

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

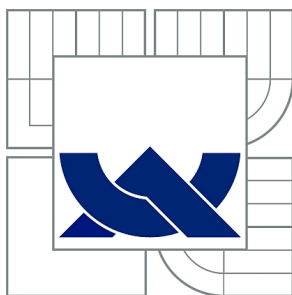
MODERNÍ KÓDOVÁNÍ ŘEČOVÉHO SIGNÁLU POMOCÍ
PŘEPARAMETRIZOVANÝCH MODELŮ

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

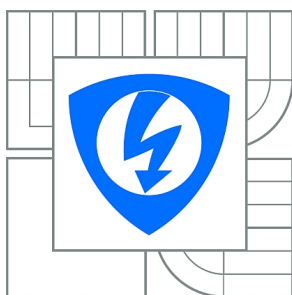
BC. ONDŘEJ ZAPLETAL

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

MODERNÍ KÓDOVÁNÍ ŘEČOVÉHO SIGNÁLU POMOCÍ PŘEPARAMETRIZOVANÝCH MODELŮ

MODERN CODING OF SPEECH SIGNALS USING OVERCOMPLETE MODELS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

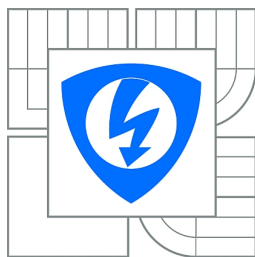
Bc. ONDŘEJ ZAPLETAL

VEDOUCÍ PRÁCE

SUPERVISOR

Mgr. PAVEL RAJMIC, Ph.D.

BRNO 2010



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Ondřej Zapletal

ID: 78368

Ročník: 2

Akademický rok: 2009/2010

NÁZEV TÉMATU:

Moderní kódování řečového signálu pomocí přeparametrizovaných modelů

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte základy teorie tzv. frejmů a srovnajte základní vlastnosti této transformace s waveletovou nebo Fourierovou transformací. Seznamte se s knihovnou funkcí "frames" pro MATLAB. Pomocí této knihovny vytvořte software, který realizuje alternativu k obvyklému lineárně-predikčnímu kódování řečových segmentů. Srovnajte účinnost a výpočetní náročnost.

DOPORUČENÁ LITERATURA:

- [1] Christensen, O.: An Introduction to Frames nad Riesz Bases. Birkhäuser, Boston-Basel-Berlin, 2003.
- [2] Chen, S. S., Donoho, D. L., Saunders, M. A.: Atomic decomposition by basis pursuit, SIAM J. Sci. Comput. 20 (1998), no. 1, 33{61, reprinted in SIAM Review, 43 (2001), no. 1, pp. 129-159.
- [3] Veselý, V., Tonner, J.: Sparse Parameter Estimation in Overcomplete Time Series Models. Austrian Journal of Statistics, Vol. 35 (2006), Number 2&3, 371-378. ISSN 1026-597X

Termín zadání: 29.1.2010

Termín odevzdání: 26.5.2010

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

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Náplní teoretické části této práce je studie přeparametrizovaných modelů. To jsou takové modely signálů, ve kterých je pro jejich parametrizaci stanoveno více proměnných, než je potřeba a následně se hledá tzv. řídké řešení pomocí iteračních algoritmů. Cílem takovéto analýzy je výběr pouze těch důležitých (řídkých) parametrů. Teorie se opírá o lineární algebru, vektorové prostory, báze a tzv. framy. Úkolem samostatného projektu této práce je popis a simulace dvou řečových kodérů: klasického kodéru na bázi lineárního predikčního kódování řeči a kodéru využívajícího přeparametrizované modely pro náhodné ARMA procesy. Součástí jejich realizace je i vytvoření dekodérů a zhodnocení kvality rekonstrukce obou z nich. Pro realizaci je využito prostředí MATLAB a knihovna funkcí pro přeparametrizované modely (toolbox „frames“).

KLÍČOVÁ SLOVA

řeč, signál, model, kodér, koeficient, přeparametrizovaný, frame, pseudoinverze, autokorelace, filtr, spektrum, šum, predikce, energie, rozptyl

ABSTRACT

The theoretical contents of this thesis are studies of overcomplete models. Those are the models of signals, on which is set for their parametrization more variables, than it's necessary and consequently there's computed so-called sparse solution via iteration algorithms. A goal of this analysis is a selection just of the considerable (sparse) parameters. The theory is based on a linear algebra, vector spaces, bases and so-called frames. The task of the individual project of this thesis is a description and simulation of two speech coders: a classical coder based on linear predictive speech coding and a coder, that's making use of overcomplete stochastic ARMA processes models. A part of their realization is to simulate their decoders and analyze their reconstruction quality. For their realization there is used MATLAB and an overcomplete models' library (toolbox frames).

KEYWORDS

speech, signal, model, coder, coefficient, overcomplete, frame, pseudoinverse, autocorrelation, filter, spectrum, noise, prediction, energy, spread

ZAPLETAL, O. *Moderní kódování řečového signálu pomocí přeparametrizovaných modelů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 72 s. Vedoucí diplomové práce Mgr. Pavel Rajmic, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Moderní kódování řečového signálu pomocí přeparametrizovaných modelů“ 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 této diplomové práce 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)

PODĚKOVÁNÍ

Děkuji vedoucímu této diplomové práce **Mgr. Pavlu Rajmíci, Ph.D.** za poskytnuté konzultace a dodané literární podklady. Dále děkuji **doc. RNDr. Vítězslavu Veselému, CSc.** z Ekonomicko-správní fakulty Masarykovy univerzity v Brně za dodaný software, teoretické podklady a speciální semináře ohledně problematiky přeparametřovaných modelů.

V Brně dne

.....

(podpis autora)

OBSAH

Úvod	11
1 Vektorový prostor	12
1.1 Generátory vektorového prostoru	12
1.2 Báze vektorového prostoru	13
1.3 Příklady vektorových prostorů	14
1.3.1 Hilbertův prostor	14
1.4 Framy	15
2 Přeparametrizované systémy lineárních rovnic a hledání řídkého řešení	17
2.1 Pseudoinverze	17
2.2 Algoritmy hledání řídkého řešení, BPA	20
3 Přeparametrizované modely signálů	21
3.1 Modelování signálů pomocí atomů	21
3.2 Modelování pomocí časových řad	22
3.2.1 Klasické ARMA modely	23
3.2.2 Přeparametrizované ARMA modely	23
3.3 Porovnání s Fourierovou a waveletovou transformací	25
4 Toolbox frames v prostředí MATLAB	27
4.1 Atomy	27
4.1.1 Generování atomů	27
4.1.2 Lorentzova funkce	28
4.2 Základní funkce a možnosti toolboxu	29
5 Úvod do parametrizace řeči	30
5.1 Vlastnosti řečového signálu	30
5.2 Segmentace řeči	31
5.3 LPC analýza segmentu řeči	32
5.4 Řečové kodéry	33
6 Návrh a realizace kodérů	35
6.1 Kodér LPC	35
6.1.1 Kodér	35
6.1.2 Dekodér	37
6.1.3 Návrh kódování do bitového toku	40

6.2	Kodér s přeparametrizovanými ARMA modely	41
6.2.1	Kodér	41
6.2.2	Dekodér	43
6.2.3	Návrh kódování do bitového toku	45
7	Výsledky realizace	47
7.1	Charakter řídkých ARMA koeficientů	47
7.2	Porovnání kvality rekonstrukce obou kodérů	50
7.2.1	Metody porovnávání kvality rekonstrukce	50
7.2.2	Výsledky práce simulovaných kodérů	53
7.3	Obrazové ukázky výstupů kodérů	55
7.4	Výhody, nevýhody a možná vylepšení kodérů	58
8	Závěr	61
	Literatura	62
	Seznam symbolů, veličin a zkratk	64
	Seznam příloh	68
A	Přiložený software	69
A.1	Seznam adresářů a souborů	69
A.2	Dema a spustitelné skripty	71

SEZNAM OBRÁZKŮ

1.1	Příklad jednoduchého framu a jeho duální frame	15
4.1	Lorentzova funkce v základním tvaru	28
5.1	Tvorba řeči a její model. Obrázek převzat z [4]	30
5.2	Model pro tvorbu řečového signálu	31
6.1	Blokové schéma kodéru LPC	35
6.2	Blokové schéma dekodéru LPC	38
6.3	Práce s impulzní charakteristikou při rekonstrukci znělého segmentu .	40
6.4	Bitový rámec pro jeden segment z LPC kodéru	41
6.5	Blokové schéma kodéru využívajícího přeparametrizované ARMA modely	42
6.6	Blokové schéma dekodéru využívajícího přeparametrizované ARMA modely	43
6.7	Realizace ARMA procesu a jeho autokorelace	45
6.8	Bitový rámec pro jeden segment z ARMA kodéru	46
7.1	ARMA koeficienty pro znělý segment pro $P = Q = 10$	48
7.2	ARMA koeficienty pro znělý segment pro $P = Q = 20$	48
7.3	ARMA koeficienty pro neznělý segment pro $P = Q = 10$	49
7.4	ARMA koeficienty pro přechodný segment pro $P = Q = 10$	49
7.5	Ukázka – kódování neznělého segmentu LPC kódérem	55
7.6	Ukázka – kódování znělého segmentu ARMA kódérem	56
7.7	Ukázka – průběh celé originální nahrávky i její ARMA rekonstrukce .	56
7.8	Ukázka – průběhy znělého segmentu originální, LPC i ARMA rekonstrukce nahrávky	57
7.9	Ukázka – autokorelace a spektrální obálka znělého segmentu (ARMA)	57
7.10	Ukázka – průběh ACF a WSS pro nahrávku <code>policie1.wav</code>	58

SEZNAM TABULEK

7.1	Úroveň kvality rekonstrukce dle \overline{ACFx}	52
7.2	Úroveň kvality rekonstrukce dle \overline{WSS}	53
7.3	Porovnání kvality rekonstrukce simulovaných kodérů	54

ÚVOD

V dnešní době patří mezi nejpoužívanější přístupy ke kompresi a modelování signálů Fourierova transformace, waveletová transformace či LPC (*Linear Predictive Coding*) analýza v případě kódování řeči. Ve všech těchto případech jsou při analýze signálu vypočítávány koeficienty, které nadále onen signál či jeho část reprezentují. Tyto stávající metody se snaží co nejdokonaleji popsat signál pomocí předem daného počtu parametrů, který určuje přesnost modelu. Vývoj směřuje k tomu, že počet takových koeficientů se zmenšuje za cenu výpočetní náročnosti. V poslední době se objevily přístupy, u kterých se ukazuje, že když se modelování úseku signálu přeparametrizuje (tzn. je stanoveno více proměnných, než je potřeba), tak výsledný odhad paradoxně může dávat lepší shodu s modelovaným signálem. Tím, že se vytvoří nadbytečné parametry, tak je umožněno z nich vybrat jen ty nejdůležitější s největším podílem na podobě signálu. Narozdíl od např. Fourierovy transformace se neomezuje jen na harmonickou funkci, ale můžeme koeficienty vztahovat k jedné ze široké palety funkcí, tzv. atomů, které si pro modelování signálu vytvoříme. Dnes existují algoritmy, které dokážou vybrat velmi málo takových nejdůležitějších parametrů, což má za následek snížení bitového toku při kódování či zvýšení kvality odhadu signálu.

První částí této práce je studium takovýchto přeparametrizovaných modelů, což obnáší kapitoly z lineární algebry, následně zpracování parametrů vedoucí k nalezení řídkého řešení – oněch pár důležitých koeficientů. Dále je zde představen toolbox *frames* v prostředí MATLAB, kde je již celá tato problematika naprogramována a který již máme hotový k dispozici. Hlavní částí této práce je návrh a simulace řečového kodéru pracujícího s tímto toolboxem pro přeparametrizované ARMA modely (*Autoregressive Moving-Average*), tedy kódování v prostoru náhodných veličin. Pro porovnání bude také simulován kodér na bázi LPC kódování, na jehož základě pracuje většina dnešních řečových kodérů. Součástí kodérů jsou i dekodéry se zvukovými výstupy rekonstrukcí. Přílohou práce je i CD, kde jsou tyto kodéry naprogramovány v prostředí MATLAB.

1 VEKTOROVÝ PROSTOR

Vektorový prostor (VP) je prostor o konečně či nekonečně mnoha dimenzích, jehož prvky jsou vektory. VP s nekonečně mnoha dimenzemi pak obsahuje nekonečný počet lineárně nezávislých vektorů. Nad všemi vektory z VP se dají provádět různé operace, které jsou studiem lineární algebry, např. sčítání vektorů, násobení vektoru skalárem, skalární součin nebo vektorový součin. V rámci hledání řídkých řešení budeme nejčastěji využívat skalární součin, zapisujeme ve tvaru $\langle \mathbf{x}, \mathbf{y} \rangle$, který je v prostoru posloupností vyjádřen jako [2]

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_i x_i y_i \quad (1.1)$$

pro posloupnosti (vektory) \mathbf{x}, \mathbf{y} a v prostoru funkcí jako

$$\langle f, g \rangle = \int f(x)g(x)dx \quad (1.2)$$

pro funkce $f(x), g(x)$.

1.1 Generátory vektorového prostoru

Systém generátorů VP je podmnožina vektorů z onoho VP, která jej generuje. Generováním se rozumí to, že každý vektor z VP je lineární kombinací vektorů z této podmnožiny. Když je počet generátorů VP větší než je jeho dimenze, pak jeden stejný bod $x \in VP$ může mít více souřadnic. Mějme dva generující vektory $\mathbf{e} = (e_1, e_2, \dots, e_n)$ a $\mathbf{f} = (f_1, f_2, \dots, f_n)$, pak je možné bod x definovat pomocí obou generujících vektorů [2]:

$$x \in VP : x = c_1 e_1 + c_2 e_2 + \dots + c_n e_n, \quad (1.3)$$

$$x \in VP : x = d_1 f_1 + d_2 f_2 + \dots + d_n f_n, \quad (1.4)$$

kde c_i, d_i jsou skalární koeficienty udávající jeho souřadnice. Tím vzniká systém generátorů VP, jehož jednotlivé generátory se dají kombinovat a převádět je jako obrazy do jiných prostorů. K tomu se využívá skalární součin (1.1):

$$x = \sum_i c_i e_i = \sum_i d_i f_i = \sum_i \langle x, f_i \rangle e_i = \sum_i \langle x, e_i \rangle f_i. \quad (1.5)$$

1.2 Báze vektorového prostoru

Bázi VP se rozumí minimální systém jeho generátorů. Je to množina lineárně nezávislých vektorů, jejichž lineární obal je roven celému uvažovanému VP. V konečně dimenzionálním prostoru o dimenzi n je bázi každá množina obsahující n lineárně nezávislých vektorů. Je-li $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n]$ bázi n -dimenzionálního VP, pak každý jeho prvek $\mathbf{v} \in \text{VP}$ se dá vyjádřit pomocí souřadnic a_i v oné bázi jako [2]

$$\mathbf{v} = \sum_{i=1}^n a_i \mathbf{b}_i. \quad (1.6)$$

Pro modelování VP se nejčastěji používají báze **ortogonální** a **ortonormální**. Má-li jeden VP více bází, potom má každá jeho báze stejný počet generujících vektorů.

Ortogonalní báze

Ortogonalní báze je taková báze, ve které jsou libovolné dva různé vektory $\mathbf{b}_i, \mathbf{b}_k$ z báze $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n]$ na sebe kolmé, tedy platí:

$$\langle \mathbf{b}_i, \mathbf{b}_k \rangle = 0, \quad \langle \mathbf{b}_i, \mathbf{b}_i \rangle \neq 0. \quad (1.7)$$

Ortonormální báze

Ortonormální báze je taková báze, ve které pro libovolné dva různé vektory $\mathbf{b}_i, \mathbf{b}_k$ z báze $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n]$ platí (1.7) a navíc každý její vektor \mathbf{b}_i je jednotkový, tedy navíc platí [11]:

$$\langle \mathbf{b}_i, \mathbf{b}_i \rangle = 1. \quad (1.8)$$

Duální báze

Existují také **duální báze**, což jsou takové báze \mathbf{B} a \mathbf{B}^* stejných dimenzí, pro jejichž vektory platí [2]:

$$\langle \mathbf{b}^*_i, \mathbf{b}_j \rangle = \begin{cases} 1 & \text{pro } i = j, \\ 0 & \text{pro } i \neq j \end{cases}$$

pro všechna $i, j \in \{1, 2, \dots, n\}$. Hovoříme potom o duálním VP, kde mají všechny prvky pro obě báze stejné souřadnice (jsou vzájemně izomorfní). Bázi \mathbf{B}^* pak nazýváme duální bázi k bázi \mathbf{B} daného VP. Zapišeme-li bázi maticí, pak duální bázi bude tvořit matice k ní inverzní.

1.3 Příklady vektorových prostorů

Existuje několik základních VP, ve kterých se v lineární algebře operuje [2, 11]. Pro naše účely jsou nejdůležitější VP se skalárním součinem, u kterých je vždy definován standardní skalární součin a z něj pak určena **norma prvku** pomocí vztahu

$$\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}. \quad (1.9)$$

Vektorový prostor $\mathbb{R}^n, \mathbb{C}^n$:

Obecný n -rozměrný prostor reálných (komplexních) čísel – množina jejich uspořádaných n -tic. Standardní skalární součin prvků $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n, \mathbb{C}^n$ se zde počítá jako

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^n x_i y_i = \mathbf{x}^T \mathbf{y}. \quad (1.10)$$

Vektorový prostor $\mathbb{R}^{m,n}$:

Množina všech matic typu $m \times n$. Standardní skalární součin matic $\mathbf{A} = (a_{ij})$ a $\mathbf{B} = (b_{ij})$ v tomto VP je vyjádřen jako

$$\langle \mathbf{A}, \mathbf{B} \rangle = \sum_{i=1}^m \sum_{j=1}^n a_{ij} b_{ij}. \quad (1.11)$$

Vektorový prostor $f : \langle a, b \rangle \rightarrow \mathbb{R}$:

Množina všech spojitých funkcí na intervalu $\langle a, b \rangle$. Pro standardní skalární součin dvou funkcí $f(x), g(x)$ z tohoto VP platí:

$$\langle f, g \rangle = \int_a^b f(x)g(x)dx. \quad (1.12)$$

1.3.1 Hilbertův prostor

Za **Hilbertův prostor** se považuje takový VP, který je:

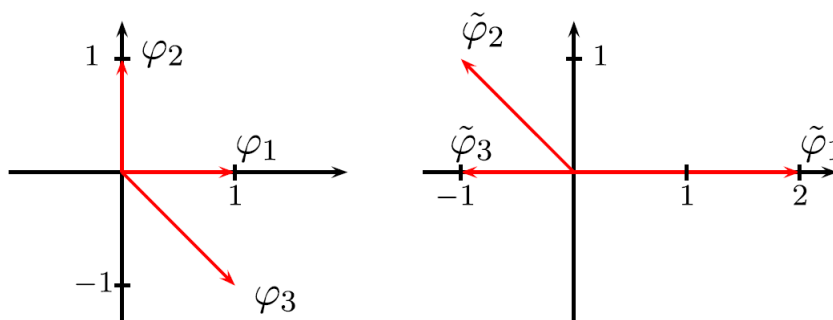
- **úplný** – každá Cauchyovská posloupnost má v tomto prostoru limitu, tzn. posloupnost „nevybíhá“ z prostoru ven,
- **separabilní** – obsahuje alespoň jednu spočetnou hustou podmnožinu (podprostor),
- **unitární** – je na něm definovaný skalární součin, který určuje metriku VP.

Jeho dimenze může být konečná i nekonečná. Výhodou je, že se dá pomocí vektorů ortogonálně promítat na spočetné množiny (podprostory). Nejjednodušším příkladem Hilbertova prostoru je $\ell^2(\mathbb{R})$, kde [11]:

$$\mathbf{x} = \{x_1, x_2, \dots\}; \quad x_i \in \mathbb{R}, \quad i \in \mathbb{N}, \quad \sum_{i=1}^{\infty} x_i^2 \leq \infty \quad (1.13)$$

1.4 Framy

Báze v konečně rozměrném VP určuje počet lineárně nezávislých vektorů, který je stejný jako dimenze VP. Když je počet vektorů v bázi větší než dimenze prostoru, můžeme stále reprezentovat jakýkoliv bod ve VP. Jednotlivé vektory však již nebudou lineárně nezávislé. Taková množina vektorů se pak nazývá **frame**. Framy jsou nástroj pro reprezentaci signálu, který je ve své podstatě nadbytečný a mluví se o **přeparametrizovaných modelech** signálu. Nadbytečnost však může být pro reprezentaci signálu výhodou. Framy jsou méně omezené než báze, a proto se používají pro jejich flexibilitu.



