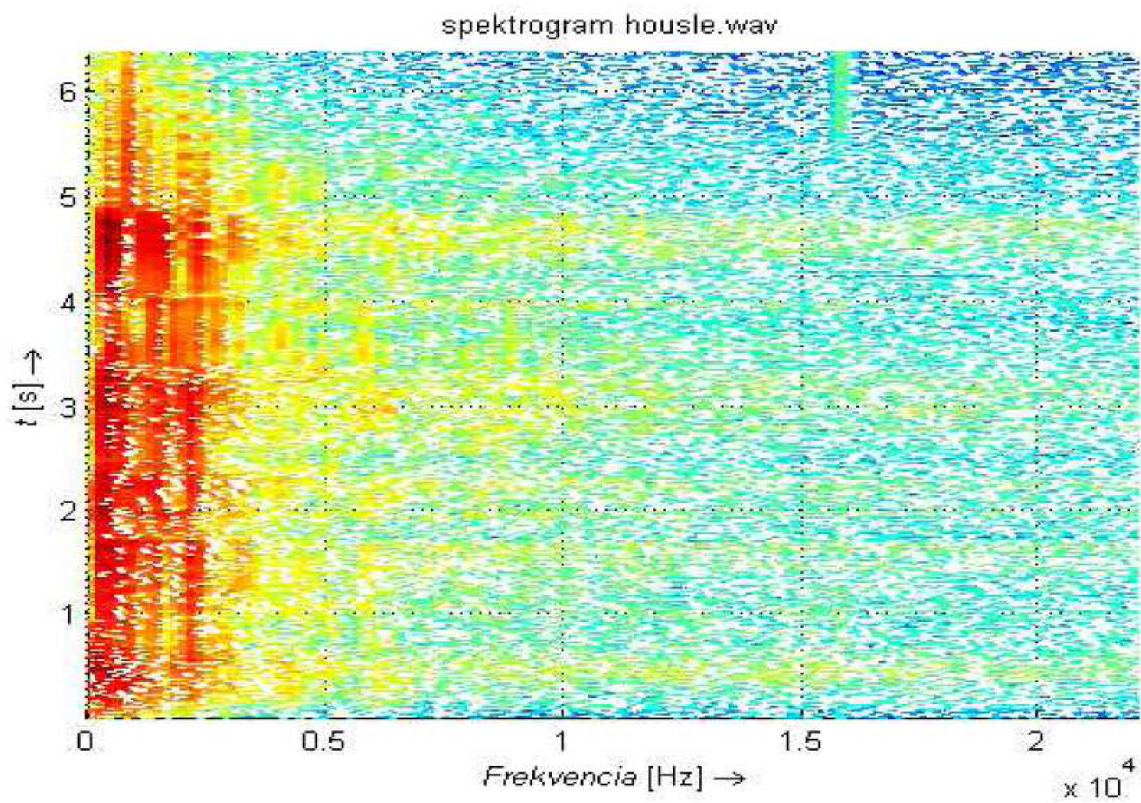
Obrázok 29: Priebek signálu cosinus_deg.wav



