

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

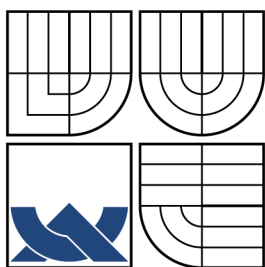
POTLAČOVÁNÍ ŠUMU V ŘEČOVÝCH SIGNÁLECH ZA
POMOCÍ ZPRACOVÁNÍ "ATRAKTORŮ"

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

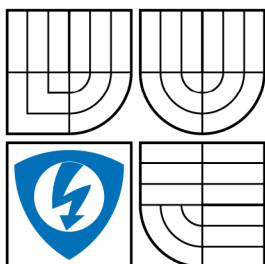
AUTOR PRÁCE
AUTHOR

TOMÁŠ LINHART

BRNO 2008



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

POTLAČOVÁNÍ ŠUMU V ŘEČOVÝCH SIGNÁLECH ZA
POMOCÍ ZPRACOVÁNÍ "ATRAKTORŮ"
SPEECH SIGNALS NOISE SUPPRESSION PROCESSING USING "ATTRACTORS"

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

TOMÁŠ LINHART

VEDOUCÍ PRÁCE
SUPERVISOR

ING. RADEK ZEZULA, PH.D.

BRNO 2008

ZDE VLOŽIT LIST ZADÁNÍ

Z důvodu správného číslování stránek

ABSTRAKT

Na řečový signál je nahlíženo jako na nelineární dynamický systém. Jako takový je potom převeden do vícedimensionálního fázového prostoru RPS, kde je použita vlastní filtrační metoda založena na sousedských vztazích časových řad zkoumaného signálu. K vytvoření vícedimensionálního fázového prostoru jsou použity metody zpoždění a falešných nejbližších sousedů.

KLÍČOVÁ SLOVA

Atraktor, multidimensionální systém, časové zpoždění, vytvořený fázový prostor, falešní nejbližší sousedé, časové řady, RPS.

ABSTRACT

Speech signal is being used in the meaning of nonlinear dynamic system. As such, it is transform to multidimensional phase space, where filtration method based on time series neighbors of analysed signal is used. For embedding phase space methods time delay and false nearest neighbors are applied.

KEYWORDS

Attractor, multidimensional system, time lag, reconstructed phase space, false nearest neighbors, time series, RPS.

LINHART T. *Potlačování šumu v řečových signálech za pomoci zpracování "atraktorů"*.
Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií.
Ústav telekomunikací, 2008. Počet stran 21. Počet stran s přílohou 28. Semestrální
práce. Vedoucí práce byl Ing. Radek Zezula, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Potlačování šumu v řečových signálech za pomoci zpracování "atraktorů"“ 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)

OBSAH

Úvod	9
1 Teoretický úvod	10
2 Postup vytvoření RPS	15
3 Metoda zpoždění	17
3.1 Volba rozumného časového zpoždění τ pomocí zobrazení atraktoru . .	17
3.2 Volba rozumného časového zpoždění τ pomocí vzájemné informace . .	20
4 Volba vhodného rozměru dimense	21
4.1 Box-assisted approach	21
4.2 False nearest neighbours (FNN)	22
5 Rekonstrukce vícedimensionálního fázového prostoru	25
6 Implementace v prostředí Matlab	26
6.1 Skript RPS.m	26
6.2 Skript segmnetation.m	28
6.3 Skript delay.m	28
6.4 Skript FNN.m	29
6.5 Skript filtrovani.m	29
6.6 Skript SNRg.m	31
6.7 Skript SNRseg.m	31
7 Vyhodnocení výsledků	34
8 Závěr	37
Literatura	38
Seznam symbolů, veličin a zkratk	39
Seznam příloh	40