Obr. 1.1: Příklad jednoduchého framu a jeho duální frame

Pro jednoduchost nyní budeme uvažovat Hilbertův prostor s velikostí 2. Mějme frame $\Phi = [\varphi_1, \varphi_2, \varphi_3] = (1, 0), (0, 1), (1, -1)$. Na první pohled je patrné, že první dva vektory jsou ortonormální bázi VP \mathbb{R}^2 a třetí je jejich lineární kombinací: $\varphi_3 = \varphi_1 - \varphi_2$. Je také zřejmé, že těmito třemi vektory lze vyjádřit jakýkoliv vektor z daného VP. To samé lze ale i říct o vektorech φ_1 a φ_2 , což znamená že vektor φ_3 může být jakýkoliv vektor z VP. Můžeme tedy vyjádřit vektor \mathbf{x} jako [9]:

$$\mathbf{x} = \langle \mathbf{x}, \varphi_1 \rangle \varphi_1 + \langle \mathbf{x}, \varphi_2 \rangle \varphi_2. \quad (1.14)$$

K tomuto vztahu (1.14) můžeme přičíst nulu, kterou budeme reprezentovat jako rozdíl vektorů φ_1 a φ_2 .

$$\mathbf{x} = \langle \mathbf{x}, \varphi_1 \rangle \varphi_1 + \langle \mathbf{x}, \varphi_2 \rangle \varphi_2 + (\langle \mathbf{x}, \varphi_1 \rangle - \langle \mathbf{x}, \varphi_2 \rangle)(\varphi_1 - \varphi_2) \quad (1.15)$$

Výraz (1.15) pak můžeme přepsat jako:

$$\mathbf{x} = \langle \mathbf{x}, 2\varphi_1 \rangle \varphi_1 + \langle \mathbf{x}, (-\varphi_1 + \varphi_2) \rangle \varphi_2 + (\langle \mathbf{x}, -\varphi_1 \rangle)(\varphi_1 - \varphi_2). \quad (1.16)$$

Ve výrazu (1.16) můžeme rozpoznat $(\varphi_1 + \varphi_2)$ jako $-\varphi_3$ a vektory uvnitř skalárních součinů pak označíme:

$$\tilde{\varphi}_1 = 2\varphi_1, \quad \tilde{\varphi}_2 = -\varphi_1 + \varphi_2, \quad \tilde{\varphi}_3 = -\varphi_1 \quad (1.17)$$

Pomocí tohoto značení můžeme přepsat výraz (1.16) jako:

$$\mathbf{x} = \langle \mathbf{x}, \tilde{\varphi}_1 \rangle \varphi_1 + \langle \mathbf{x}, \tilde{\varphi}_2 \rangle \varphi_2 + \langle \mathbf{x}, \tilde{\varphi}_3 \rangle \varphi_3 = \sum_{i=1}^3 \langle \mathbf{x}, \tilde{\varphi}_i \rangle \varphi_i. \quad (1.18)$$

Maticově lze pak zapsat framy jako:

$$\Phi = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \end{bmatrix}, \quad \tilde{\Phi} = \begin{bmatrix} 2 & -1 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad (1.19)$$

a lze tedy napsat:

$$\mathbf{x} = \sum_{i=1}^3 \langle \mathbf{x}, \tilde{\varphi}_i \rangle \varphi_i = \Phi \tilde{\Phi} \mathbf{x}. \quad (1.20)$$

Tímto způsobem jsme vytvořili tzv. **duální frame** $\tilde{\Phi}$ k framu Φ [9].

2 PŘEPARAMETRIZOVANÉ SYSTÉMY LINEÁRNÍCH ROVNIC A HLEDÁNÍ ŘÍDKÉHO ŘEŠENÍ

Přeparametrizované systémy lineárních rovnic jsou takové soustavy lineárních rovnic, ve kterých figuruje více proměnných, než je potřeba. Mají zpravidla nekonečně mnoho řešení a používají se k deklaraci parametrů, ze kterých je následně vybíráno tzv. řídké řešení.

Hledání řídkého řešení, neboli výběr nejdůležitějších parametrů (např. při použití framů), je poměrně obecný pojem, který nabádá k pestré paletě matematických postupů jak ho dosáhnout. K tomu se využívá vektorových prostorů, do kterých se problematika transformuje. Omezme se na to, že VP je generován maticí reálných prvků. Pro tento postup se obvykle používá pseudoinverze [15].

Při hledání řešení rozlišujeme dva typy prostorů o konečném počtu dimenzí:

- **vektorový** – pracuje se s vektory, které představují n -tice čísel,
- **funkcionální** – pracuje se se spojitými funkcemi v intervalu $\langle a, b \rangle$.

2.1 Pseudoinverze

Pseudoinverze je důležitým nástrojem používaným při hledání řídkých řešení. Cílem je získat řešení s minimální energií chyby. Mějme případ soustavy lineárních rovnic, kdy je pseudoinverzí řešena soustava

$$\mathbf{T}\boldsymbol{\xi} = \boldsymbol{x}, \quad (2.1)$$

kde \boldsymbol{x} je zkoumaný vektor ve VP $X := \mathbb{R}^n$; $\boldsymbol{\xi}$ je vektor parametrů z prostoru \mathbb{R}^m ; $\mathbf{T} = (t_{ij})$ je matice generátorů prostoru X o rozměru $n \times m$. Tato matice je operátorem zobrazení $\mathbb{R}^m \rightarrow \mathbb{R}^n$. Obecnou podmínkou pro řešení pseudoinverze je, že všechny obrazy patří do konečné množiny, neboli úplného VP. Mohou nastat tři případy:

1. $n = m$: pak se jedná o klasickou inverzi matice za podmínky, že je singulární,
2. $n > m$: pak řešení existuje jen tehdy, jsou-li zkoumané vektory ve VP X lineární kombinací všech jeho generátorů,
3. $n < m$: existuje zpravidla nekonečně mnoho řešení (přeparametrizováno).

Nás zajímá případ 3, kdy se budou počítat tzv. normální rovnice pro hledání řešení s nejnižší energií.

Normální rovnice

Normální rovnice jsou takové rovnice, jejichž řešením se snažíme dostat minimální rozdíl pravé a levé strany. Normální se jmenují proto, že se zde pracuje s normami prvků pro účel minimalizace chyby. Jsou analogií hledání minimální chyby predikce u klasických predikčních modelů. V našem případě je třeba řešit rovnici $\mathbf{T}\boldsymbol{\xi} = \mathbf{x}$ s neznámým vektorem $\boldsymbol{\xi}$. Je zde splněna podmínka, že \mathbf{T} je matice $n \times m$, kde $n \leq m$, a tak je možno řešit rovnici jako normální. Snahou je získat zde vektor $\boldsymbol{\xi}$, který obsahuje co nejmenší počet významných (řídkých) parametrů. Obecně nelze takovou rovnici vyřešit naprosto přesně, je tedy stanovena numerická přesnost ε , se kterou je počítáno. Řešená rovnice vypadá následovně:

$$\|\mathbf{x} - \mathbf{T}\boldsymbol{\xi}\| < \varepsilon/2. \quad (2.2)$$

Pro její řešení je použita metoda nejmenších čtverců – LSQ (*Least Squares*), která zajistí, aby byla tato podmínka splněna. Rovnice (2.2) se dá přepsat na

$$\mathbf{T}^T \mathbf{T} \boldsymbol{\xi} = \mathbf{T}^T \mathbf{x}, \quad (2.3)$$

což je ona **normální rovnice**. Jejím řešením je $\boldsymbol{\xi} = (\mathbf{T}^T \mathbf{T})^{-1} \mathbf{T}^T \mathbf{x}$. Existuje-li inverzní matice k normální matici $\mathbf{T}^T \mathbf{T}$, pak je jejím řešením přímo výsledek výpočtu metodou LSQ. Normální rovnice slouží k nalezení duálního framu prostoru, ve kterém má soustava řídké řešení. Bude aplikována k vyřešení rovnice pseudoinverze (2.4). Využívá je např. algoritmus BPA (*Basis Pursuit Algorithm*), který je popsán v kapitole 2.2. [15]

Při výpočtu pseudoinverze se dále využívá ortogonální projekce, kdy všechny body x_i jsou promítuty do lineárního podprostoru $H \subseteq X$. Tento ortogonální (kolmý) průmět označme jako $\hat{\mathbf{x}} = \mathbf{P}\mathbf{x}$, kde \mathbf{P} je matice ortogonální projekce. Cílový podprostor o dimenzi n je generován vektory $\Phi = \{\phi_j\}_{j \in J}$, kde $J = \{1, 2, \dots, m\}$ [15].

Je třeba řešit rovnici:

$$\hat{\mathbf{x}} = \mathbf{T}\boldsymbol{\xi}_j = \sum_{j \in J} \boldsymbol{\xi}_j \phi_j, \quad (2.4)$$

kterou když pro přehled rozepíšeme, tak vypadá následovně:

$$\hat{\mathbf{x}} = [\phi_1, \dots, \phi_m] \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_m \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & \dots & t_{1m} \\ t_{21} & t_{22} & \dots & t_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ t_{n1} & t_{n2} & \dots & t_{nm} \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_m \end{bmatrix},$$

kde $\phi_j = \mathbf{T}\varepsilon_j$ je j -tý sloupec generující matice \mathbf{T} . Vektor proměnných ξ_j obsahuje více parametrů než je potřeba. Cílem celého procesu je získat ξ'_j takový, který je řídký a ukazuje jen na důležité generátory z matice \mathbf{T} .

Postup řešení je následující:

Pseudoinverzí se snažíme získat matici $\mathbf{T}^* : X \rightarrow \ell^n(J)$ z původní \mathbf{T} , aby platilo:

$$\mathbf{T}^* \mathbf{x} = \langle \mathbf{x}, \phi_j \rangle_{j \in J}, \quad (2.5)$$

což ve své podstatě představuje vážený průměr analogický s funkcionálním modelem, kde se vyjádří jako $\int_J x(t)\phi_i(t)dt$. Pro reprezentaci \mathbf{x} je potřeba získat hodnoty ξ_i . Naším cílem teď bude vybrat taková ξ_i , u kterých bude zjištěna minimální energie. Využijeme k tomu ortogonální projekce, kdy platí [15]:

$$\mathbf{P} : X \rightarrow H, \quad \mathbf{P}(\mathbf{P}\mathbf{x}) = \mathbf{P}\mathbf{x}, \quad (2.6)$$

$$\mathbf{T}^* \underbrace{(\hat{\mathbf{x}} + \mathbf{x}^\perp)}_{\mathbf{x}} = \mathbf{T}^* \hat{\mathbf{x}} + \underbrace{\mathbf{T}^* \mathbf{x}^\perp}_0. \quad (2.7)$$

Jak je vidět, tak kolmá složka \mathbf{x} se anuluje a tím pádem platí $\mathbf{T}^* \hat{\mathbf{x}} = \mathbf{T}^* \mathbf{x}$. To usnadní další výpočty. V další fázi se obvykle používá Moore-Penroseova pseudoinverze, která vyšetřuje ξ^+ pomocí minimální Euklidovské normy [15]:

$$\|\xi^+\| = \operatorname{argmin}_{\hat{\mathbf{x}}=\mathbf{T}\xi} \|\xi\|. \quad (2.8)$$

Je zavedena nová matice \mathbf{T}^+ , pro kterou ve vztahu k ξ^+ a \mathbf{x} platí:

$$\mathbf{T}^+ := \mathbf{T}^{-1} \mathbf{x} =: \xi^+ \quad (2.9)$$

$$\xi := \xi^+ = \mathbf{T}^+ \mathbf{x} = \mathbf{R}^+ \mathbf{T}^* \mathbf{x} = \mathbf{R}^+ \xi', \quad (2.10)$$

kde \mathbf{R}^+ je korelační matice

$$\mathbf{R} = \mathbf{T}^* \mathbf{T} = (\mathbf{R}_{kj}), \quad \mathbf{R}_{kj} = \phi_k^* \phi_j = \langle \phi_j, \phi_k \rangle \quad (2.11)$$

a ξ' je množina diskretizovaných hodnot \mathbf{x} :

$$\xi' := \mathbf{T}^* \mathbf{x} = [\phi_1^* \mathbf{x}, \dots, \phi_m^* \mathbf{x}]^T = [\langle \mathbf{x}, \phi_1 \rangle, \dots, \langle \mathbf{x}, \phi_m \rangle]^T. \quad (2.12)$$

Dále se použije vztah

$$\hat{\mathbf{x}} = \mathbf{T} \xi^+ = \mathbf{T} \mathbf{R}^+ \mathbf{T}^* \mathbf{x} = \mathbf{T}' \xi', \quad (2.13)$$

ve kterém platí:

$$\mathbf{T}' := [\phi'_1, \dots, \phi'_m] := \mathbf{T} \mathbf{R}^+, \quad (2.14)$$

kde j -tý sloupec matice \mathbf{R}^+ obsahuje hledané parametry ξ'_j pro generátor ϕ'_j , aby bylo možno odhadnout $\hat{\mathbf{x}}$ dle vztahu (2.4).

Pseudoinverze nemá tendenci sama o sobě dávat řídká řešení, protože počítá s velkým množstvím čísel a jejich zaokrouhlováním vzniká chyba. Proto bývá ve vlastních procedurách pro hledání řídkého řešení doplňována dalšími algoritmy [15].

2.2 Algoritmy hledání řídkého řešení, BPA

Aby bylo nalezeno požadované řídké řešení, je třeba VP testovat. Nejdříve se testuje každý vektor zvlášť, pak všechny dvojice, trojice až nakonec všechny naráz. Jako neoptimálnější řešení se jeví to s nejmenším počtem vektorů a s co nejmenší odchylkou (aplikace Moore-Penroseovy pseudoinverze) [15].

Bylo již navrženo mnoho algoritmů od různých tvůrců pro hledání řídkého řešení z přeparametrizovaného. Nás bude nejvíce zajímat ten algoritmus, který bude použit při kódování řeči v rámci této práce. Jedná se o vícestavovou iterativní proceduru kódovanou v programu MATLAB, která se ukázala být odolná vůči šíření výpočetních chyb při řešení i špatně nastavených úloh. Procedura je založena na algoritmu **BPA** (*Basis Pursuit Algorithm*) a navržena pro řešení vektorů konečných dimenzí, následně byla rozšířena i na řešení funkcionálních úloh [8, 15].

Algoritmus BPA má následující iterační kroky:

(krok 0) Je nastaven počáteční odhad $\xi^{(0)} = \mathbf{T}^+ \hat{\mathbf{x}}$.

(krok 1) V tomto kroku je zavedeno kritérium optimality $\|\mathbf{x} - \hat{\mathbf{x}}\| \rightarrow \min$ pracující s numerickou přesností $\|\mathbf{x} - \mathbf{T}\xi\| < \varepsilon/2$. Vzniká tak ε -suboptimální řešení $\xi^{(1)}$, které zpravidla ještě není řídké.

(krok 2) V dalším kroku se hledá řešení $\xi^{(2)} = \operatorname{argmin}_{\xi \in \ell^2(J)} \|\xi\|_{w,1}$ s nejnižší energií chyby, které již je považováno za blízké řídkému řešení.

(krok 3) V předposledním kroku je již nalezeno řídké ε -suboptimální řešení $\xi^* := \{\xi_j^{(2)}\}_{j \in F^*}$ zvolením prahové hodnoty $\delta > 0$ pro jeho výpočet.

(krok 4) Spuštění celého procesu znovu od (kroku 1) s tím, že novým počátečním odhadem bude $\xi^{(0)} = \xi^*$ získané v této iteraci. Do další iterace je množina J nahrazena redukovanou množinou F^* , což představuje zúžení úlohy do menšího podprostoru.

Dalšími používanými algoritmy jsou např. IRLS (*Iteratively Reweighted Least Squares*) nebo OMP (*Orthogonal Matching Pursuit*) [3].

3 PŘEPARAMETRIZOVANÉ MODELY SIGNÁLŮ

V 90. letech 20. století započal výzkum spočívající ve zdokonalení stávajících přístupů k modelování signálů. Ty dnes fungují tak, že se u signálu matematicky analyzuje jeho chování a výstupem je sada koeficientů, které ho v modelu charakterizují. Počet koeficientů je předem pevně stanoven v závislosti na řádu modelu. Nevýhodou těchto koeficientů je, že je třeba je udávat velmi přesně a všechny mají relativně stejnou váhu (aby bylo dosaženo požadované přesnosti). Přeparametrizované modely vytvoří při analýze signálu takových koeficientů nekonečně mnoho, protože je zde operováno s maticemi, které mají více proměnných než řádků. Z těchto koeficientů jsou následně vybírány jen ty nejdůležitější a jsou jim přiřazeny váhy. Celému modelu je nastavena přesnost s jakou má pracovat a na základě toho jsou koeficienty vybírány, ty ostatní (nevýrazné) zůstávají nepoužity – najde se řídká reprezentace signálu [3, 15].

3.1 Modelování signálů pomocí atomů

Signál je možno pomocí přeparametrizovaných modelů modelovat tak, že ho rozdělíme na vhodně dlouhé segmenty a pro ty pak vygenerujeme banku atomů. Atomy jsou elementární signály (vektory), pomocí kterých se pak bude odhadovaný signál modelovat, konkrétně součtem vhodně váhovaných vybraných atomů. Velikost banky atomů, ze kterých se vybírá, je závislá na typu signálu, nejčastěji jich bývá v řádu stovek, pro složitější a především pro obrazové signály jich pak bývá tisíce až desetitisíce. Nasazením algoritmů hledání řídkého řešení, např. již zmiňovaného BPA, se z těchto mnoha atomů vyberou ty nejdůležitější, které budou signál reprezentovat [8, 15].

Tento model nám již přiblíží konkrétní významy pro pojmy z lineární algebry:

\mathbf{x} ... zkoumaný signál délky n

$\hat{\mathbf{x}}$... odhadovaný signál (aproximace \mathbf{x})

ϕ_j ... jednotlivé atomy délky n [ϕ_1, \dots, ϕ_m] generující VP

ξ_j ... vektor vah (ξ_1, \dots, ξ_m)

\mathbf{T} ... matice atomů rozměru $n \times m$ (tvoří bázi VP)

Odhadovaný signál se dá charakterizovat rovnicí

$$\hat{\mathbf{x}} = \mathbf{T}\boldsymbol{\xi}_j = \sum_{j \in J} \boldsymbol{\xi}_j \phi_j = \boldsymbol{\xi}_1 \phi_1 + \dots + \boldsymbol{\xi}_m \phi_m. \quad (3.1)$$

Matice \mathbf{T} se také nazývá **slovník**. Jeho „slova“ jsou atomy, z nichž se snažíme vybrat jen ty nejdůležitější, na které odkazuje řídce reprezentované $\boldsymbol{\xi}_j$. To je řešením rovnice (2.1), kdy stanovíme neznámý vektor $\boldsymbol{\xi}$ o m proměnných ($m > n$), aby byl model přeparametrizován [15].

Algoritmy hledání těch správných atomů jsou nejčastěji založeny na hledání takové báze vygenerovaného VP, ve které mají atomy nejnižší energii. Výsledkem bývá zpravidla výběr jednotek atomů – záleží na velikosti banky atomů, v tomto případě čím je větší, tím pro nás lépe – větší šance že tam bude ten nejpodobnější signálu.

3.2 Modelování pomocí časových řad

Teorie o časových řadách je používána pro modely s velkým počtem parametrů. Pro modelování je použit modifikovaný algoritmus BPA. Pro jednoduchost se omezme na ARMA modely pro časově invariantní stacionární časové řady. Po provedení a analýze mnoha numerických simulací byla zjištěna následující stanoviska [16]:

(1) Časové řady nejsou spolehlivě schopny nalézt skoro žádné takové parametry modelu, které by mohly redukovat původně špatně navržený přeparametrizovaný model. Narozdíl od ostatních modelů zde není potřeba brát ohled na řády, což je u standardních postupů prvotním krokem. Pro krátké řady (do 100 vzorků) podávají řídké parametry přesnější predikci v porovnání s těmi, založenými na standardní maximální pravděpodobnosti z modelu *frames*. Pro delší řady (500 a více vzorků) dávají obě techniky téměř stejné výsledky.

(2) Když model obvykle závisí na odhadovaných parametrech, jejich přesnost se dá zvětšit opakováním aplikace algoritmu BPA.