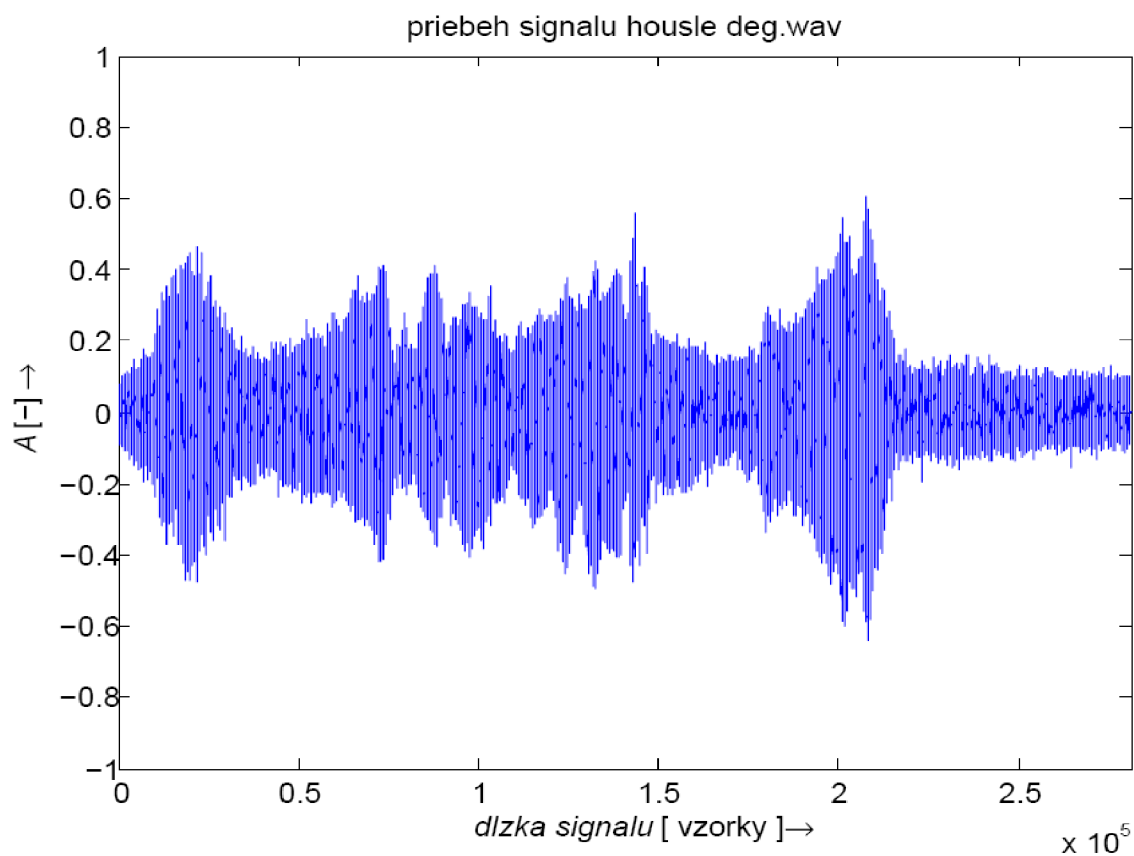
Obrázok 30: Spektrogram signálu cosinus_deg.wav