SEZNAM OBRÁZKŮ

1.1	Souřadná soustava trojrozměrného fázového prostoru vybraného dynamického systému.	11
1.2	Lorenzův atraktor.	13
3.1	2D atraktor zobrazený pro různá časová zpoždění opakované hlásky "u".	18
3.2	Znázornění časového zpoždění na lidském magneto-kardiogramu. . .	19
4.1	Vzorky roztrženy do 16 boxů ve dvojdimensionální mřížce, za použití box-assisted metody.	21
4.2	Srovnání doby vyhledávání pro různé metody. Nahoře jsou použita dvojdimensionální data, dole trojdimensionální.	23
4.3	Podíl falešných nejbližších sousedů jako funkce rozměru vytvářené dimense.	24
6.1	Algoritmus sestavování matic RPS pro jednotlivé segmenty. Je součástí skriptu <code>RPS.m</code>	27
6.2	Příklad sestavení rámce z vyfiltrovaných hodnot matice RPS.	27
6.3	Výstup skriptu <code>FNN.m</code> . Podíl falešných nejbližších sousedů jako funkce rozměru vytvářené dimense.	29
6.4	Ukázka ze skriptu <code>filtrovani.m</code>	30
6.5	Výpočet <i>globálního</i> SNR ve skriptu <code>SNRg.m</code>	31
6.6	Výpočet <i>segmentálního</i> SNR ve skriptu <code>SNRseg.m</code>	33
7.1	Výsledky SNR pro aditivní Gaussův bílý šum. Počítáno pomocí skriptu <code>SNRg.m</code> 6.6.	35
7.2	Výsledky SNR pro aditivní Gaussův bílý šum. Počítáno pomocí skriptu <code>SNRseg.m</code> 6.7.	36

SEZNAM TABULEK

7.1	Naměřené výsledky při použití skriptu SNRg.m.	34
7.2	Naměřené výsledky při použití skriptu SNRseg.m.	35

ÚVOD

Tato práce se zabývá analýzou nelineárních metod potlačení šumu v řečových signálech, na který je nahlíženo jako na nelineární dynamický systém. Jako takový, na něj nelze aplikovat klasické metody potlačování šumu spočívající v analýze spektrálního rozložení signálu a šumu za účelem jejich vzájemné separace a filtrace. Iregularní signály z nelineárních zdrojů totiž působí v širokém pásmu spektra a proto je složité identifikovat jejich komponenty ve spektru a přitom je nezaměnit za složky šumu.

Proto se využívají speciální metody, které zohledňují nelineární strukturu signálů. Ty spočívají v převedení nelineárních dynamických systémů do vícedimenzionálního stavového prostoru RPS (Reconstructed Phase Space), ve kterém se aproximují geometrické struktury atraktorů. Atraktory popisují chování dynamického nelineárního systému v čase a mohou mít různou podobu. V této práci je do podoby atraktoru převáděn řečový signál. Ze sousedských vztahů jednotlivých opožděných vektorů daného signálu tvořících atraktor lze získat informace, na jejichž základě se potlačí šum v tomto řečovém signálu.

K vytvoření vhodného stavového prostoru RPS (atraktoru) musí být nejprve získány dva hlavní vstupní parametry. Jedná se o vhodné časové zpoždění signálu a minimální dostatečnou dimenzi prostoru. Hodnoty těchto proměnných se ukázaly jako klíčové při tvorbě optimálního RPS a optimální RPS je zase rozhodující pro získání nejlepších možných výsledků při zvýrazňování řeči z nelineárního dynamického systému. Pokud jsou tyto parametry nalezeny, může se vytvořit samotný rekonstruovaný fázový prostor RPS. V něm teprve se rozložený zarušený signál vyfiltruje a vhodnou metodou zpětného rozložení RPS převede do původní podoby.

Potřebný teoretický základ i samotná realizace v programovacím prostředí Matlab jsou uvedeny v následujících kapitolách.