Vektory představují prostor o konečně mnoha dimenzích. Funkcionální přístup umožňuje podstatně přesnější reprezentaci prvků v prostoru, narozdíl od diskrétního vektorového přístupu. Celá tato problematika je transformována do Hilbertova prostoru $L^2 := L^2(\Omega, \mathcal{A}, \mathcal{P})$ s náhodnými proměnnými definovanými v pravděpodobnostním prostoru $(\Omega, \mathcal{A}, \mathcal{P})$; Ω je jevové pole (množina všech náhodných proměnných definovaných v prostoru), \mathcal{A} je systém všech podmnožin jevového pole Ω a \mathcal{P} pravděpodobnost jevů definovaných na \mathcal{A} [2]. Numerická studie pak ukazuje výhody a limity tohoto přístupu při aplikaci přeparametrizovaných ARMA modelů pro nestacionární časové řady [16].

3.2.1 Klasické ARMA modely

Většina současných modelů pro modelování časových řad vychází ze statistiky a lineární predikce, tzn. vzorky následující jsou odhadnuty z charakteru vývoje vzorků předšlých. Lineární predikce vychází z tzv. **ARMA modelů** (*Autoregressive Moving-Average*). Tyto modely se vyznačují velkou přesností predikce, ale je poměrně obtížné najít jejich parametry. ARMA vznikl sloučením dvou dílčích modelů MA a AR, které vypadají následovně:

MA ... model klouzavých průměrů (*Moving-Average*). Je to v podstatě lineární kombinace hodnot bílého šumu, který reprezentuje náhodnou chybu. Matematický zápis modelu MA řádu q vypadá takto:

$$\text{MA}(q)_t = b_0 - b_1 e_{t-1} - b_2 e_{t-2} - \dots - b_q e_{t-q} + e_t, \quad (3.2)$$

kde t je čas v jistém okamžiku, b_0, b_1, \dots, b_q jsou parametry modelu, $e_{t-1}, \dots, e_{t-q}, e_t$ je bílý šum – náhodná chyba, která do systému vstupuje. Model $\text{MA}(q)_t$ se dá reprezentovat také číslicovým filtrem typu FIR řádu q .

AR ... autoregresní model (*Autoregressive*). Lze popsat jako lineární kombinace vlivů minulých hodnot. Matematický zápis modelu AR řádu p vypadá takto:

$$\text{AR}(p)_t = a_0 - a_1 y_{t-1} - a_2 y_{t-2} - \dots - a_p y_{t-p} + e_t, \quad (3.3)$$

kde a_0, a_1, \dots, a_p jsou autoregresní koeficienty a $y_{t-1}, y_{t-2}, \dots, y_{t-p}$ jsou předcházející hodnoty časové řady, e_t je opět náhodná chyba. Číslicový filtr typu $\text{AR}(p)_t$ je IIR filtr řádu p .

ARMA modely jsou využívány nejčastěji ve statistických aplikacích, např. predikce budoucího vývoje trhu v ekonomice, předpovědi počasí či predikce řeči, kdy je použit jen model AR a mluví se o LPC kódování řeči. Pro určení parametrů takových modelů je důležitá velikost výkonu chyby predikce, od které se odvíjí určení vhodného řádu ARMA modelu [2].

3.2.2 Přeparametrizované ARMA modely

Pro modelování časových řad pro přeparametrizované modely použijeme přeparametrizovaného ARMA modelu, který umožňuje vyjmout nadbytečné parametry. Princip jeho návrhu je následující [16, 17]:

Nazvěme H_t separabilním uzavřeným prostorem v L^2 , rozděleným historií bodů X_t v čase t a zavedme ortogonální operátor projekce $P_t : L^2 \rightarrow H_t$. Ortogonalizací X_t získáme prostor $H_t = \{Z_{t+1-j}\}_{j=1}^{\infty}$, kde Z_t je nekorelováno, definováno jako $Z_t =$

$X_t - P_{t-1}X_t$; $t \in \mathbb{Z}$. Omezme se, že X_t je **stacionární systém** typu ARMA(p, q) a Z_t bílý šum, značíme WN($0, \sigma^2$). Tedy X_{t+1-j} i Z_{t+1-j} jsou framy v prostoru H_t . Rozdělením obou framů dostaneme nový přeparametrizovaný frame $\{U_{t+1-j}\}_{j=1}^{\infty} = \{X_{t+1-j}\}_{j=1}^{\infty} \cup \{Z_{t+1-j}\}_{j=1}^{\infty}$ v H_t . Stanovením P, Q jako $0 \leq p \leq P \leq \infty$, resp. $0 \leq q \leq Q \leq \infty$, dostaneme sice přeparametrizovaný, ale stále atomický rozklad odhadu \hat{X}_t [16]:

$$\hat{X}_{t+1} = P_t X_{t+1} = \underbrace{\sum_{j=1}^P \phi_j X_{t+1-j}}_{\text{AR}(P)_t} + \underbrace{\sum_{k=1}^Q \Theta_k Z_{t+1-j}}_{\text{MA}(Q)_t} =: T_t^{P,Q} \xi =: \underbrace{\sum_{i=1}^{P+Q} U_{t+1-i} \xi_i}_{\text{ARMA}(P,Q)_t}, \quad (3.4)$$

kde atomy představují $U_{t+1-j} =: X_{t+1-j}$ pro $j = 1, \dots, P$ a $U_{t+1-P-k} =: Z_{t+1-k}$ pro $k = 1, \dots, Q$, kde $\xi := \{\Phi, \Theta\}$ symbolizuje odpovídající sloučení posloupností koeficientů $\Phi := \{\Phi_j\}_{j=1}^P$ a $\Theta := \{\Theta_j\}_{k=1}^Q$. Dále platí, že $\mathbf{T}_t := \mathbf{T}_t^{P,Q} : \ell^2(J) \rightarrow H_t$, $J := \{1, \dots, P+Q\}$ je operátor konečné dimenze.

Přeparametrizovaný model ARMA(P, Q) reprezentuje odhad řady $\hat{X}_{t+1} = \mathbf{T}_t^{P,Q} \xi$ s konečným, nicméně dostatečně nadhodnocenými řády $P, Q : p \leq P < \infty, q \leq Q < \infty$. Tento zápis je analogií reprezentace odhadu signálu (3.1), ve slovníku jsou zde náhodné procesy. Ve většině případů je plně dostačující volba $P = Q = 10$. Pro výpočty je zaveden další operátor: \mathbf{T}_t^* . Řadu bude zobrazovat z prostoru H_t do prostoru $H'_t \subseteq \ell^2(J)$ o stejné dimenzi. Jeho aplikací do obou stran rovnice (3.4) se její podoba změní na systém o $M := P+Q$ lineárních rovnicích:

$$b_t = R_t \xi, \quad \text{aneb} \quad b_i(t) = \sum_{j=1}^M R_{ij}(t) \xi_j \quad \text{pro} \quad i \in J, \quad (3.5)$$

kde $b_i(t) := \langle X_{t+1}, U_{t+1-i} \rangle$ symbolizuje sdružený moment 2. řádu řady X_{t+1} a i -tého atomu U_{t+1-i} a $R_{ij}(t) := \langle U_{t+1-j}, U_{t+1-i} \rangle$ symbolizuje sdružený moment 2. řádu j -tého a i -tého atomu, který je jejich kovariancí vzhledem k střední nulové hodnotě. Je-li ARMA(p, q) systémem s nulovou střední hodnotou a také kauzální s autokovarianční funkcí $\gamma(h) := \text{cov}(X_{t+h}, X_t) = \langle X_{t+h}, X_t \rangle$, pak můžeme rovnici (3.5) vyjádřit jako [16]

$$\mathbf{b} = \mathbf{R} \xi, \quad (3.6)$$

$$\text{kde} \quad \mathbf{b} = \begin{bmatrix} \gamma_P \\ \sigma^2 \psi_Q \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \mathbf{\Gamma}_P & \sigma^2 \mathbf{\Psi}^* \\ \sigma^2 \mathbf{\Psi} & \sigma^2 \mathbf{I}_Q \end{bmatrix} \quad \text{a} \quad \xi = \begin{bmatrix} \Phi_P \\ \Theta_Q \end{bmatrix},$$

kde jednotlivé symboly zastupují prvky skupiny prvků ze vztahu (3.4):

$$\gamma_P := [\gamma(1), \dots, \gamma(P)]^T \text{ jsou autokovarianční koeficienty modelu AR}(P),$$

$$\psi_Q := [\psi(1), \dots, \psi(Q)]^T \text{ jsou šumové koeficienty modelu MA}(Q),$$

$$\sigma^2 := \gamma(0) / \sum_{k=0}^{\infty} |\psi_k|^2 = \gamma(0) / \|\psi\|^2,$$

$$\begin{aligned} \Gamma_P &:= [\gamma(i-j)]_{i,j=1}^P \dots \text{Toeplitzova matice } P \times P, \\ \Psi &:= [\psi(i-j)]_{i,j=1}^{Q,P} \dots \text{Toeplitzova matice } Q \times P, \\ \mathbf{I}_Q &\text{ je jednotková matice řádu } Q, \\ \Phi_P &:= [\Phi_1, \dots, \Phi_P]^T \dots \text{neznámé koeficienty typu AR}(P), \\ \Theta_Q &:= [\Theta_1, \dots, \Theta_Q]^T \dots \text{neznámé koeficienty typu MA}(Q). \end{aligned}$$

Hodnoty γ_P a ψ_Q jsou počítány skrze **Inovační algoritmus** (IA) [19]. Jedná se o operaci nad vzorky autokorelace (autokovariance), jejímž výsledkem je vektor, obsahující jen její významné vzorky. Hlavním cílem je získání kauzální reprezentace modelu [1]. IA je také součástí toolboxu *frames*.

Φ_P a Θ_Q jsou neznámými pro vícestupňový algoritmus využívající modifikovaný algoritmus BPA-4, upravený pro přeparametrizovaný ARMA(p,q) model – BPA-4 zde funguje stejně, jen použije jiné proměnné. Algoritmus již najde řídká řešení pro časové řady, tzn. vyřeší rovnici (3.5). Tento model bude použit při modelování řeči pomocí náhodných procesů [16, 17].

3.3 Porovnání s Fourierovou a waveletovou transformací

Pro lepší představu o přeparametrizovaných modelech signálu si ještě uvedme porovnání s dnes nejpoužívanějšími transformacemi – pokud jde o signály: Fourierovou a waveletovou [10].

Fourierova transformace (FT):

Spočívá v rozložení signálu na jednotlivé harmonické složky. „Atomy“ jsou zde jen harmonické funkce, tj. $A \cos(\omega t + \varphi_0)$, u které lze měnit frekvenci, amplitudu – „váhu“ a počáteční fázi. Transformace má tu výhodu, že je výpočetně nenáročná, velmi rozšířená a jednoduchá pro aplikaci. Omezení zde spočívá v tom, že můžeme modelovat jen z harmonických složek, které se v celém uvažovaném intervalu chovají homogenně, tedy když vynecháme jednu harmonickou složku, tak tím dojde ke zneřádnění signálu v celém intervalu. Přesnost transformace se odvíjí od počtu použitých harmonických složek.

Waveletová transformace (WT):

Tato transformace již není omezena jen na harmonický signál, ale zkoumá podobnost signálu s jednou vhodně zvolenou vlnkou (wavelet) – „atom“. Vlnka může mít nespočetně podob, je zde pouze podmínka její konvergence na obou stranách intervalu, tzn. musí mít konečnou energii. Transformace probíhá v několika stupních, kdy

je vlnka postupně zužována a hledá se podíl zastoupení detailů (vyšších frekvencí) v signálu. Tato transformace již dává znatelně lepší výsledky než FT, protože zde došlo k redukci počtu koeficientů, pochopitelně za cenu větší výpočetní náročnosti. O kvalitě modelování rozhoduje hloubka dekompozice – počet stupňů ve kterých se analyzuje.

Přeparametrizované modely:

Jedná se o nejnovější a pochopitelně o výpočetně nejnáročnější přístup. Signál je zde porovnáván s celou bankou atomů (slovník), nejčastěji se 128, 256 a 512 atomy – dle charakteru signálu [3]. Algoritmy pro vyhledání a váhování těch správných atomů jsou nastaveny tak, že z nich vzejde jen velmi málo atomů, které po vhodném váhování dají dohromady co nejpodobnější odhad signálu při použití co nejmenšího počtu koeficientů. Bude-li nějaký málo váhovaný atom vynechán, potom se nepřesnost projeví jen v tom místě signálu, kde ho bylo potřeba použít a ne v celém intervalu. Oproti předchozím dvou uvedeným transformacím je zde ještě výhoda adaptivní báze prostoru, ve kterém se pracuje. FT i WT mají tuto bázi pevnou.

4 TOOLBOX FRAMES V PROSTŘEDÍ MATLAB

Kompletní úlohu pro realizaci přeparametrizovaných modelů a hledání řídkého řešení realizuje toolbox zvaný **frames** pro MATLAB, který na naši fakultu dodal doc. RNDr. Vítězslav Veselý, CSc. z Ekonomicko-správní fakulty Masarykovy univerzity v Brně. Tento toolbox se sestává ze spousty zdrojových kódů – M-filů, popř. zkompilovaných P-filů, jenž každý má svůj dílčí úkol při hledání řešení.

4.1 Atomy

Základem pro modelování jsou atomy. Toolbox nabízí několik mateřských funkcí, ze kterých se jednotlivé atomy vytvářejí. Tyto mateřské funkce najdeme ve složce „/Frames“ ve složce „atoms“. Každou základní funkci reprezentuje jeden M-file. Funkce generující atomy mohou být jakékoliv funkce převážně s konečnou energií.

4.1.1 Generování atomů

Jak je již zmíněno výše, jednotlivé atomy jsou generovány z mateřských funkcí (M-filů). Pro variabilitu je nutné, aby v mateřských funkcích byly nadefinovány parametry, které ovlivňují tvar funkce a tedy i její podobu a vytváří tak samotné atomy. Mateřské funkce jsou definovány ve tvaru `funkce(t,p)`, kde `t` udává matici o velikosti $m \times n$, která definuje definiční obor, ve kterém budou jednotlivé funkční hodnoty funkce propočteny. Parametr `p` pak definuje parametry funkce, které jsou definovány přímo ve funkci. Jsou to např. posunutí, změna časového měřítko, modulace atd.

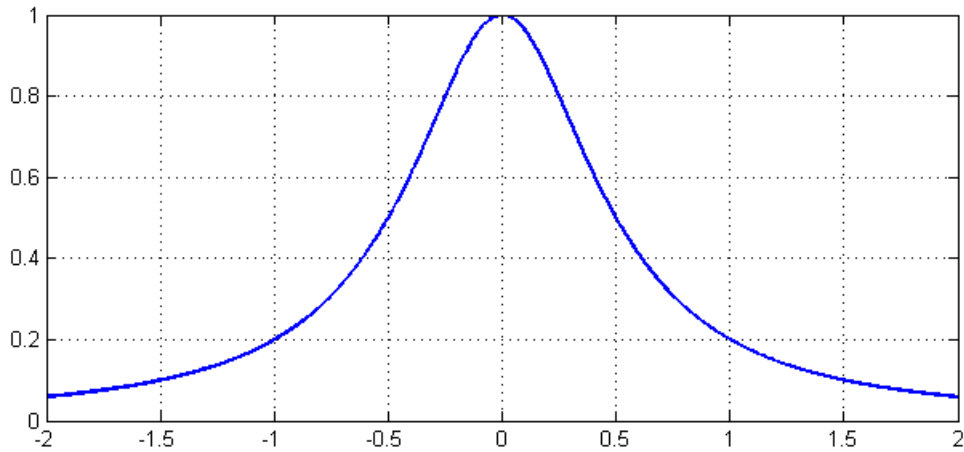
Volání funkce může probíhat několika způsoby:

- **Vygenerování jednoho atomu** – realizuje se zavoláním samotné mateřské funkce, není ani potřeba uvádět její parametry.
- **Vygenerování několika atomů z jedné mateřské funkce** – parametry jsou vkládány do matice, počet atomů je počet sloupců v této matici.
- **Vytváření atomů pomocí předdefinovaných funkcí toolboxu** (např. `frsample`) – lze vygenerovat obrovské množství atomů z různých mateřských funkcí jedním příkazem. Vždy se zadá název mateřské funkce, vektory parametrů a funkce již automaticky vygeneruje požadované atomy.

4.1.2 Lorentzova funkce

Často používanou mateřskou funkcí je **Lorentzova funkce**, která je také součástí toolboxu. Je popsána rovnicí (4.1) a její průběh v základním tvaru je na obrázku 4.1.

$$L(t) = \frac{1}{1 + 4t^2}. \quad (4.1)$$



Obr. 4.1: Lorentzova funkce v základním tvaru

Je zde umožněno nastavit tři parametry a modifikovat funkci do tvaru (4.2):

- a = posuv ve směru osy x ,
- b = změna měřítka osy x (modulace šířky),
- c = modulace harmonickým signálem.

$$L(t) \rightarrow L\left(\frac{t-a}{b}\right) \cos 2\pi ct. \quad (4.2)$$

Pomocí parametrů je tak možné funkci různě upravovat dle potřeby, tím se vytváří banka atomů. Do dalších výpočtů se pak posílá název mateřské funkce (v tomto případě `loren`) společně s maticí parametrů a, b, c .

Dalšími příklady mateřských funkcí jsou **pravoúhlý impulz**, **Gaussova funkce** či **Gaborova funkce**.

4.2 Základní funkce a možnosti toolboxu

S atomy je dále operováno v řadě funkcí, které se podílí na hledání řídkého řešení. Například je zprvu potřeba určení podoby Hilbertova prostoru, tzn. zjistit, jak vypadá norma a skalární součin. Na tyto úkoly existují samostatné funkce jako součást toolboxu, konkrétně `ipstd` a `frnorm`. Zajistí také automatické rozpoznání funkcionálního a vektorového (diskrétního) módu. U vektorového módu bude prováděn klasický skalární součin – $\mathbf{T}^* \mathbf{x} = \langle \mathbf{x}, \phi_i \rangle$, popř. komplexně sdružený, půjde-li o komplexní čísla. V případě funkcionálního módu se skal. součiny počítají jako integrály s využitím Newtonovy aproximační metody integrování. Pro výpočet korelační matice slouží funkce `frcmx`. Samotné řešení framů pak spouští funkce `frsol`. Pro vykreslení framu se volá funkce `frplot`. Důležitým údajem, se kterým se pracuje, je `frix(j) = n.ss` – zjištění údajů o atomu v j -tém sloupci matice parametrů, kde n je pořadové číslo atomu v segmentu a ss je číslo segmentu, neboli mateřské funkce. To nám pomůže identifikovat jeden konkrétní atom.

Pro aplikaci časových řad a náhodných procesů zde existují například funkce `armagen` – vygenerování realizace ARMA procesu se zadanými parametry, dále pak `armasim` – dopředná predikce ARMA procesu.

Hlavní funkcí, která zastřešuje hledání řídkého řešení, je `frappr`. Tato funkce vyžaduje nastavení několika parametrů, z nichž je nejdůležitější parametr `ZeroTolr`. Ten určuje jak moc bude aproximace přesná a to dvěma možnými způsoby:

1. Zadání počtu nejvhodnějších atomů, které vybere.
2. Zadání tolerance, která pak rozhodne, které atomy budou vybrány a které nikoliv.

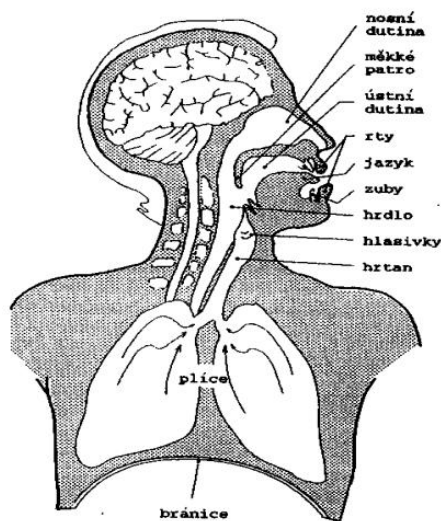
Výstupem celého cyklu výpočtů je skupina atomů s vypočtenými vahami, které model vyhodnotil jako řídké řešení pro zadaný vstupní segment.

Mezi další možnosti toolboxu patří také nastavení vlastností chování, např. nastavení defaultních parametrů, nastavení práce s grafy či regulace chybových hlášení – `frset`, `frget`.