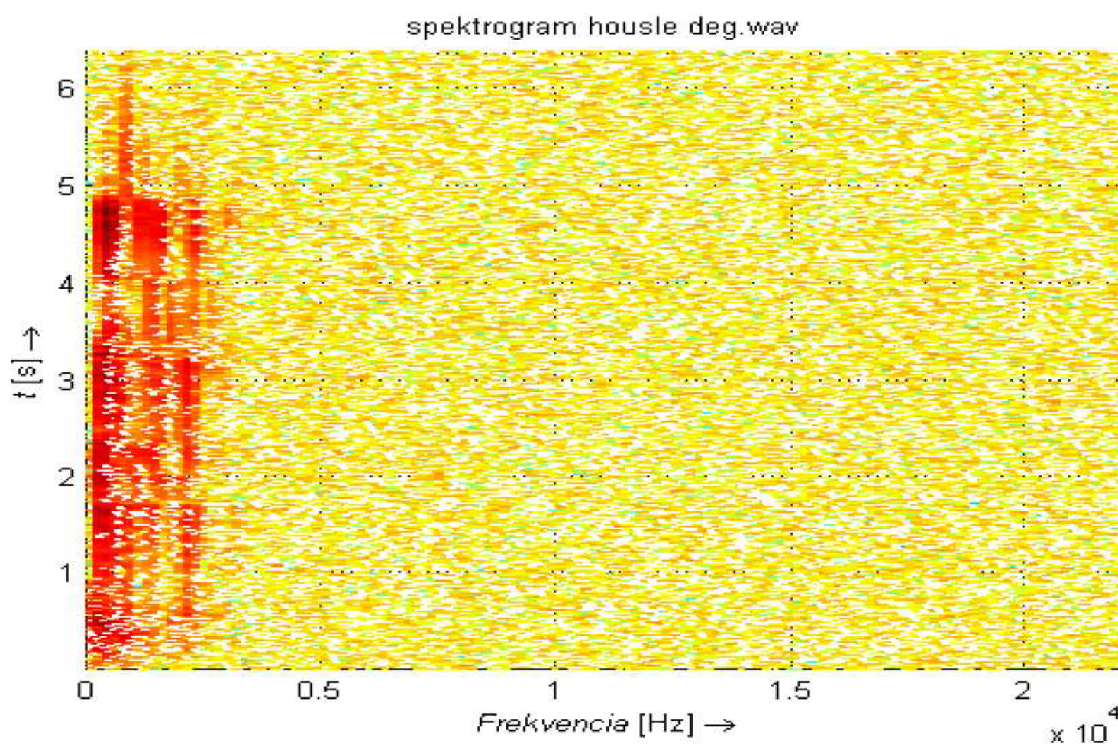
Obrázok 31: Priebek signálu housle.wav



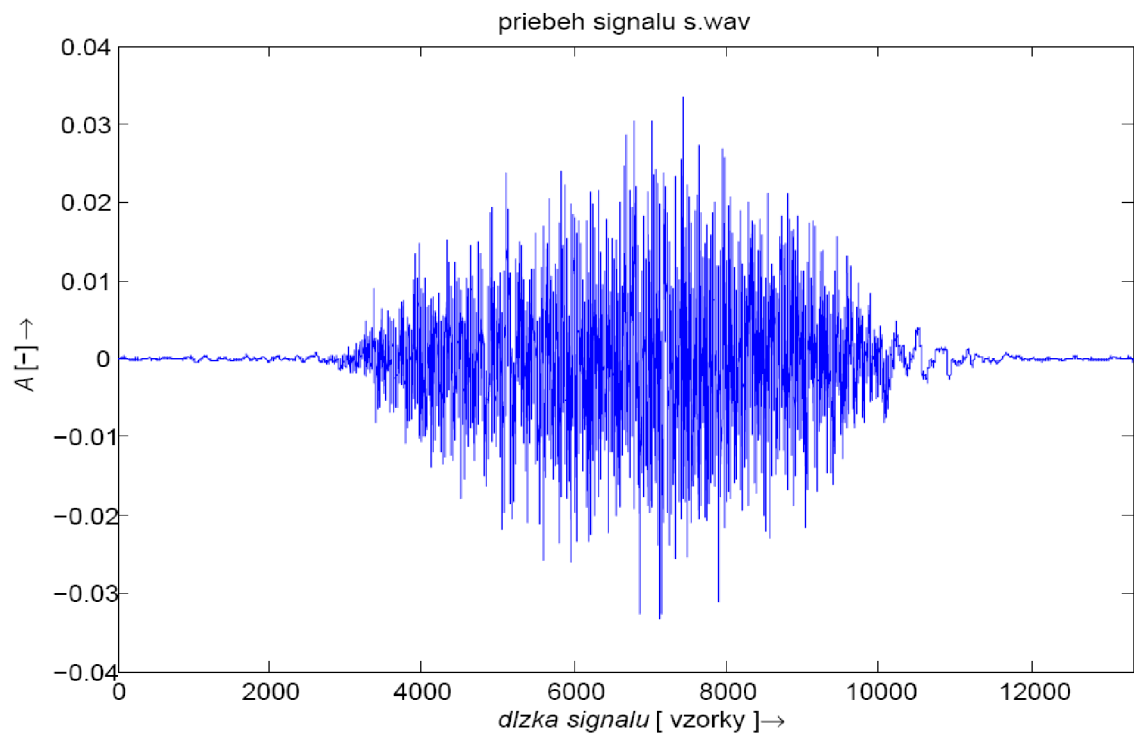
Obrázok 32: Spektrogram signálu housle.wav