5 ÚVOD DO PARAMETRIZACE ŘEČI

5.1 Vlastnosti řečového signálu

Jako řečové signály jsou uvažovány akustické signály vycházející z dutiny ústní. Tyto signály slouží primárně k dorozumívání mezi lidskými jedinci a mají svá charakteristická specifika, která nám v dalším zkoumání výrazně zúží množinu předpokládaných signálů k parametrizaci (např. periodicitu u některých hlásek). Na obrázku 5.1 je zakreslena cesta tvorby řeči a uvedeny části lidského těla, které se na ní podílejí:

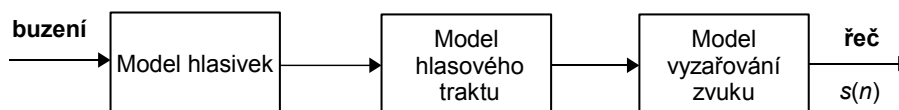


Obr. 5.1: Tvorba řeči a její model. Obrázek převzat z [4]

Princip tvorby hlásek spočívá v tom, že dýchací svaly (zejména bránice) vyženou z plic vzduch, který pak postupně prochází jednotlivými částmi hlasového traktu (hrtan, hlasivky, hrdlo, měkké patro, ústní a nosní dutina, jazyk, zuby, rty), přičemž každá tato část má jistý podíl na výsledném znění hlásky. **Hlásky** máme **znělé** (*a, e, i, o, u, y, b, d, g, v, z, ...*) a **neznělé** (*p, t, k, f, s, c, ch, č, x, ...*). U znělých hlásek se rozeznávají hlasivky, kdežto u neznělých nikoliv. Můžeme si to v praxi otestovat tak, že si při vyslovování znělé hlásky přiložíme ruku na krk do oblasti hlasivek a pak jasně ucítíme, že před samostatným vyslovením znělé hlásky se rozezní – začnou vibrovat. Vibrace vzniká tím, že pod tlakem proudu vzduchu vycházejícího z plic dochází k modulaci tohoto proudu hlasivkami. Kmitočet hlasivek je charakterizován základním tónem lidského hlasu, který má frekvenci o tzv. **pitch periodě**, což lze chápat jako kmitočet základního tónu, který tvoří základ znělých hlásek. Tento kmitočet se u každého člověka liší a určuje výšku hlasu jisté osoby. Kmitočet základního tónu je jedním z parametrů charakterizujících řečový signál.

Dále jsou zde důležité podíly vyšších harmonických kmitočtů, které hlas dobarvují a posluchač tak pozná „kdo to mluví“. Dalším důležitým parametrem je **hlasitost** (zesílení signálu) [4].

Celá soustava řečtovorných orgánů se dá charakterizovat jako model složený z několika systémů, z nichž každý určitým způsobem upravuje proud vzduchu vydechnutého z plic. Nazývá se **model hlasového traktu** a jeho schéma je zakresleno na obrázku 5.2 [4].



Obr. 5.2: Model pro tvorbu řečového signálu

5.2 Segmentace řeči

Při kódování řeči se celistvý řečový signál dělí na krátké úseky, tzv. **segmenty**, jejichž délka je nejčastěji 10–30 ms (tj. 80–240 vzorků při $f_{vz} = 8$ kHz) [12]. Signál v takto dlouhém úseku je považován za kvazistacionární a je tedy možné ho popsat jednou sadou parametrů. Segmenty jsou pak analyzovány a rozhodující je jejich typ. Rozlišujeme čtyři základní typy úseků řeči:

1. **znělý segment** – považován za periodický, dá se modelovat pomocí několika harmonických složek,
2. **neznělý segment** – neperiodický, považován za náhodný, dá se modelovat vhodně vyfiltrovaným bílým šumem,
3. **pauza** – hlasitost segmentu je natolik malá, že ho lidské ucho neslyší,
4. **přechodný segment** – kombinace dvou typů segmentů, nejčastěji nástup či doznívání znělé hlásky.

Rekonstrukce řeči ze segmentů v časové oblasti:

Aby docházelo při rekonstrukci řeči k plynulému navazování segmentů na sebe, používá se překryv segmentů (overlap), který se nejčastěji volí 50 % délky segmentu. Aby při následné syntéze (skládání segmentů za sebe) bylo dosaženo správné rekonstrukce, je třeba ošetřit následující dvě podmínky:

1. Vynásobit segment váhovacím oknem

To se dělá proto, aby při přičtení segmentů s překryvem nedocházelo ke zvětšení energie v oblasti překryvu. Nejčastěji se používá Hannovo a Hammingovo váhovací okno.

2. Zajistit synchronizaci pitch periody

Jednotlivé segmenty mají zpravidla jiné pitch periody a je třeba, aby na sebe navazovaly. Toto se řeší markováním period, což je algoritmus, který na základě znalosti velikosti segmentu a překryvu vypočítává pozice začátků jednotlivých period v segmentu. Zabrání se tak vzniku fázového rozdílu period v po sobě jdoucích segmentech.

Tomuto způsobu rekonstrukce se říká metoda překrývání přesahu se synchronním základním tónem řeči, též TD-PSOLA (*Time Domain Pitch Synchronous Overlap Add*) [12, 14].

5.3 LPC analýza segmentu řeči

LPC analýza (*Linear Predictive Coding*) hledá parametry modelu hlasového traktu jako IIR filtru (model AR), který je popsán přenosovou funkcí

$$H(z) = \frac{1}{A(z)}, \quad (5.1)$$

$$A(z) = 1 + a_1z^{-1} + a_2z^{-2} + \dots + a_pz^{-p} = \sum_{i=1}^p a_i z^{-i}, \quad (5.2)$$

kde a_1, a_2, \dots, a_p jsou **LP koeficienty** (*Linear Predictive*) řádu p . Při $f_{vz} = 8$ kHz se nejčastěji volí $p = 10$.

Nalezení LP koeficientů

Samotná lineární predikce vychází z faktu, že ze známých vzorků signálu se dají předpovědět vzorky následující. Předpověď je počítána jako lineární kombinace známých vzorků, proto se také nazývá lineární. Při každé predikci dochází k určitým nepřesnostem v předpovědi a vzniká chyba predikce. Algoritmus počítající LP koeficienty je založen na minimalizaci takovéto chyby. Kompletní rozbor této problematiky je v [4, 21]. Výsledky bádání nad minimalizací chyb vedou k následujícím dvěma krokům:

1. Sestavení soustavy rovnic s autokorelačními koeficienty $R(i)$ segmentu:

$$\begin{aligned}
 R(0)a_1 + R(1)a_2 + \dots + R(p-1)a_p &= -R(1) \\
 R(1)a_1 + R(0)a_2 + \dots + R(p-2)a_p &= -R(2) \\
 &\vdots \\
 R(p-1)a_1 + R(p-2)a_2 + \dots + R(0)a_p &= -R(p)
 \end{aligned} \tag{5.3}$$

2. Řešení této soustavy *Levinson-Durbinovým algoritmem* [4], kdy po provedení všech jeho kroků obdržíme následující sadu čísel:

$$\begin{array}{cccccc}
 a_1^{(1)} & a_1^{(2)} & a_1^{(3)} & \dots & a_1^{(p)} \\
 & a_2^{(2)} & a_2^{(3)} & \dots & a_2^{(p)} \\
 & & a_3^{(3)} & \dots & a_3^{(p)} \\
 & & & \ddots & \vdots \\
 & & & & a_p^{(p)}
 \end{array} \tag{5.4}$$

LP koeficienty $a_i = a_i^{(p)}$ pak leží v pravém sloupci této soustavy a fungují jako AR koeficienty rekonstrukčního filtru hlasového traktu viz vztah (5.1). Nejvýznamnější vlastností rekonstrukce dle nich je vyhlazení průběhu spektrální obálky řečového signálu [4].

5.4 Řečové kodéry

V dnešní době existuje pestrá škála řečových kodérů. Dají se rozdělit do dvou skupin:

a) Kodéry založené na kódování tvaru signálu

Jsou to např. ADPCM (*Adaptive Differential Pulse Code Modulation*), CVSD (*Continuously Variable Slope Delta*), které téměř nevyužívají charakteristik řečového signálu a jsou založeny na delta modulaci či diferenční pulzně-kódové modulaci. Výstupní bitové rychlosti se pohybují nejčastěji mezi 12 a 32 kbit/s. Jsou již jen málo využívány hlavně pro nízký stupeň komprese [7].

b) Kodéry založené na statistickém modelu hlasového traktu

Základem těchto kodérů je LPC analýza jednotlivých segmentů. Nalezené LP koeficienty jsou pak všelijakými způsoby dále kódovány pro pozdější přenos (např. vektorová kvantizace), což umožňuje celé kódování zefektivnit. Dále jsou také ze segmentů určeny další nezbytné parametry – pitch perioda, zesílení a typ segmentu. Na

tomto základě fungují dnes nejpoužívanější řečové kodéry – CELP (*Code-Excited Linear Prediction*), VSELP (*Vector Sum Excited Linear Prediction*) či RPE-LTP (*Regular Pulse Excitation – Long-Term Prediction*). Výstupní bitové rychlosti se pohybují od 2,4 do 12 kbit/s [7].

6 NÁVRH A REALIZACE KODÉRŮ

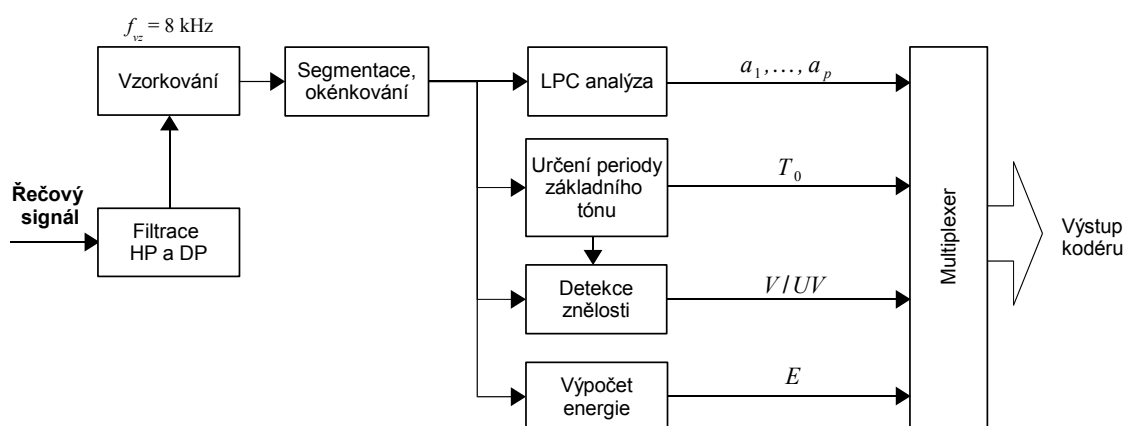
Hlavní částí této práce je simulace dvou řečových kodérů i s dekodéry, které realizují analýzu a syntézu řeči pomocí klasické metody LPC a v druhém případě se jedná o moderní metodu přeparametrizovaných modelů. Jako programovací jazyk je použit MATLAB, ve kterém máme k dispozici již zmiňovaný toolbox *frames*. Výsledky práce těchto dvou kodérů jsou uvedeny v sekci 7.

6.1 Kodér LPC

Jedná se již o mnohokrát vyzkoušený základní kodér tohoto typu. Hlavní účelem jeho simulace je reference pro porovnání výsledků s druhým kodérem s přeparametrizovanými modely.

6.1.1 Kodér

Jako kodér LPC jsem použil mnou již navržený kodér z mé bakalářské práce na téma *LPC kódování řeči* [21]. Jeho blokové schéma je na obrázku 6.1.



Obr. 6.1: Blokové schéma kodéru LPC

Jeho funkce je následující:

Na jeho vstupu bude řečový signál ve **.wav* souboru, který se načte do podoby vektoru pomocí funkce `wavread` [20]. Jeho součásti a princip funkce jsou následující [21]:

Filtrace HP a DP

Načtený zvuk může obsahovat nežádoucí nízkofrekvenční složky, které je třeba pro

další analýzu odstranit (preemfáze). Tyto složky odstraní horní propust (HP) s mezní frekvencí 60 Hz. Protože bude použita vzorkovací frekvence $f_{vz} = 8$ kHz pro telefonní kvalitu, je nutné mít splněnou Nyquistovu podmínku pro vstupní zvukový signál, a proto jej vyfiltrujeme dolní propustí (DP) s mezní frekvencí 4 kHz. Kombinaci HP a DP místo pásmové propusti použijeme proto, že PP s parametry pro filtrování řeči by měla několikanásobně vyšší řád.

Vzorkování

Vyfiltrovaný řečový signál bude převzorkován z původní vzorkovací f_{vz} na 8 kHz.

Segmentace a okénkování

Posloupnost vzorků zde bude rozdělena na segmenty, které budou následně podrobovány další analýze. Délka segmentu `win` je volitelná v rozmezí 180 až 240 vzorků, což při $f_{vz} = 8$ kHz odpovídá 22,5 až 30 ms. Dolní hranice je zvolena tak, aby bylo možno spolehlivě detekovat periodu základního tónu i pro nejnižší hranici pro detekci, t.j. 80 Hz. Horní hranice byla zvolena tak, aby byl segment ještě považován za kvazistacionární [12]. Dalším parametrem je překryv `overlen`, který je možno zvolit v rozmezí 0 až 50 % (v programu je ovšem zadáván v počtu vzorků). Co se týče okénkování, tak kodér LPC byl pro účely analýzy vyladěn pro pravoúhlé okno. Okno pro sestavení segmentu řeší až poslední blok pro sestavení nahrávky. U druhého kodéru s přeparametrizovanými modely bylo pro analýzu použito Hannovo okno.

Zjištění periody základního tónu T_0 (pitch)

Perioda základního tónu pro segment je vyhodnocována v rozmezí 80 až 400 Hz [12]. Nejčastěji jsou používány metody autokorelační a keprální [5]. V programu jsem ovšem použil již vyladěnou metodu AMDF (*Average Magnitude Difference Function*), použitou v [21]. AMDF je funkce

$$AMDF(\tau) = \sum_{n=0}^{N-1} |s(n) - s(n + \tau)|, \quad (6.1)$$

kde τ označuje zpoždění vzorků. Na této funkci jsou potom hledána minima, dle kterých je stanovena perioda základního tónu [5].

Detekce znělosti

Jedná se o nastavení tzv. *V/UV bitu*, který určuje o jaký druh segmentu se jedná. Rozhodování probíhá na základě analýzy tří parametrů: existence periody základního tónu T_0 , normalizovaného autokorelačního koeficientu $r1$ [6] a počtu průchodu signálu nulou $n0$. Tyto parametry jsou váhovaně sčítány a v případě, že je součet vyšší než jistý práh, je segment vyhodnocen jako neznělý *V/UV bit* je nastaven na 1. Je-li znělý, pak je *V/UV bit* nastaven na 0 [21].

Zjištění energie segmentu E

V tomto případě je určena jednoduše pomocí průměru kvadrátů hodnot signálu v segmentu. Při rekonstrukci bude sloužit jako zesílení rekonstrukčního filtru.

$$E = \frac{1}{N} \sum_{n=1}^N s^2(n). \quad (6.2)$$

LPC analýza

Úkolem LPC analýzy je stanovit koeficienty a_1, \dots, a_p rekonstrukčního filtru (viz vztah (6.3)). Pro každý segment bude vypočítána autokorelace a získané autokorelační koeficienty budou užity pro Levinson-Durbinův algoritmus pro vyřešení soustavy jejich rovnic (5.3) [4, 12]. Tímto algoritmem je získána sada LP koeficientů důležitých pro rekonstrukci. Jejich počet se odvíjí od zadaného parametru `order`, který je možno volit v rozmezí 2 až 20, standardně se používá `order = 10`, pro tzv. LPC-10 analýzu [6].

Multiplexer

Tento blok sdružuje jednotlivé zjištěné parametry pro každý segment do jednoho rámce. Příklad výsledného rámce je na obrázku 6.4 v kapitole 6.1.3, která se věnuje kódování dat potřebných pro rekonstrukci. Z těchto rámců je potom vytvořen bitový tok směrem k přijímači. Ve výsledném programu není multiplexer realizován, protože by byl pro účely této práce nadbytečný a zbytečně by komplikoval chod programu.

6.1.2 Dekodér

Blokové schéma dekodéru řeči z LP koeficientů je na obrázku 6.2.

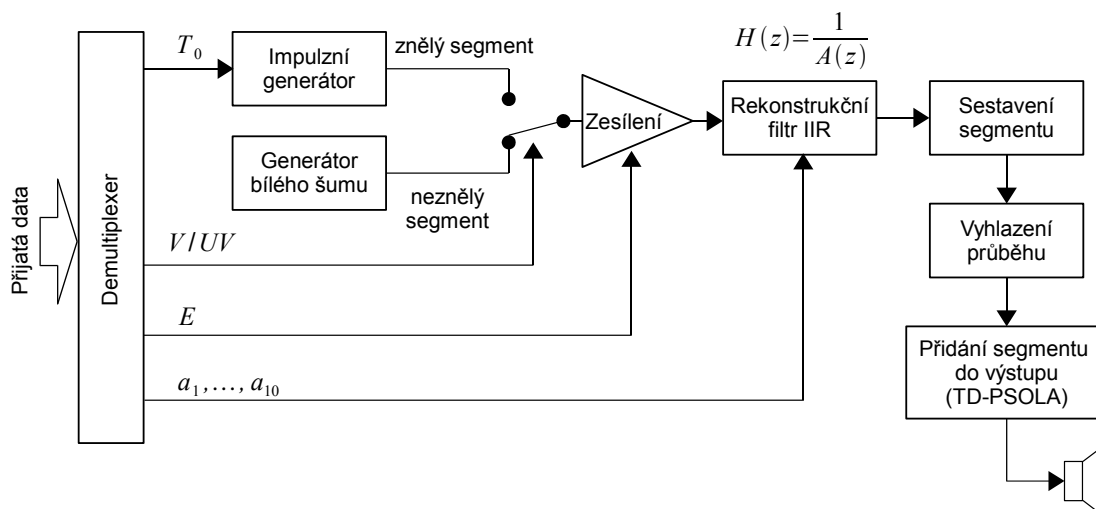
Schéma vychází ze vzoru z [6] a jeho konkrétní funkce byla doladěna na základě experimentů. Rekonstrukce probíhá dvěma způsoby dle typu segmentu.

Neznělý segment

V případě neznělého segmentu je situace jednodušší – rekonstruuje se pomocí filtrace bílého šumu rekonstrukčním filtrem. Tento filtr má přenosovou funkci určenou LP koeficienty:

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 + \sum_{i=1}^p a_i z^{-i}}, \quad (6.3)$$

Následně je dle energie E signálu v segmentu upraveno jeho zesílení. Takto vytvořený segment délky `win` je následně poslán k přidání do výstupní nahrávky.



Obr. 6.2: Blokové schéma dekodéru LPC

Znělý segment

Tento typ segmentu se rekonstruuje poněkud složitěji. Opět je použit rekonstrukční filtr se stejnou přenosovou funkcí – viz vztah (6.3), akorát nás zajímá jeho impulzní odezva. Znělý segment je rekonstruován jako odezva na sérii jednotkových impulzů, které vstupují do filtru jako periodická sekvence s periodou odpovídající základní periodě řeči T_0 . O zesílení rozhoduje opět energie E . V praxi je ještě nutné postarat se o homogenitu segmentu (zabránění nežádoucích zesílení a šumu způsobených překrytím impulzních odezev) a zajistit návaznost period v rámci segmentu. Impulzní odezvy filtru se při řádu analýzy $p = 10$ až překvapivě podobaly původní periodě vlastní řeči. Nicméně, použitý filtr je typu IIR, jeho odezva je tedy nekonečná a je třeba ji někde „utnout“. Experimenty vedly ke stanovení následujícího postupu:

1. Vytvoření impulzní odezvy délky $2T_0$.
2. Utlumení její pravé části.
3. Poskládání tzv. supersegmentu.
4. Výběr segmentu délky win se synchronizovanou periodou.
5. Vyhlazení nespojitostí v oblasti návaznosti period.
6. Přidání segmentu do nahrávky.

Při použití odezvy délky T_0 by docházelo k velmi nepřírozenému napojení na další periodu, proto jsem pro modelování použil **odezvu délky $2T_0$** . Pro případ pomalé

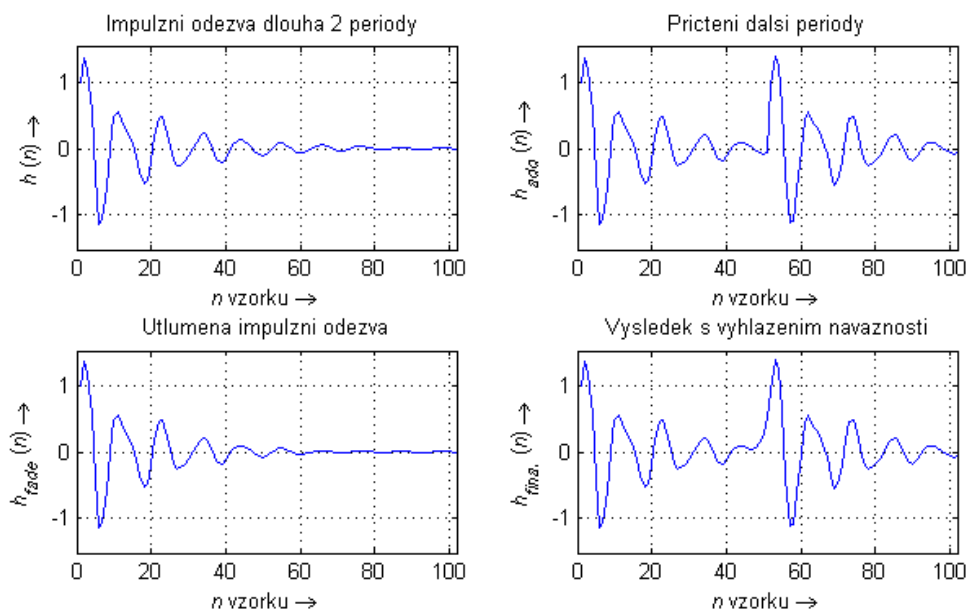
konvergence odezvy k nové hodnotě je ještě tato odezva od jisté části tlumena tak, aby byla **v čase $2T_0$ zcela utlumena**. Poskládáním několika takhle upravených odezev s překryvem délky T_0 dostaneme úsek signálu, který nazvěme **supersegment**. Z tohoto supersegmentu je třeba zřezat segment délky win tak, aby byla zajištěna synchronizace period s předchozím segmentem. Ta je zajištěna podprogramem na bázi **rekonstrukce TD-PSOLA**, který pracuje s hodnotami T_0 , win a $overlen$. Pro každý segment je vypočtena hodnota $d1$ – vzdálenost začátku první periody od začátku segmentu a $d2$ – vzdálenost začátku poslední periody od konce segmentu. Ze znalosti uvedených hodnot a $d2$ z předchozího segmentu pak je spočtena $d1$ důležitá pro sestavení segmentu. Je tak dosaženo **synchronizace period** pro celou nahrávku bez potřeby přídatných parametrů z kodéru. Další úprava spočívá ve vyhlazení návazností jednotlivých period. U většiny hlásek totiž dochází v bodě návaznosti k fázovým či derivačním nespojitostem, což má za následek značné zkreslení výsledného signálu. Tento problém jsem vyřešil kvadratickým **vyhlazením** v oblasti kolem zlomu. Algorismus pracuje se třemi hodnotami derivace:

1. Derivace signálu ve vzdálenosti 5 vzorků před zlomem.
2. Derivace v bodě zlomu.
3. Derivace 1–3 vzorky za bodem zlomu (pro realizaci vyhlazení musí být kladná).

Pomocí těchto tří hodnot je vytvořena po částech lineární derivace, na základě které je vymodelován vyhlazený průběh signálu v oblasti zlomu. Tato úprava se velmi výrazně projeví na kvalitě výsledné rekonstrukce. Průběhy ukazující výsledky jednotlivých kroků úpravy impulzní odezvy jsou na obrázku 6.3. Impulzní charakteristika zde patří náběhu samohlásky „a“.

Přidání segmentu do výstupu

Zbývá už jen dát to celé dohromady – segmenty se překrývají v $overlen$ vzorcích, je tedy třeba zajistit správnou návaznost a okénkování. Je-li procentuální překryv na maximum, tedy 50 %, je výhodné použít některé z běžně používaných váhovacích oken, např. Hannovo. V programu k tomu existuje parametr $window$ (okno pro sestavení nahrávky), který umožňuje výběr z těchto oken: Hammingovo, Hannovo, Blackmanovo a Čebyševovo. Je-li překryv menší, pak je lepší použít lineární tlumění začátku/konce segmentu v oblasti překryvu – $window = 0$. Takto váhované segmenty se na sebe napojí pomocí přičtení překryvu a vzniká výsledná rekonstruovaná nahrávka.



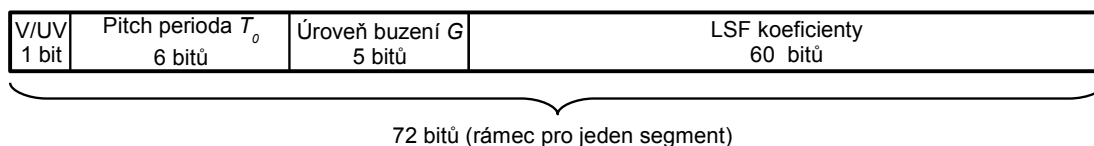
Obr. 6.3: Práce s impulzní charakteristikou při rekonstrukci znělého segmentu

6.1.3 Návrh kódování do bitového toku

Parametry nezbytné pro rekonstrukci pomocí LP koeficientů:

- a_1, \dots, a_p – vektor délky p obsahující LP koeficienty,
- E – hodnota energie (zesílení) v segmentu,
- T_0 – perioda základního tónu řeči,
- $V/UV \text{ bit}$ – 1 bit určující typ segmentu (znělý/neznělý).

LP koeficienty jako takové jsou nevhodné pro přenos – je třeba je přesně kvantovat a každý z nich je důležitý pro výslednou kvalitu rekonstrukce. V praxi se přenáší v podobě LSF (*Line Spectral Frequencies*) koeficientů ω_i [13]. Ty korespondují s formantovými frekvencemi a jsou již vhodné pro účely kódování. Podoba možného datového rámce pro jeden segment je na obrázku 6.4. Rámec přenáší 10 LSF koeficientů korespondujících s deseti LP koeficienty (LPC-10 analýza). Detailní popis tohoto rámce je v [21].



Obr. 6.4: Bitový rámeček pro jeden segment z LPC kodéru

6.2 Kodér s přeparametrizovanými ARMA modely

Přeparametrizované modely mají velice příznivé předpoklady pro efektivní kompresi řeči, poněvadž vlastnosti řečových signálů jsou omezené a dá se předpokládat jejich vymodelování pomocí velmi malého počtu důležitých koeficientů. Následující návrh a realizaci jsem vytvořil na základě teoretických i praktických poznatků.

6.2.1 Kodér

Kodér pro aplikaci přeparametrizovaných modelů je v mnohém podobný kodéru LPC. Hlavní rozdíl je v povaze výstupních koeficientů, které jsou počítány vyspělejšími algoritmy. Jeho blokové schéma je na obrázku 6.5. Samotná realizace v programu MATLAB se od následně popisovaného kodéru mírně odlišuje. Onou odlišností není princip funkce, ale není plně automatizován tak, jako kodér LPC. Důvody, které k tomu vedly, byly způsobeny vysokou výpočetní náročností a nutností uživatelské obsluhy při výpočtech prostřednictvím toolboxu *frames*. Výstupy kodéru jsou nahrávány do **.mat* souborů, ze kterých si potom dekodér bere potřebné informace pro rekonstrukci, v tomto případě již automaticky.

Jeho funkce je následující:

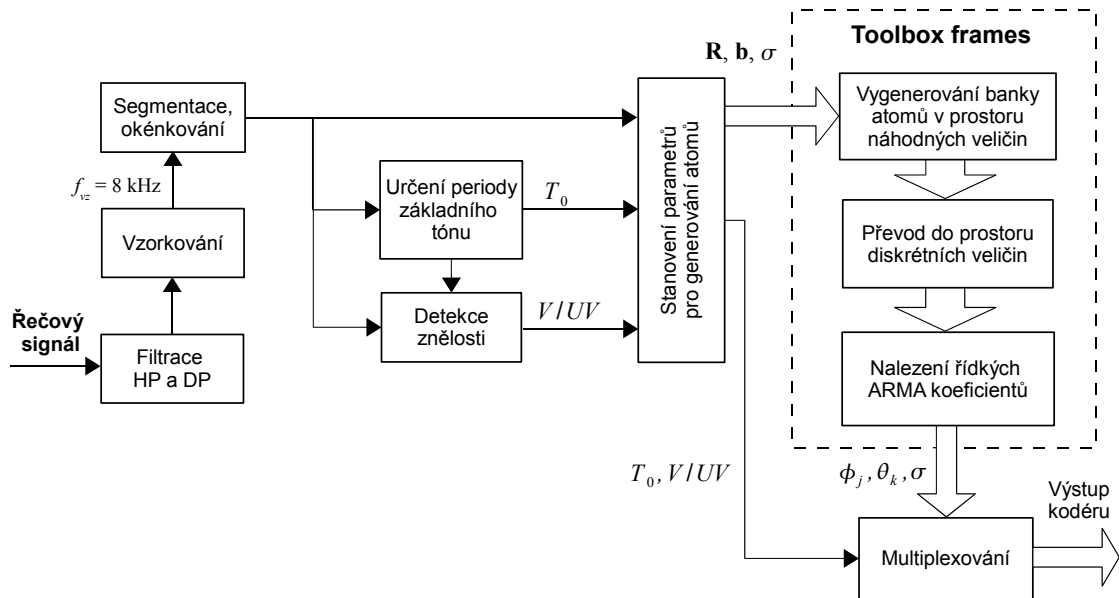
Na jeho vstupu je opět řečový signál ve *wav* souboru, který je načten do podoby vektoru pomocí funkce *wavread*. Následuje to samé, co v případě kodéru LPC; jedná se o tyto části:

Filtrace HP a DP, Vzorkování, Segmentace, Zjištění periody základního tónu, Detekce znělosti, Multiplexer.

Funkce i účel těchto bloků jsou identické s LPC kodérem.

Stanovení parametrů pro generování atomů

Pro použití toolboxu *frames* pro nalezení řídkých ARMA koeficientů je třeba stanovit následující parametry:



Obr. 6.5: Blokové schéma kodéru využívajícího přeparametrizované ARMA modely

- Nahodnocené řády P, Q odpovídající AR, MA modelům,
- váhy AR, MA koeficientů,
- σ – rozptyl Gaussovského šumu pro rekonstrukci (obdoba E u LPC),
- matice \mathbf{R}, \mathbf{b} ze vztahu $\mathbf{b} = \mathbf{R}\xi$ (3.6).

První dvě položky jsou nastaveny explicitně: P, Q se volí nejčastěji $P = Q = 10$, popř. 20. Váhy pro koeficienty slouží jako potlačení významnosti těch méně důležitých, tj. AR pro neznělý a MA pro znělý segment. Např. nastaví-li se váhy pro znělý segment na 0,5, pak musí být libovolný MA koeficient dvakrát významnější, aby byl vybrán jako řídký. Hodnoty σ, \mathbf{R} a \mathbf{b} jsou počítány z hodnot signálu v segmentu – pomocí autokorelace, inovačního algoritmu a sestavení Toeplitzových matic (viz kapitola 3.2.2). Neznělé segmenty jsou zde ponechány v původní podobě, ty znělé jsou analyzovány ve formě sekvence několika po sobě jdoucích period vynásobených Hannovým oknem. Hodnota σ se vypočítá jako první vzorek autokorelace segmentu.

Toolbox *frames* – nalezení řídkých ARMA koeficientů

Jedná se o jádro celého kodéru. Celý tento blok realizuje funkce `frappr`, která zastřešuje kompletní výpočet řídkých ARMA koeficientů. Je volána skriptem:

```
[Xi,s,ZeroTol,INFO] = frappr([],R,b,lambda,po,xi0,ZeroTolr,AtomINFO),
```

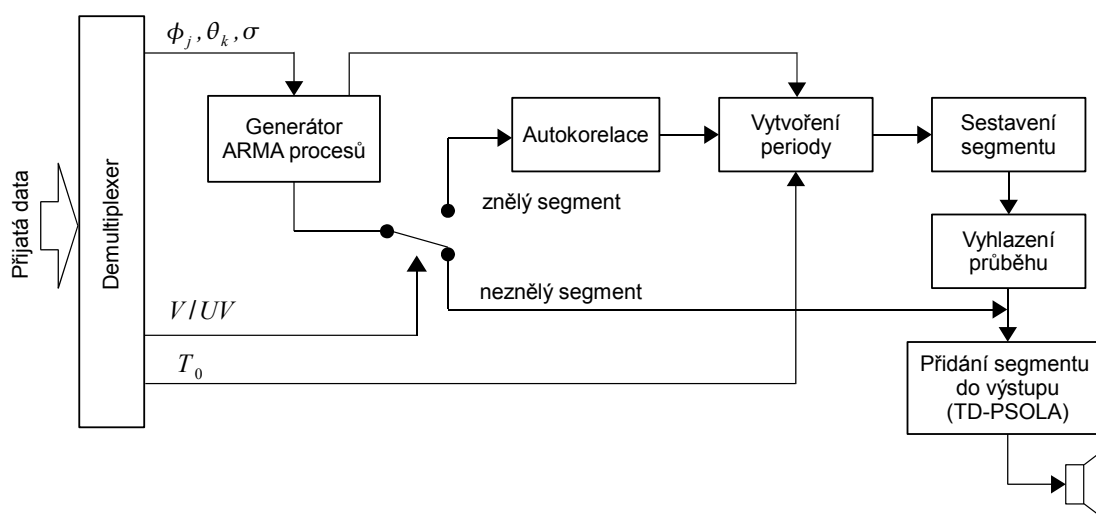
kde vstupní parametry znamenají následující: `[]` je defaultní počet bodů, pro které bude algoritmus generovat atomy, \mathbf{R}, \mathbf{b} pak ony již stanovené matice \mathbf{R} a \mathbf{b} , `lambda` je

vyhlazovací parametr (explicitně), po jsou váhy koeficientů, \mathbf{x}_i je počáteční odhad koeficientů (defaultně všechny rovny jedné), ZeroTolr je parametr, určující „řídkost“ řešení v procentech dovolené odchylky a AtomINFO obsahuje hodnoty o koeficientech (typ, řád atd.). Výstupy funkce jsou: \mathbf{X}_i je matice, obsahující hodnoty koeficientů ξ_i pro všechny fáze výpočtů (tedy i ty výsledné řádké), \mathbf{s} je standardní odchylka rozptylu šumu vypočítaná z reziduálního šumu, ZeroTol je pak absolutní hodnota řídkosti dosažená po posledním iteračním kroku a INFO obsahuje přídavné informace z výpočtů.

Nejdůležitějším výstupem jsou koeficienty \mathbf{X}_i obsahující $\boldsymbol{\xi} = \{\boldsymbol{\Phi}_P, \boldsymbol{\Theta}_Q\}$. Funkce je zapíše do matice, která má 5 sloupců – v prvním jsou jejich počáteční odhady, v dalších čtyřech jsou výsledky po každém kroku modifikovaného BPA-4 algoritmu. Matice je ještě rozdělena na část AR – horních P řádků ($\boldsymbol{\Phi}_P$) a MA – spodních Q řádků ($\boldsymbol{\Theta}_Q$). Požadované řádké koeficienty $\{\phi_j, \theta_k\}$ jsou v posledním, tedy pátém sloupci. Koeficient, který nebyl shledán jako řídký, má zde hodnotu 0. Ty řídké jsou na pozici příslušného řádu a mají nenulovou hodnotu.

6.2.2 Dekodér

Blokové schéma dekodéru řeči z řídkých ARMA koeficientů je na obrázku 6.6.



Obr. 6.6: Blokové schéma dekodéru využívajícího přeparametrizované ARMA modely

Narozdíl od dekodéru LPC je zde odlišnost v podstatě rekonstrukce. Co se týče skládání segmentu z period či skládání celé nahrávky ze segmentů, existuje zde stejný mechanismus. Není zde používán filtr, ale generuje se přímo realizace ARMA procesu z koeficientů z kodéru.

Generátor ARMA procesů

V tomto bloku jsou generovány ARMA procesy popsány v kapitole 3.2.1. Funkce na jejich generování je součástí *frames* pod názvem *armagen*. Pro vygenerování jedné realizace stačí zadat řídící koeficienty $\{\phi_j, \theta_k\}$, rozptyl šumu σ , délku procesu a počáteční podmínky.

Aplikace takovýchto procesů do výsledných rekonstrukcí řeči si vyžádala mnoho experimentů. Opět se odlišují postupy dle znělosti segmentu:

Neznělý segment

Náhodné statistické ARMA procesy se ukázaly jako výborné pro modelování neznělých hlásek. Nejsou totiž citlivé na konkrétní průběh realizace, protože ucho vnímá jen jeho šumové vlastosti. Místo zesílení je zde použit rozptyl bílého šumu σ , který dostatečně reprezentuje míru zesílení signálu. Jedná se tedy o vytvoření pomocí jediné realizace procesu délky *win* s danými parametry.

Znělý segment

Realizace ARMA procesů mají různé průběhy, přičemž jejich průběhy samotné se ukázaly jako nedostatečné pro vymodelování znělého segmentu. Bylo tedy třeba experimentovat a na základě výsledků určit postup rekonstrukce. Byla zjištěna následující fakta:

1. V dostatečně dlouhé realizaci nachází několik úseků s průběhem velmi podobným periodě původního segmentu.
2. Autokorelace každé realizace má téměř totožný průběh, navíc velmi podobný periodě původního segmentu.

Tím byly nalezeny dva významné „kamery“, na kterých bylo možno stavět. Čím delší a uhlazenější realizace byla, tím její autokorelace byla nepříznivě hladší. Ukázka vygenerovaného ARMA procesu a jeho autokorelace pro samohlásku „a“ je na obrázku 6.7. První myšlenkou bylo vyrobit několik realizací a jejich autokorelace zprůměrovat, aby se vyrovnaly rozdíly průběhů při různých realizacích. Druhou myšlenkou pak bylo váhovaně sečíst průběh autokorelace realizace s nějakou vlastní částí průběhu realizace. Experimenty prokázaly vhodnost použití obou těchto nápadů. Postup rekonstrukce je tedy následující:

1. Generace ARMA procesu na základě ϕ_j , θ_k a σ , délka realizace minimálně 6 period T_0 .
2. Nalezení periody délky T_0 s nejvyšší energií v intervalu T_0 v celé realizaci, která začíná jejím maximem (kvůli synchronizaci s autokorelací).



Obr. 6.7: Realizace ARMA procesu a jeho autokorelace

3. Výpočet normované autokorelace realizace procesu.
4. Vážený součet v poměru 1:3 (nejsilnější perioda : autokorelace).
5. Opakovat body 1 až 4 (2–3 krát), spočítat aritmetický průměr průběhů.

V této chvíli je již k dispozici signál, ze kterého se vezme úsek délky $2T_0$. Dále pokračuje rekonstrukce stejně jako u LPC (ultumení, supersegment, vyhlazení návazností period, skládání nahrávky).

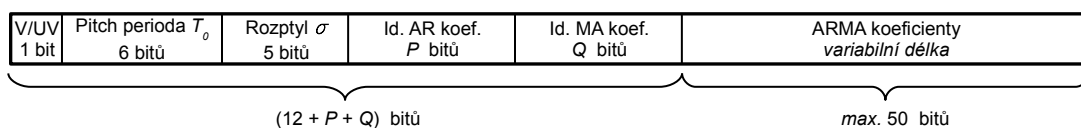
Data důležitá pro rekonstrukci pomocí ARMA (ϕ_j , θ_k , σ , T_0 atd.) si program bere z připravených *.mat souborů (pro každý segment je zvláštní soubor).

6.2.3 Návrh kódování do bitového toku

Parametry nezbytné pro rekonstrukci pomocí ARMA procesů:

- ϕ_j – vektor délky P obsahující řádké AR koeficienty + nuly
- θ_k – vektor délky Q obsahující řádké MA koeficienty + nuly
- σ – hodnota rozptylu pro segment
- T_0 – perioda základního tónu řeči
- $V/UV \text{ bit}$ – 1 bit určující typ segmentu (znělý/neznělý)

U řídkých ARMA koeficientů záleží především na jejich konkrétních nalezených řádech a není třeba je kódovat tak přesně, jako v případě LPC. Následující návrh počítá s 5 bity pro každý z nich (tj. 32 možných hladin). Pro identifikaci řádu budou ještě použita pole identifikací AR a MA koeficientů. Ta budou obsahovat jedničky na pozici řádů vybraných řídkých koeficientů. Ostatní jejich bity budou naplněny nulami. Za nimi bude pole obsahující vlastní hodnoty koeficientů. Podoba navrhovaného rámce pro jeden segment je na obrázku 6.8. Jeho délka bude proměnlivá, závislá na počtu řídkých ARMA koeficientů.