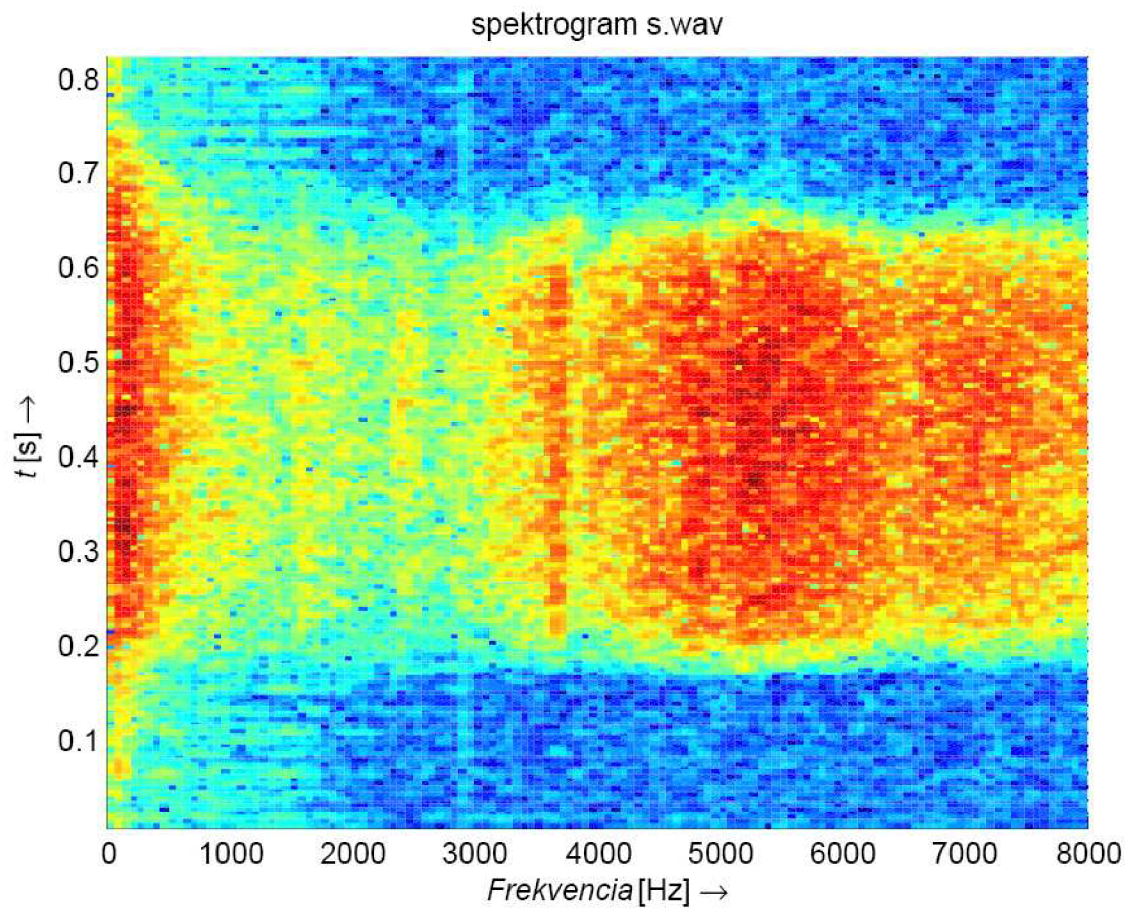
Obrázok 33: Priebek signálu house_deg.wav



Obrázok 34: Spektrogram signálu house_deg.wav



Obrázok 35: Priebek signálu s.wav



Obrázok 36: Spektrogram signálu s.wav