Obr. 6.8: Bitový rámec pro jeden segment z ARMA kodéru

Například při volbě $P = Q = 10$ (nepřenáší se – přijímač zná hodnoty P, Q) a počtu 7 řídkých ARMA koeficientů bude výsledná délka rámce $1+6+5+2\cdot 10+7\cdot 5 = 67$ bitů.

7 VÝSLEDKY REALIZACE

Oba simulované řečové kodéry byly postupně zdokonalovány až do spolehlivě fungující podoby. Tato kapitola se věnuje jejich výstupům a porovnání výsledků jejich činnosti. Hlavním účelem jejich vyladění bylo dosažení co nejvěrnější zvukové podoby vzhledem k originální nahrávce.

7.1 Charakter řídkých ARMA koeficientů

Jako první výsledky simulace uvedu charakter rozložení řídkých koeficientů AR a MA po třetím kroku iterace procedury `frappr`. V této fázi jsou již nalezeny ty významné, ovšem uživatelská obsluha této aplikace zde dovoluje vybrat počet koeficientů, které půjdou do posledních dvou iterací. Na začátku iteračního algoritmu jsou stanoveny zadané řády modelu P , Q . Prvním zjištěním bylo, že výsledný počet řídkých koeficientů není závislý na velikosti zadaných řádů P , Q . Např. při volbě $P = Q = 10$, tedy celkem se vybíralo z 20 koeficientů a při $P = Q = 20$ (celkem 40 koeficientů) byl vybrán v obou případech stejný počet těch řídkých. Situace je znázorněna na obrázcích 7.1 a 7.2. Co již měly rozdílné, byl výběr jejich konkrétních řádů. Pro příklad uveďme následující situaci:

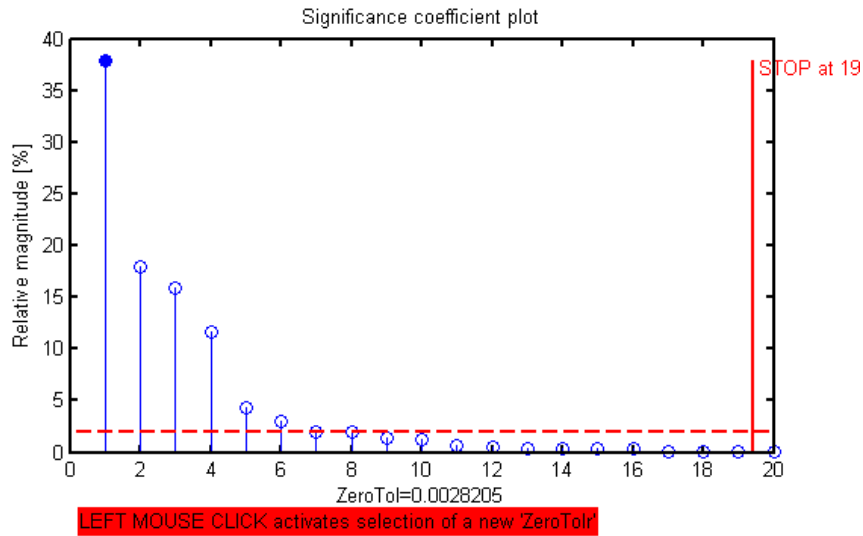
Mějme znělý segment, kde jsou jako ty nejvýznamnější vybírány koeficienty AR. Při volbě $P = Q = 10$ mohl algoritmus vybírat mezi AR se zpožděním v rozmezí 1 až 10. Dvěma nejvýznamnějšími koeficienty se tak mohly stát AR(3) a AR(5) (údaj v závorce znamená jejich zpoždění). Naopak při volbě $P = Q = 20$ bylo možné, že algoritmus vyhodnotil pro stejný segment jako nejvíce řídké koeficienty AR(6) a AR(18). Tato skutečnost svědčí o citlivosti algoritmu na počáteční podmínky, protože zde bylo použito lineární programování. Nebylo pravidlem, že koeficienty nejnižších řádů byly ty nejvýznamnější. Algoritmus vybral vždy tu nejvíce řídkou konstelaci ze všech možných, i ze zdánlivě nepodstatných koeficientů.

Charakter ARMA koeficientů pro různé typy segmentů

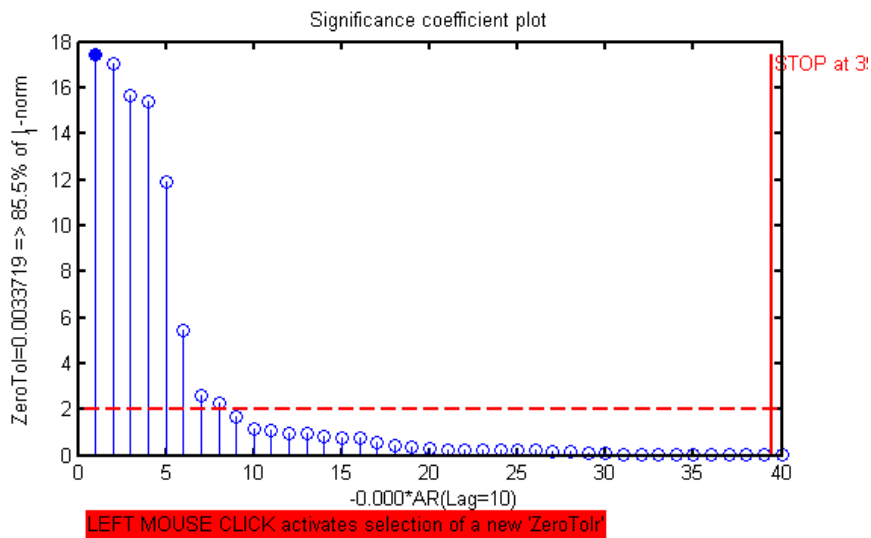
Znělé segmenty

U znělých segmentů se ukázalo, že řídkých koeficientů je velmi málo. Jako vyhovující jsem bral všechny, které vyhovovaly podmínce `ZeroTolr = 2%`. Jejich počet se lišil podle kvality nahrávky – u méně kvalitních jich bylo více, protože jejich průběh byl složitější a významnými se stávaly i koeficienty MA. Globálně se ale pohybujeme v počtu 5 až 10 řídkých koeficientů dle kvality nahrávky a typu hlásky. Na obrázku 7.1 jsou znázorněny seřazené koeficienty AR i MA dle jejich významnosti pro znělý segment při volbě $P = Q = 10$. Pro porovnání jsou na dalším obrázku 7.2

koeficienty pro stejný segment při volbě $P = Q = 20$.



Obr. 7.1: ARMA koeficienty pro znělý segment pro $P = Q = 10$

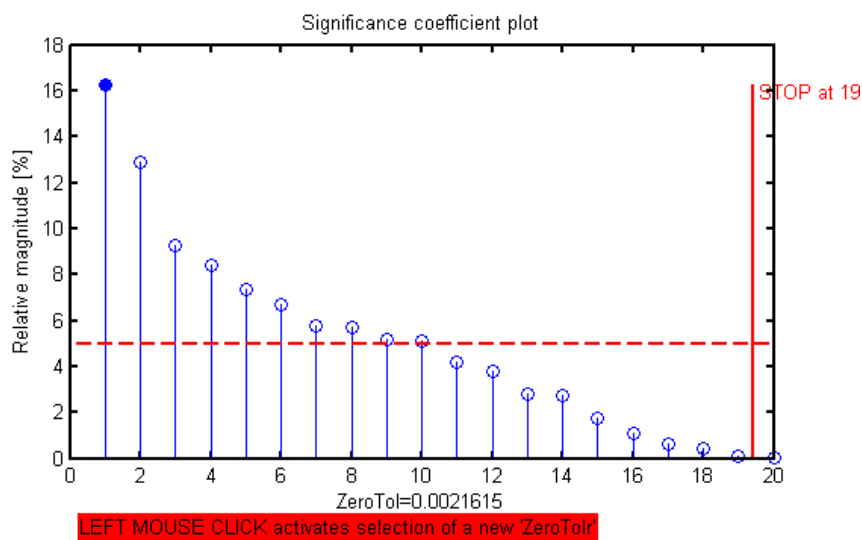


Obr. 7.2: ARMA koeficienty pro znělý segment pro $P = Q = 20$

Neznělé segmenty

Neznělé části řeči mají charakter šumu, obsahují tedy mnoho frekvencí a tím pádem je obecně charakterizuje větší počet koeficientů s podobnými vahami. To se ukázalo i při hledání těch řídkých, kdy vycházela téměř lineárně klesající posloupnost významnosti koeficientů. Vzhledem k nižší energii a tím pádem menšímu podílu neznělých hlásek na subjektivní kvalitě výsledné rekonstrukce, byla zvednuta `ZeroTolr` na 5%. I tak této podmínce většinou vyhovovalo do deseti koeficientů, především

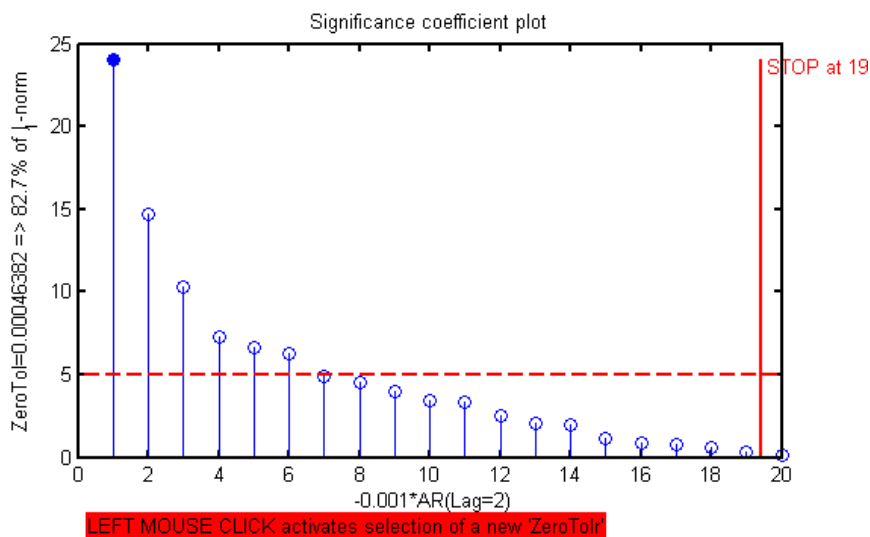
typu MA. Na obrázku 7.3 jsou znázorněny seřazené ARMA koeficienty pro neznělý segment.



Obr. 7.3: ARMA koeficienty pro neznělý segment pro $P = Q = 10$

Segmenty na přechodu znělosti-neznělosti

Pro tyto segmenty docházelo k výběru obou typů AR i MA koeficientů mezi řídké. Znázorněny jsou na obrázku 7.4.



Obr. 7.4: ARMA koeficienty pro přechodný segment pro $P = Q = 10$

7.2 Porovnání kvality rekonstrukce obou kodérů

7.2.1 Metody porovnávání kvality rekonstrukce

Teorie zpracování řeči rozlišuje dva základní přístupy k porovnání kvality rekonstrukce řeči od originálu:

- *Subjektivní metody,*
- *Objektivní metody.*

Subjektivní metody jsou založeny na sluchovém vnímání posluchače či skupiny posluchačů. Rozlišuje se **kvalita** a **srozumitelnost** rekonstrukce. V případě kvality se jedná o zkoumání čistoty rekonstrukce, zda-li rekonstrukce není příliš zašuměná, syntetická, arytmiická, přibarvená atd. Srozumitelností se rozumí porozumění věcné stránky řeči, tedy obsahu jejího sdělení. Testování srozumitelnosti se provádí pro různě dlouhé slovní celky (slabiky, celá slova, věty, ...) pro smysluplná spojení a také pro náhodná spojení písmen či slov. Volené veličiny nejsou nijak blíže specifikovány, proto se nejčastěji používají smyšlená relativní a intervalová hodnocení, např. „hvězdičkování“ kvality rekonstrukce, třídění na nesrozumitelné a lépe srozumitelné apod. Existují dva základní druhy subjektivních metod [12]:

- Metoda relativní preference (priority),
- metody číselného hodnocení kvality.

Detailnější rozbor těchto metod je v [12]. V našem případě budeme hodnotit jak kvalitu, tak srozumitelnost, pomocí relativní škály jedné až pěti hvězdiček (pět hvězdiček znamená nejlepší, pro ucho prakticky shodný vjem jak pro originální nahrávku).

Objektivních metod existuje celá řada, kdy je v každém případě porovnávána jistá měřitelná fyzikální veličina, vypočtená z průběhu signálu. Příklady těchto metod jsou:

- *Artikulační index,*
- *SNR (Signal-to-Noise Ratio),*
- *Metoda WSS (Weighted-Spectral Slope),*
- *Metody založené na LP analýze,*
- *Metoda PSQM (Perceptual Speech Quality Measurement).*

Všechny tyto metody a další jsou detailně rozebírány v [6]. Pro naše účely jsou důležité následující tři metody:

SNR – odstup signálu od šumu

Parametr SNR určuje pro každý segment vztah [6]

$$SNR_{\text{dB}} = 20 \cdot \log_{10} \frac{\sum_n s^2(n)}{\sum_n [s(n) - \hat{s}(n)]^2}, \quad (7.1)$$

kde $s(n)$ je původní signál nahrávky a $\hat{s}(n)$ je jeho rekonstrukce. Tato metoda dává jedno číslo pro celý segment, které vyzníkne na základě rozdílnosti hodnot sobě odpovídajících vzorků. Jelikož při rekonstrukci dochází k fázovému posunu signálu danou především aplikací automatického markování period TD-PSOLA, nabývá SNR malých (tj. nepříznivých) zejména pro znělé hlásky, které jsou jinak i velmi kvalitně rekonstruované. Metodu SNR jsem do výsledného hodnocení nezařadil, protože její výsledky nepodávali relevantní informaci (např. znělé souhlásky byly vždy negativně klasifikovány a nezáleželo na podobnosti jejich podobnosti s původním průběhem).

Rozdílnost autokorelací (ACFx)

Výše uvedené důvody mne přiměly ke zkoumání podobnosti jiných parametrů. Jako vhodný nástroj se ukázaly být autokorelace obou signálů. Oproti vlastnímu průběhu signálu totiž nejsou vzájemně fázově posunuty, a tak rozdílnost jejich průběhů má vyšší vypovídací hodnotu o kvalitě rekonstrukce. Rozdíl autokorelací je dán vztahem

$$\varepsilon_{\text{ACF}} = \sqrt{E} \cdot \frac{\sum_n |\text{acf}(s(n)) - \text{acf}(\hat{s}(n))|}{\sum_n (|s(n)| + |\hat{s}(n)|)}, \quad (7.2)$$

kde $\text{acf}()$ značí normalizovanou autokorelaci a E pak energii signálu v segmentu. Vztah (7.2) jsem odvodil na základě experimentů a porovnání konkrétních výsledků se subjektivním posouzením kvality pro konkrétní hodnoty. Vynásobením odmocninou energie signálu jsem zavedl kvůli zvýhodnění znělých segmentů, které by jinak měly hodnotu ε_{ACF} podstatně nižší v porovnání s nízkoenergetickými neznělými segmenty. Jelikož hodnota samotného ε_{ACF} nabývala hodnot v řádech 10^{-4} až 10^{-3} , bylo nutné ji upravit na hodnotu, která splňuje následující dvě podmínky:

1. nabývá pro hodnotícího přehledných hodnot,
2. je nezávislá na délce okna.

Vztah (7.2) jsem tedy upravil na

$$ACFx = 2 \cdot 10^4 \frac{\varepsilon_{\text{ACF}}}{N} = 2 \cdot 10^4 \frac{\sqrt{E}}{N} \cdot \frac{\sum_n |\text{acf}(s(n)) - \text{acf}(\hat{s}(n))|}{\sum_n (|s(n)| + |\hat{s}(n)|)}, \quad (7.3)$$

kde N je počet vzorků segmentu (délka okna `win`). Hodnota $ACFx$ již nabývá „pěkných“ hodnot od 0 teoreticky do nekonečna, v praxi se pro jednotlivé segmenty

pohybuje v intervalu (0; 10) (čím více, tím horší rekonstrukce). Pro celou nahrávku je pak počítána střední hodnota \overline{ACFx} . Aby se zabránilo zvýhodňování nahrávek s výskytem pauz v řeči, které mají velmi malou hodnotu $ACFx$ a tím pádem snižují hodnotu tohoto ukazatele, je tato střední hodnota počítána jen z hodnot nad stanoveným prahem. Onen práh je nalezen následovně:

Hodnoty $ACFx$ pro všechny segmenty jsou seřazeny vzestupně a je vybrána hodnota prvku na pozici 0,95 (kvůli odstranění vlivu nejvyšších hodnot). Hledaný práh pak tvoří 20 % hodnoty onoho prvku.

Kvalita rekonstrukce se pak dá určit dle tabulky 7.1.

Tab. 7.1: Úroveň kvality rekonstrukce dle \overline{ACFx}

\overline{ACFx}	Úroveň kvality rekonstrukce
méně než 1	Velmi kvalitní
1–2	Kvalitní
2–3	V postačující kvalitě
3–4	Zhoršená kvalita
4–5	Špatná kvalita
5 a více	Velmi špatná kvalita

Metoda váhované strmosti spektra (WSS)

Tato metoda je založena na analýze spektra ve 36 překrývajících se frekvenčních pásmech. Každé pásmo je stanoveno dle tabulky kritických pásem z psychoakustiky. Výpočet probíhá ve třech krocích a výsledná mezirámcová míra spektrální vzdálenosti se spočítá jako [6]

$$d_{WSSM}(|S(\omega; m)|, |\hat{S}(\omega; m)|) = K + \sum_{k=1}^{36} w_{k,m} \left[|S(\omega; m)| - |\hat{S}(\omega; m)| \right]^2, \quad (7.4)$$

kde S , \hat{S} jsou spektrální funkce originálu, resp. rekonstrukce, k značí pořadí frekvenčního pásma, m pořadí segmentu, člen K je parametr vztahující se k celkové hladině akustického tlaku originální nahrávky a $w_{k,m}$ je soubor vah pro každé pásmo a segment. Tato metoda je v její plné podobě náročná na implementaci (např. stanovení vah), proto jsem účely této práce použil její základní myšlenku pro určení druhého parametru WSS pro objektivní zhodnocení nahrávek. Tou myšlenkou je porovnávání spekter originálu a rekonstrukce. Zajímavé výsledky podávalo běžné porovnání spekter mezi jejich odpovídajícími vzorky. Parametr WSS je počítán dle vztahu

$$WSS = 4 \cdot \log_{10} \left(2 \cdot 10^4 \sqrt{E} \cdot \frac{\sum_n \left[|S(n)| - |\hat{S}(n)| \right]^2}{\sum_n \left[|S(n)| \cdot |\hat{S}(n)| \right]} \right). \quad (7.5)$$

Veličina WSS byla upravena podobným způsobem jako $ACFx$, aby dosahovala reprezentativních hodnot v intervalu $(0; 10)$. Zde bylo nutné ještě hodnotu zlogaritmovat, aby bylo dosaženo hodnot ve stejném rozmezí, jako u $ACFx$. Stejně tak byly upraveny podmínky pro výpočet střední hodnoty \overline{WSS} . Kvalita rekonstrukce se zde dá odhadnout dle tabulky 7.2.

Tab. 7.2: Úroveň kvality rekonstrukce dle \overline{WSS}

\overline{WSS}	Úroveň kvality rekonstrukce
méně než 3	Velmi kvalitní
3–5	Kvalitní
5–7	V postačující kvalitě
7–8	Zhoršená kvalita
8–10	Špatná kvalita
10 a více	Velmi špatná kvalita

Obecně se dá říci, že segment, pro který je hodnota $ACFx$ nebo WSS větší než 10, je nekvalitně zrekonstruován. I přes provedené úpravy uvedených vztahů se nepodařilo zcela vymýtit obecné znevýhodnění znělých segmentů, kdy tyto veličiny nabývají i tak vyšších hodnot. Jejich odstup od hodnot u neznělých segmentů se ale výrazně snížil. Aby měly střední hodnoty \overline{ACFx} a \overline{WSS} vypovídající hodnotu, je třeba analyzovat nahrávku obsahující několik slov, včetně pauz. U mírně zašuměných nahrávek měly zpravidla nižší hodnoty.

7.2.2 Výsledky práce simulovaných kodérů

Kodéry byly testovány pro různé nahrávky – hlásky, slova, věty, krátké úryvky ze zábavných vystoupení apod. Obecně by se dala úroveň kvality rekonstrukce slovně popsat následovně:

Kodér LPC při řádu analýzy $p = 10$ vykazoval velmi kvalitní rekonstrukci. Nejvíce o tom vypovídala zvuková stránka u znělých hlásek. Výborné rekonstrukce bylo dosaženo např. u hlásek a , o , u , h , p , l , r . Naopak hůře už dekodér zvládal hlásky t , $č$, v , a především $ř$. Snížením řádu kodéru se analogicky zhoršovala kvalita nahrávky. Obecně horší rekonstrukci vykazoval kodér při řádech $p \leq 6$.

Kodér s přeparametrizovanými ARMA modely podával rekonstrukci srovnatelnou s LPC řádu $p = 7$. Tvary průběhů jednotlivých rekonstrukcí se lišily dle podoby konkrétních vygenerovaných ARMA procesů. To nejvíce vadí na úrovni segmentů, kdy dochází k diskontinuitám ve zvukové podobě sousedních segmentů.

Ve zvukové spojitosti nahrávky se tento kodér nevyrovnal LPC kodéru. Naopak se ukázalo, že koeficienty se dokáží přizpůsobit drobnému zašumění nahrávky a při rekonstrukci takovéto nahrávky podával lepší výsledky, dokonce došlo k mírnému odšumění nahrávky. Počty koeficientů pro dosažení požadované přesnosti byly zpravidla menší než 10. U některých znělých hlásek s jednoduchým průběhem (např. *i*) postačily třeba i dva koeficienty.

Posouzení a porovnání kvality vybraných krátkých nahrávek jsou zaznamenány v tabulce 7.3. Pro každou nahrávku jsou v horním řádku údaje z LPC a v dolním z ARMA rekonstrukce. Všimněte si vlivu snížení řádu LPC rekonstrukce na hodnoty objektivních porovnávacích veličin (spodní část tabulky) vzhledem údajům pro řídké ARMA koeficienty.

Tab. 7.3: Porovnání kvality rekonstrukce simulovaných kodérů

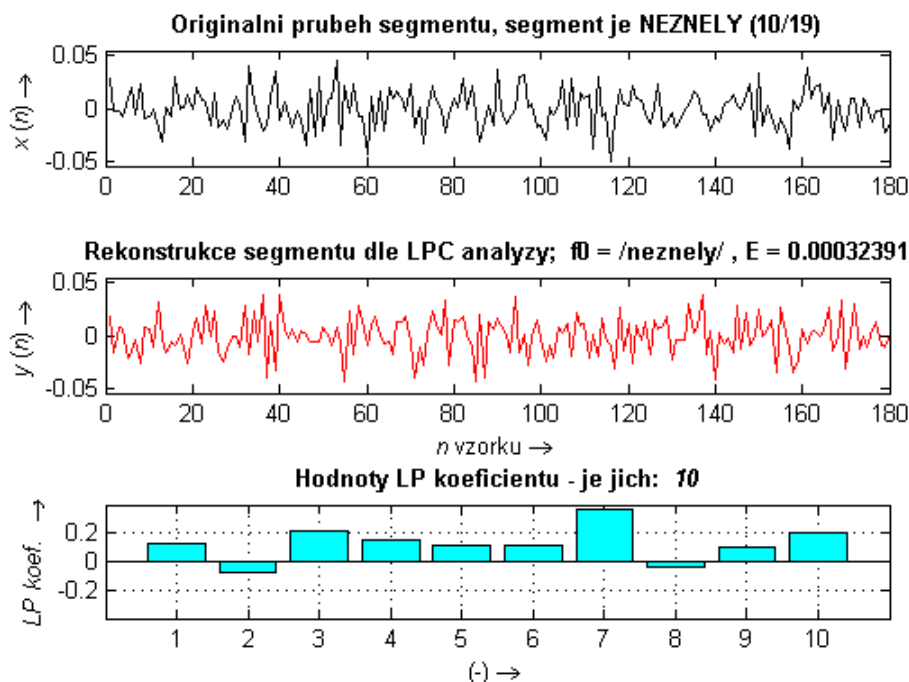
Název	Text v nahrávce	Prům. poč. koeficientů	Subjekt. porov.		Objekt. porov.	
			Kvalita	Srozum.	\overline{ACFx}	\overline{WSS}
nula1.wav	„Nula.“	LPC: 10	****	***	2,74	6,28
		ARMA: 8,06	***	***	4,32	7,06
policie1.wav	„Městská policie,...“ „...prosím.“	LPC: 10	*****	*****	0,96	3,78
		ARMA: 9,20	****	***	1,44	4,55
jednalz.wav	„Jedna.“ (ženský hlas)	LPC: 10	*****	*****	3,27	7,70
		ARMA: 5,61	***	****	4,65	8,45
horiii.wav	„Hoříí.“	LPC: 10	***	****	2,40	6,91
		ARMA: 7,03	***	****	3,16	7,80
tri1z.wav	„Tři.“ (ženský hlas)	LPC: 10	****	****	1,80	6,40
		LPC: 8	****	****	1,90	6,50
		LPC: 6	****	****	2,10	6,67
		ARMA: 7,67	****	****	1,91	6,65
humus.wav	„Humus.“	LPC: 10	***	***	1,39	4,00
		LPC: 8	***	***	1,65	4,45
		LPC: 6	**	**	2,92	4,95
		ARMA: 9,19	****	****	1,62	5,05

7.3 Obrazové ukázky výstupů kodérů

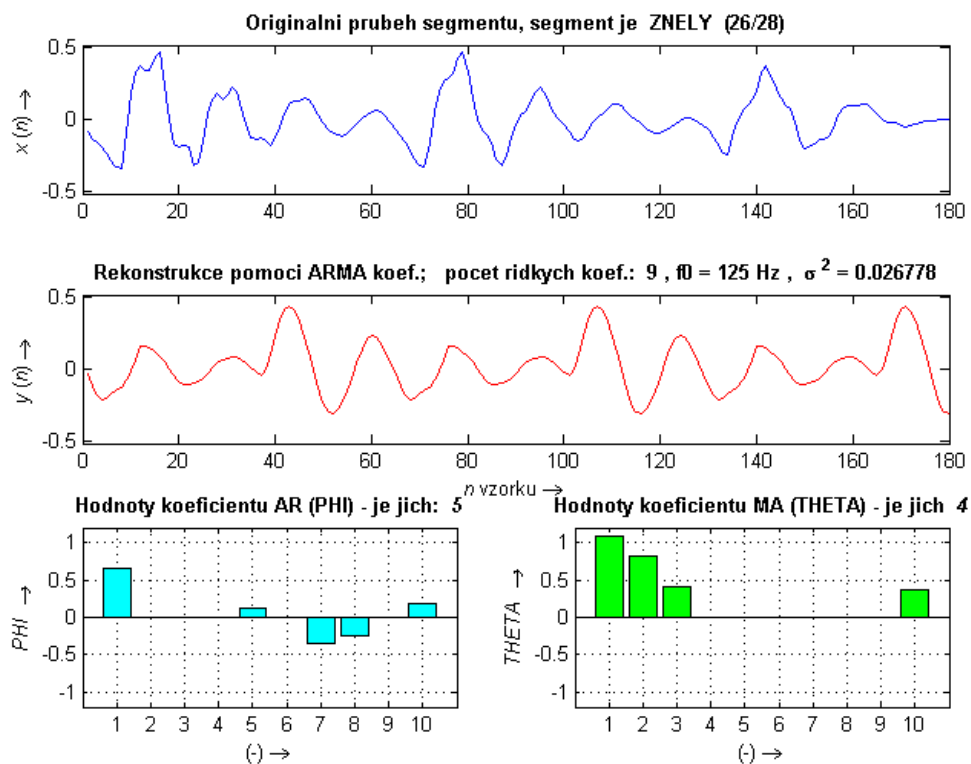
Zde jsou ukázky některých výstupů simulace kodérů v prostředí MATLAB. Jedná se o:

- Průběh originálního a rekonstruovaného neznělého segmentu pomocí LPC kodéru (Obr. 7.5)
- Průběh originálního a rekonstruovaného znělého segmentu pomocí ARMA kodéru (Obr. 7.6)
- Průběh celé originální nahrávky i její ARMA rekonstrukce (Obr. 7.7)
- Zobrazení originálního segmentu s jeho LPC i ARMA rekonstrukcí (Obr. 7.8)
- Ukázka autokorelací a spektrálních obálek znělého segmentu (Obr. 7.9)
- Průběh $ACFx$ a WSS pro nahrávku `policie1.wav` (Obr. 7.10)

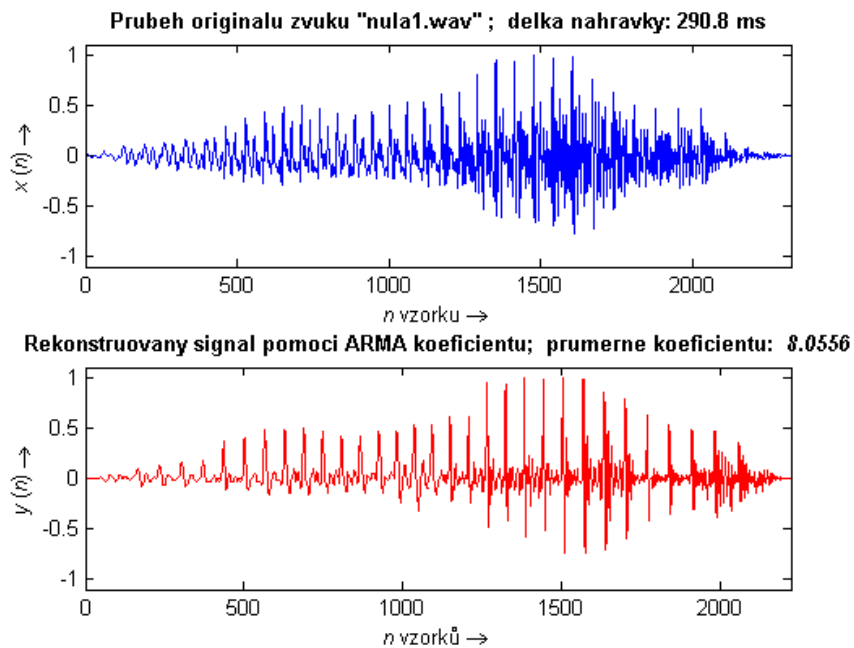
Tyto výstupy lze získat ze skriptů `DEMO_koder_LPC.m`, `DEMO_koder_ARMA.m` a `DEMO_porovnani.m`. Více informací o těchto demech je v přílohách v sekci A.2.



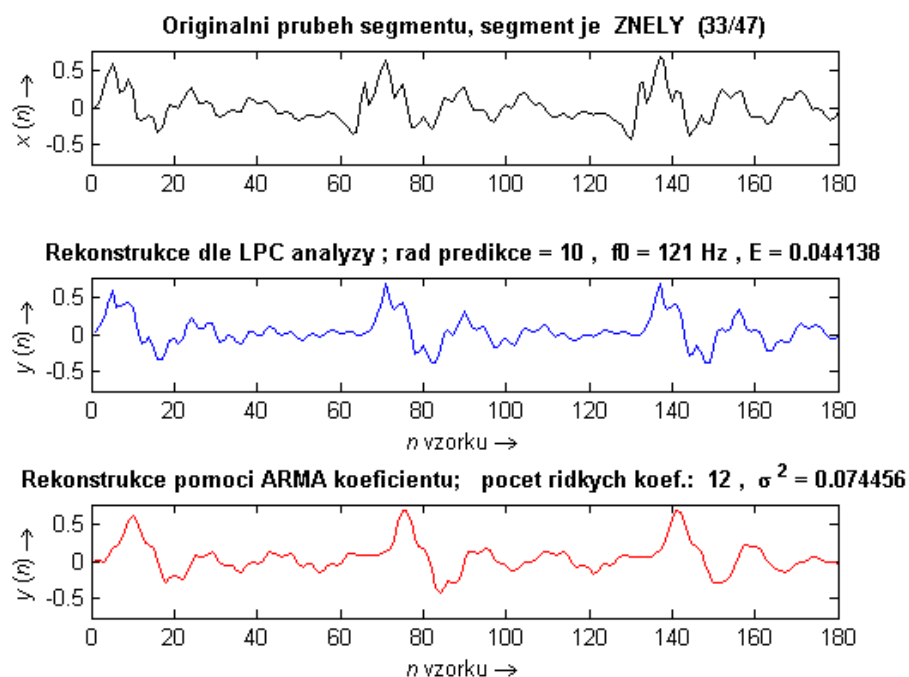
Obr. 7.5: Ukázka – kódování neznělého segmentu LPC kodérem



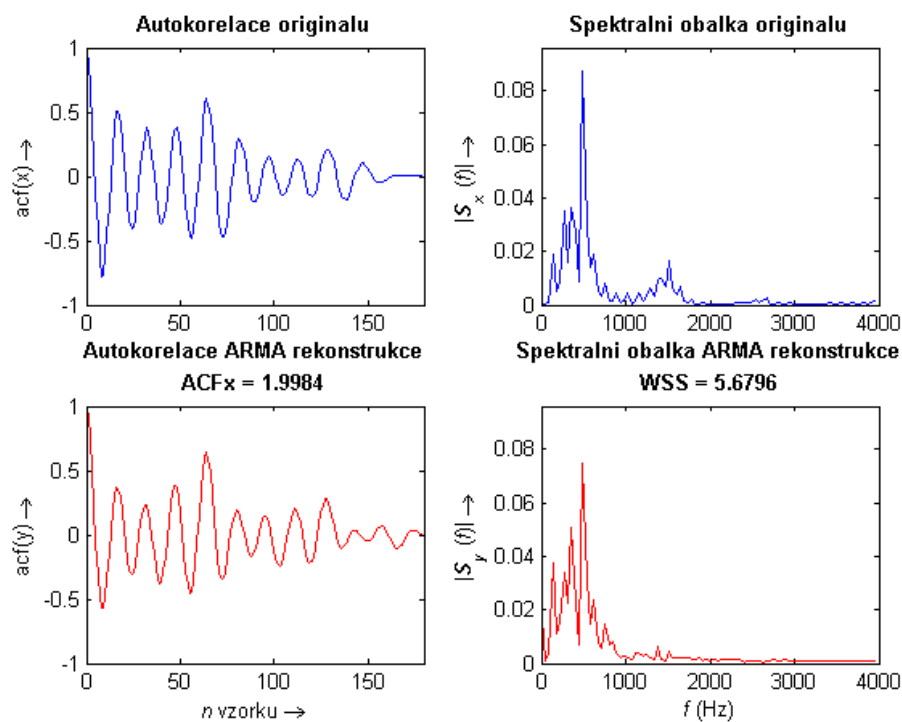
Obr. 7.6: Ukázka – kódování znělého segmentu ARMA kóděm



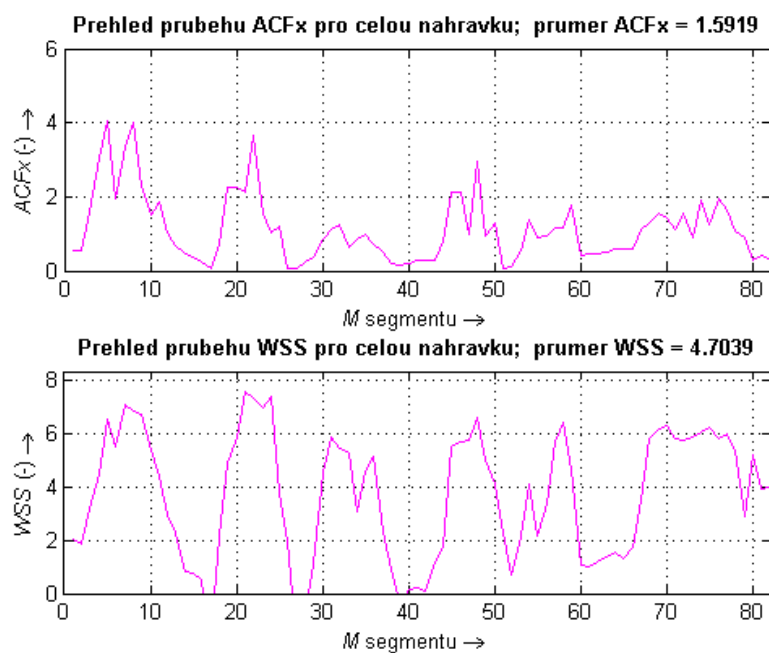
Obr. 7.7: Ukázka – průběh celé originální nahrávky i její ARMA rekonstrukce



Obr. 7.8: Ukázka – průběhy znělého segmentu originální, LPC i ARMA rekonstrukce nahrávky



Obr. 7.9: Ukázka – autokorelace a spektrální obálka znělého segmentu (ARMA)



Obr. 7.10: Ukázka – průběh $ACFx$ a WSS pro nahrávku `policie1.wav`

7.4 Výhody, nevýhody a možná vylepšení kodérů

Principy činností obou kodérů byly shledány jako schopné kódování a rekonstrukce řeči. Nejvýraznějším rozdílem je výpočetní náročnost obou algoritmů. Ta je nepoměřitelně vyšší pro přeparametrizované modely, protože toolbox *frames* využívá iterační algoritmy na bázi lineárního programování. Funkce v něm jsou uzpůsobeny široké škále použití (nejen pro ARMA modely v prostoru náhodných veličin), proto trvá celý výpočetní cyklus déle. Konkrétně pro segment dlouhý 180 vzorků trvalo nalezení řídkých koeficientů na počítači s dvoujádrovým procesorem o taktovací frekvenci 2,53 GHz okolo 10s, po zohlednění nutnosti uživatelské obsluhy se dostáváme průměrně na 30s. Na tuto dobu má významný vliv konstelace vstupních parametrů procedury `frappr` společně s konkrétním průběhem signálu – např. jen ve 2. iteračním kroku algoritmu BPA-4 trvaly výpočty od 0,7s do 22,5s. Rekonstrukce nahrávky o 82 segmentech trvala 8s. V případě LPC kodéru se hovoří o třech sekundách pro kompletní analýzu i rekonstrukci nahrávky čítající 592 segmentů (na tom samém počítači).

Shrnutí výhod a nevýhod obou typů kodérů vypadá následovně:

LPC kodér

- (+) nízká výpočetní náročnost nezávislá na typu segmentu
- (+) snadná rekonstrukce řeči
- (+) věrná podoba průběhu u znělých hlásek
- (–) málo plynulé přechody mezi znělými a neznělými úseky rekonstrukce – souvisí s jednoduchým buzením a rozdílným typem rekonstrukce obou typů segmentů
- (–) vysoké požadavky na kvantování všech koeficientů

Kodér s přeparametrizovanými ARMA modely

- (+) adaptace počtu a typu koeficientů na konkrétní průběh signálu
- (+) možnost adaptace kvantování koeficientů dle jejich významnosti
- (–) velmi vysoká výpočetní náročnost závislá na průběhu signálu v segmentu
- (–) složitější a nejednoznačná rekonstrukce řeči (náhodné procesy)

Možná vylepšení těchto kodérů jsou následující:

LPC kodér

Při zachování typu a jednoduchosti kodéru existuje již velmi málo možností, jak ho zdokonalovat. Například je před vlastní LPC analýzou možná úprava řečového signálu dle psychoakustického modelu. Tím se ale změní charakter koeficientů na PLP (*Perceptual Linear Predictive*). Mnoho prostoru je v možnostech kódování do bitového toku. Díky diferenčnímu kódování či vektorové kvantizaci dochází k výraznému snižování bitového toku, což ale opět mění typ kodéru na již zmiňované typy CELP, VSELP apod.

Co se týče dekódování a rekonstrukce, tak zde je zásadním nedostatkem jednoduché buzení typu jen impulzy/jen šum, čímž dochází k nesouvislým přechodům mezi znělými a neznělými hláskami. Tento problém se řeší zavedením dalších typů segmentů (přechodný, explozivní nástup, ...) a na základě nich jsou vytvářeny složitější impulzní sekvence v kombinaci se šumem. Tím se dosáhne vyšší kvality buzení a přirozenější rekonstrukce [5, 12].

Kodér s přeparametrizovanými ARMA modely

U tohoto kodéru se nabízí celá řada vylepšení v obou jeho částech. Při kódování je zde možno nastavit širokou škálu parametrů a hledáním nových kombinací těchto parametrů při pokusech na větším množství nahrávek bude možno naleznout vhodný algoritmus pro jejich správný výběr. Současná verze s toolboxem prakticky neumožňuje takovýto algoritmus hledat, protože vlastní hledání řídkého řešení je natolik časově náročné, že nedovoluje provádět rozsáhlé a automatizované analýzy. Dalším již hmatatelnějším vylepšením je odstranění šumu ze signálu před vlastní analýzou. Pokusy ukázaly, že šum komplikuje průběh signálu, což vede k nežádoucímu zrovnoměnění důležitosti jednotlivých ARMA koeficientů. Kódování koeficientů bude skýtat podbné možnosti a kompresní algoritmy jako u LPC kodéru.

U rekonstrukce z vygenerovaných ARMA procesů je zcela jistě možné zlepšit získání konkrétního průběhu, který vykazuje nejvěrohodnější zvukovou podobu původní řeči. Uvedená metoda sčítání autokorelace s částí průběhu realizace byla stanovena na základě experimentů s různými typy znělých hlásek, zakódovaných pomocí různého počtu koeficientů.

U obou typů dekodérů lze například ještě zavést plynulý přechod periody základního tónu řeči mezi segmenty, nebo ještě lépe zavést blok její predikce.

8 ZÁVĚR

Moderní metody kódování signálů směřují ke snižování počtu koeficientů pro model jejich popisu. Mezi tyto metody patří i přeparametrizované modely, které byly v této práci využity. Jejich princip spočívá v tom, že je model signálu popsán pomocí více proměnných než je potřeba, což umožní širší výběr a vhodně nastavené algoritmy jsou schopny vybrat jen ty nedůležitější z nich. Říká se tomu hledání řídkého řešení. V rámci této práce byl navržen a simulován kodér využívající tyto algoritmy pro aplikaci náhodných ARMA procesů při kódování řeči. Při kódování se ukázalo, že pro popis segmentu řeči o délce 22,5 ms stačí pro jeho popis 5 až 10 koeficientů – dle složitosti průběhu signálu. Stanovené odchylky modelů přitom činily 2 % pro znělé a 5 % pro neznělé segmenty. Komplikovanější již byla zpětná rekonstrukce řeči, kdy se samotná realizace ARMA procesu ukázala jako nepostačující k vymodelování znělého segmentu. Po řadě provedených experimentů bylo nakonec nalezeno řešení – modelování pomocí autokorelace ARMA procesu váhovaně sečtené s částí jeho vlastní realizace. Tím, že se jedná o modelování pomocí náhodných procesů, není vlastní průběh rekonstrukce pokaždé stejný, nicméně pro ucho jsou tyto odlišnosti nerozpoznatelné. Náhodné ARMA procesy s řídkými parametry tedy byly shledány jako vhodný nástroj pro modelování řeči.

Společně s tímto kodérem byl ještě simulován kodér pracující na základě LPC analýzy pro porovnání výstupů. LPC kodér vykazoval lepší subjektivní kvalitu rekonstrukce při použití řádu predikce 10. Pro řád predikce 7 již byly výstupní nahrávky srovnatelné. Nejvýraznější rozdíl obou kodérů je ve výpočetní náročnosti, ten s přeparametrizovanými modely vyžadoval podstatně více matematických operací pro získání řídkých koeficientů. Oba tyto kodéry jsou naprogramovány v prostředí MATLAB a jsou k dispozici na příloženém CD.

LITERATURA

- [1] ANDERSON, P. L., KAVALIERIS, L., MEERSCHAERT, M. M. : *Innovations algorithm asymptotics for periodically stationary time series with heavy tails*. Journal of Multivariate Analysis 99, pps. 94 - 116, 2008.
Dostupné z URL: <<http://www.stt.msu.edu/mcubed/innolimHT.pdf>>.
- [2] BROCKWELL, P. J. , DAVIS, R. A. : *Time Series: Theory and Methods (Springer Series and Statistics)*. 3rd edition, Springer Science + Business Media, LLC. New York, USA. ISBN 0-387-97429-5
- [3] BRUCKSTEIN A.M., DONOHO D.L., ELAD M. *From Sparse Solutions of Systems of Equations to Sparse Modeling of Signals and Images*. SIAM REVIEW, Society for Industrial and Applied Mathematics, Vol. 51 (2009), No.1, pp. 34–81.
- [4] ČERNOCKÝ, J. : *Číslíkové zpracování řeči: LPC*. Brno: Vysoké učení technické v Brně, Fakulta informačních technologií, 2003.
- [5] ČERNOCKÝ, J. : *Zpracování řečových signálů – studijní opora* Skriptum k předmětu. Brno: Vysoké učení technické v Brně, Fakulta informačních technologií, 2006.
- [6] DELLER, J. R. , HANSEN, J. H. L. , PROAKIS, J.G: *Discrete-Time Processing of Speech Signals (IEEE Press Classic Reissue)*. Wiley-IEEE Press; Reprint edition, 1999. ISBN 0780353862
- [7] GOLD, B., MORGAN, N. : *Speech and audio signal processing: Processing and perception of speech and music*. John Wiley & Sons, Inc., New York 2000, ISBN 0-471-35154-7
- [8] CHEN, S.S., DONOHO, D.L., SAUNDERS, M.A. : *Atomic decomposition by basis pursuit*. SIAM J. Sci Comput. 20 (1998), no.1, reprinted in SIAM Review, 43 (2001)
- [9] KOVAČEVIĆ J. , CHEBIRA A. : *An Introduction to Frames*. Now Publishers, 2008. ISBN 1-60198-068-X
- [10] MITRA, S.K. : *Digital Signal Processing: A Computer-Based Approach (Mcgraw-Hill Series in Electrical and Computer Engineering)* (Hardcover). Mcgraw-Hill College, University of California, Berkeley 1997, ISBN 0-070-42953-7.

- [11] RUDIN, W.: *Analýza v reálném a komplexním oboru*. 2. přepracované vydání, Academia, Praha, 2003.
- [12] SMÉKAL, Z.: *Číslíkové zpracování řeči*. Skriptum k předmětu. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009.
- [13] SMÉKAL, Z., VRBA, K., MALENOVSKÝ, V., SYSEL, P.: *Optimalizace ANSI-C kódu realizujícího kodek G.723.1 pro signálový procesor TMS320C6400*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2003.
- [14] SYDRAL, A., STYLIANOU, Y., GARRISON, L., CONKIE, A., SCHROETER, J.: *TD-PSOLA versus harmonic plus noisemodel in diphone based speech synthesis*. AT&T Labs-Research, New Jersey, USA, 2000.
- [15] VESELÝ, V.: *Functional modeling: The concept of overcompleteness*. Brno: Masarykova univerzita, Ekonomicko-správní fakulta, 2006.
- [16] VESELÝ, V., TONNER, J.: *Sparse Parameter Estimation in Overcomplete Time Series Models*. Austrian Journal of Statistics, Vol. 35, 2006, Number 2&3, 371-378. ISSN 1026-597X
- [17] VESELÝ, V.: *Hilbert-space techniques for spectral representation in terms of overcomplete bases*, Proceedings of the summer school DATASTAT'2001, Čihák u Žamberka (I. Horová, ed.), Brno: Masarykova univerzita, Přírodovědecká fakulta, Ústav aplikované matematiky, Česká republika, 2002. str. 259–273.
- [18] VESELÝ, V.: *From wavelets to frames: theory and applications*. Prezentace. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009.
- [19] VESELÝ, V.: *Sparse estimators with applications to time series forecasting and PM10 modeling*. Přednáška. Institut für Statistik, TU Graz, Österreich, 2008.
- [20] The MathWorks, Inc., *Matlab* [počítačový program]. Ver. 7.5.0.342 (R2007b), 2007.
- [21] ZAPLETAL, O.: *LPC kódování řeči*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. Vedoucí bakalářské práce Mgr. Pavel Rajmic, Ph.D.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

AMDF	Average Magnitude Difference Function
AR	Autoregressive – Autoregresivní
ARMA	Autoregressive Moving-Average – Model, sdružující AR a MA
BPA	Basis Pursuit Algorithm
BPA-4	BPA se čtyřmi iteračními kroky
DP	Dolní propust
FIR	Finite Impulse Response – Konečná impulzní odezva
FT	Fourierova transformace
HP	Horní propust
IA	Innovation Alorithm – Inovační algoritmus
IIR	Infinite Impulse Response – Nekonečná impulzní odezva
LSQ	Least Squares – Metoda aproximace pomocí nejmenších čtverců
LP	Linear Predictive – Lineární prediktivní (typ koeficientů)
LPC	Linear Predictive Coding – Lineární prediktivní kódování
LSF	Line Spectral Frequencies – Čárové spektrální frekvence
MA	Moving-Average – Pohyblivý průměr
PP	Pásmová propust
SNR	Signal-to-Noise Ratio – Odstup signálu od šumu
TD-PSOLA	Time Domain Pitch Synchronous Overlap Add – Metoda přičtení přesahu v časové oblasti se synchronizací periody základního tónu
VP	Vektorový prostor
WSS	Weighted-Spectral Slope – Váhovaná strmost spektra
WT	Waveletová transformace
$a_i^{(p)}$	LP koeficienty predikce řádu p

$A(z)$	Polynom stupně p obsahující LP koeficienty
$ACFx$	Míra difference autokorelací
\overline{ACFx}	Střední hodnota $ACFx$ pro celou nahrávku
$\mathbf{AR}(p)_t$	Autoregersní model řádu p
b_q	Charakteristické parametry modelu $\mathbf{MA}(q)_t$
\mathbf{b}	Matice sdružených momentů 2. řádu pro koeficienty AR, MA
$e(n)$	Chyba predikce
e_t	Náhodná chyba
E	Energie signálu v segmentu
f_{vz}	Vzorkovací frekvence
H_t	Separabilní uzavřený podprostor v L^2
$H(z)$	Přenosová funkce modelu artikulačního traktu
$\ell^2(\mathbb{R})$	Hilbertův prostor diskretních reálných posloupností
$L(t)$	Lorentzova funkce
L^2	Hilbertův prostor spojitých náhodných veličin
m	Počet sloupců matice \mathbf{T} (počet atomů ve slovníku)
$\mathbf{MA}(q)_t$	Model klouzavých průměrů řádu q
n	Počet řádků matice \mathbf{T} (délka atomu)
N	Počet vzorků v segmentu
p	Řád LPC analýzy
P	Počáteční nahodnocený řád koeficientů AR
\mathbf{P}	Matice ortogonální projekce
q	Řád koeficientů MA
Q	Počáteční nahodnocený řád koeficientů MA
r_1	Normalizovaný autokorelační koeficient

\mathbf{R}	Autokorelační matice
\mathbf{R}^+	Adjungovaný autokorelační operátor k \mathbf{R}
$s(n)$	Původní řečový signál
$\hat{s}(n)$	Rekonstruovaný řečový signál
$S(n)$	Spektrální funkce původního signálu $s(n)$
$\hat{S}(n)$	Spektrální funkce rekonstruovaného signálu $\hat{s}(n)$
T_0	Perioda základního tónu (tzv. pitch perioda)
\mathbf{T}	Matice atomů (slovník)
\mathbf{T}^+	Adjungovaný operátor k \mathbf{T}
V/UV	Symbol označující znělost-neznělost hlásky
WSS	Spektrální diference
\overline{WSS}	Střední hodnota WSS pro celou nahrávku
$\hat{\mathbf{x}}$	Odhad vektoru \mathbf{x}
\mathbf{x}^\perp	Ortogonální průmět \mathbf{x}
$\langle x, y \rangle$	Skalární součin dvou prvků x, y vektorového prostoru
$\ \mathbf{x}\ $	Norma prvku vektorového prostoru
X_t	Soubor hodnot veličiny $x(n)$ v Hilbertově prostoru
y_t	Hodnoty vzorků zkoumané veličiny metodou $\mathbf{AR}(p)_t$
Z_t	Soubor hodnot nekorelovaného šumu v Hilbertově prostoru
ε_{ACF}	Diference autokorelací
θ_k	Jednotlivé řídké koeficienty typu MA
Θ	Soubor všech koeficientů typu MA
ξ	Soubor neznámých parametrů modelu
ξ_j	Soubor řídkých parametrů modelu (váhy)
σ	Rozptyl Gaussovského bílého šumu

φ_i	Vektory generující frame
ϕ_j	Jednotlivé řídké koeficienty typu AR
Φ	Frame
Φ	Soubor všech koeficientů typu AR
ω_i	LSF koeficienty
d1	Vzdálenost začátku první celé periody od začátku segmentu
d2	Vzdálenost začátku poslední periody od konce segmentu
frappr	Procedura zastřešující hledání řídkého řešení v přeparametrizovaném systému
order	Řád LPC analýzy
overlen	Délka překryvu segmentu
win	Délka segmentu (okna)
window	Váhovací okno pro sestavení nahrávky
ZeroTolr	Procentuální hranice nulové tolerance pro frappr

SEZNAM PŘÍLOH

A	Přiložený software	69
A.1	Seznam adresářů a souborů	69
A.2	Dema a spustitelné skripty	71

A PŘILOŽENÝ SOFTWARE

Přílohou této práce je jeden CD disk s elektronickou verzí textu ve formátu PDF pod názvem DP_Zapleta10.pdf a soubory potřebnými k simulaci zmiňovaných kódérů v prostředí MATLAB.

Verze použitého MATLABu: 7.5.0.342 (R2007b)

Verze přiloženého toolboxu *frames*: 2009

A.1 Seznam adresářů a souborů

Adresáře:

„/mat_soubory“

Adresář s *.mat soubory s parametry pro rekonstrukci řeči pomocí ARMA procesů. Soubory byly vytvořeny pomocí spustitelných skriptů pro nalezení řídkých ARMA koeficientů (viz příloha A.2). Dema kódérů si potom automatizovaně načítají tyto soubory.

„/funkce“

Adresář obsahuje funkce *.m, které jsou využívány hlavními skripty pro fungování kódérů.

„/zvuky“

Adresář obsahuje nahrávky ve formátu *.wav, určené pro kódování.

„/Frames“

Adresář s toolboxem *frames*, včetně podsložky „/Atoms“ – mateřské funkce atomů.

Soubory z adresáře „/funkce“:

*(soubory označené * jsou úzce vzpjaty s toolboxem frames a byly vytvořeny doc. Veselým a jeho kolektivem)*

acf.m*

Výpočet autokorelace + její normalizace a výběr její pravé poloviny.

ALL_rekonstrukce_ARMA.m

Funkce zastřešující funkci dekodéru z přeparametrizovaných ARMA modelů pro celou nahrávku.

`analiza_parametru.m`

Funkce, která zastřešuje kompletní analýzu nahrávky pro LPC kódér (segmentace, detekce znělosti, získání LP koeficientů, základní periody apod.).

`armagen.m*`

Funkce pro relaizaci ARMA procesu.

`armasim.m*`

Funkce pro relaizaci dopředné predikce ARMA procesu.

`detekce_znelosti.m`

Detekce znělosti segmentů.

`energie.m`

Výpočet energie segmentů.

`errors_ARMA.m`, `errors_LPC.m`, `errors_porovnani.m`

Funkce zajišťující korektnost nastavitelných parametrů spustitelných dem.

`f0_amdf.m`

Funkce detekující periodu základního tónu řeči metodou AMDF.

`filtr_vzork_8khz.m`

Filtrace vstupního signálu a převzorkování na $f_{vz} = 8$ kHz.

`ia.m*`

Inovační algoritmus.

`lpc_koef.m`

Zjištění LP koeficientů zvoleného řádu pro segmenty.

`norm_autokor_koef.m`

Výpočet normalizovaného autokorelačního koeficientu pro segmenty.

`obj_chyby.m`

Výpočet objektivních rozdílových parametrů signálů.

`period_mark.m`

Funkce zajišťující markování period (synchronizace period v rámci celé nahrávky – TD-PSOLA).

`pocet_n0.m`

Funkce zjišťující počet průchodů signálu v segmentu nulou.

`rekonstrukce_ARMA.m`

Funkce podřízená funkci `ALL_rekonstrukce_ARMA.m` realizující rekonstrukci dle ARMA pro jeden segment.

`rekonstrukce_LPC.m`

Funkce zastřešující funkci dekodéru LPC pro celou nahrávku.

`segmentace.m`

Rozdělení nahrávky na menší úseky (segmenty).

`vyhlazeni.m`

Funkce realizující vyhlazení návazností period v rámci rekonstrukce nahrávky.

A.2 Dema a spustitelné skripty

Nejdůležitějšími soubory na CD jsou tři dema ohledně kodérů a dva skripty na výrobu zdrojových `*.mat` souborů. Ve všech případech mají tyto spustitelné skripty a dema v záhlaví nastavitelné parametry (viz níže). Zde uvádím ty parametry, pro něž byl kodér vyladěn a jsou nastaveny defaultně:

`win = 180;` – délka segmentu (180 vzorků),
`overlen = 60;` – délka překryvu segmentů (60 vzorků, tj. 33%),
`order = 10;` – řád LPC analýzy,
`window = 0;` – váhovací okno pro sestavení nahrávky (lineární tlumení v oblasti překryvu).

`DEMO_koder_LPC.m`

Toto demo je spustitelný skript (ne funkce), který realizuje kompletní analýzu nahrávky pro účely LPC kódování + následnou rekonstrukci. Součástí jsou grafické výstupy i zvukové ukázky. Nastavitelné parametry jsou přehledně specifikovány a vysvětleny v horní části kódu skriptu. Toto demo má následující módy:

`show_segmenty = 0;` – Skript provede na pozadí analýzu a rekonstrukci nahrávky, výstupem je přehrávání originální a rekonstruované nahrávky + zobrazení jejich grafických průběhů.

`show_segmenty = 1;` – Bude se jednat o analýzu s krokováním po segmentech (s ukázkami průběhů segmentů, koeficientů, hodnot parametrů) + samozřejmě výstup celkové nahrávky.

`vypocet_chyb = 0;` – Proběhne jen rekonstrukce.

`vypocet_chyb = 1;` – Budou navíc počítány objektivní rozdílové parametry a zobrazeny patřičné grafy.

Jako vstupní zvuky se dají použít všechny soubory v adresáři „/zvuky“.

Možnými výstupy dema jsou grafy a hodnoty viz obrázky 7.5, 7.7, 7.9 a 7.10.

`DEMO_koder_ARMA.m`

Toto demo je spustitelný skript (ne funkce), který realizuje rekonstrukci nahrávky pomocí přeparametrizovaných ARMA modelů. Tento skript je pouze dekodérem! Je nutné, aby analyzovaná nahrávka měla již vytvořené `*.mat` soubory s parametry v adresáři „`mat_soubory`“. V praxi je pouze omezen výčet možných nahrávek, pro které lze spustit (ty, které jsem připravil). Nastavitelné parametry jsou i zde přehledně specifikovány a vysvětleny v horní části kódu skriptu. Existují zde stejné módy dle `show_segmenty` a `vypocet_chyb`.

Jako vstupní zvuky se dají použít zvuky s názvy:

`hasice`, `jedna1z`, `policie1`, `praha1`, `deset1-2`, `dve`, `tri1z`, `humus`, `nula1`, `horiii`.

Možnými výstupy dema jsou grafy a hodnoty viz obrázky 7.6, 7.7, 7.9 a 7.10.

`DEMO_porovnani.m`

Toto demo je spustitelný skript (ne funkce), který realizuje oba typy rekonstrukcí pro celou nahrávku. Výčet nahrávek je opět omezen jen na ty, které jsem připravil analýzou přeparametrizovanými modely. Toto demo v podstatě jen slučuje předchozí dvě dema do jedné přehledné formy. Opět má módy dle `show_segmenty` a `vypocet_chyb` a informace v horní části kódu skriptu.

Jako vstupní zvuky se dají použít jen zvuky s názvy viz `DEMO_koder_ARMA.m` (jen ty mají vytvořené `*.mat` soubory s ARMA koeficienty).

Možnými výstupy dema jsou grafy a hodnoty viz obrázky 7.8, dále pak obrázky 7.7, 7.9 a 7.10 s tím, že obsahují ještě další graf s výsledky druhého typu rekonstrukce.

`segmentace_pro_ARMA.m`

Skript, který vytváří `*.mat` soubor s nasegmentovanou nahrávkou + základními parametry, který je brán jako zdrojový pro skript `nalezeni_koeficientu_ARMA.m` a je využit i demy pro rekonstrukci. Informace pro obsluhu jsou v horní části kódu skriptu.

`nalezeni_koeficientu_ARMA.m`

Skript, který vytváří `*.mat` soubory s parametry pro rekonstrukci pomocí ARMA procesů. Skript využívá funkcí toolboxu `frames`. Analýza se zadává pro každý segment ručně a celkově trvá řádově desítky sekund pro jeden segment, nutná je uživatelská obsluha. Informace pro nasatevní parametrů a obsluhu jsou v horní části kódu skriptu.