1 TEORETICKÝ ÚVOD

Základními prostředky zpracování signálů jsou v dnešní době **lineární metody**. Jejich dominantní role je dána historickým vývojem, neboť jako vývojově starší jsou všeobecně zavedeny a využívají běžných matematických prostředků. Jsou jednodušší, což naopak umožňuje dále prohlubovat standardní metody analýzy a syntézy příslušných algoritmů srozumitelným způsobem. Díky jednoduchému popisu je vlastní činnost a případně její změny v čase pro uživatele průhledná a aplikace tak dobře přístupné. Třída lineárních metod je definována platností principu superpozice; díky této definici je třída jasně vymezena, poznatky a přístupy lze v jejím rámci dobře zobecňovat a metody přehledně klasifikovat.

Naproti tomu **nelineární metody** a odpovídající systémy jsou vymezeny jen negativně - neplatí u nich princip superpozice. To je ovšem vymezení velmi vágní, jehož důsledkem je, že není možno formulovat mnoho obecně platných zákonů a pravidel, které by se vztahovaly na všechny nelineární systémy. Následně není možno ani vytvořit obecné metody popisu, třídění, analýzy a syntézy nelineárních metod. Přitom se ukazuje, že je mnoho praktických úloh různého charakteru, u nichž lineární metody selhávají z principiálních důvodů a naopak aplikování nelineárních metod na tyto úlohy často přináší překvapivě optimistické výsledky. Ukázkami mohou být časté a zdánlivě jednoduché případy, jako je potlačování neaditivního šumu, např. multiplikativního šumu, zpracování signálů generovaných nebo ovlivněných systémy s nelineárními charakteristikami, nebo hledání vztahů mezi signály, jejichž spektra se nepřekrývají.

V této práci se využívají pro zvýraznění řečového signálu nelineární metody vycházející z teorie chaosu. Z té je zde použito mnoho termínů a pro pozdější pochopení popisovaných metod a postupů bude vhodné je předem vysvětlit (převzato z [11], [12] a [2]):

Dynamický systém

Dynamický systém je definován pomocí dynamických podmínek, které popisují změnu tohoto systému v čase. Stav systému v libovolném časovém okamžiku je potom popsán vektorem, který celý leží ve stavovém prostoru. Dynamické podmínky jsou většinou zadány soustavou diferenciálních rovnic, které popisují změnu stavového vektoru v čase. Změna stavu dynamického systému se děje provedením těchto diferenciálních rovnic a nahrazením starého stavového vektoru vektorem novým. Dynamický systém může být:

deterministický - lze jej poměrně přesně popsat.

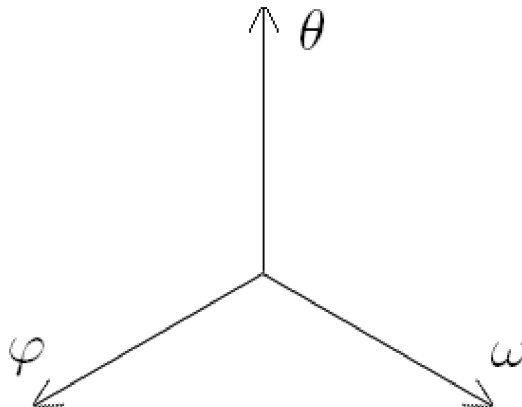
stochastický - jsme odkázáni pouze na statistické vlastnosti takového systému (například střední hodnota, disperse, směrodatná odchylka, centrální moment a jiné).

Stavová proměnná

Za dynamický systém budeme považovat systém, který je popsán konečnou množinou stavových proměnných. Stavové proměnné jsou vybrané fyzikální veličiny, které svou proměnlivostí v čase jednoznačně popisují vývoj systému. To znamená, že okamžitý stav systému plně určuje jeho vývoj. Jako příklad stavových proměnných může být úhlová výchylka φ , úhlová rychlost ω a fáze budící síly θ .

Fázový prostor

Hodnoty stavových proměnných lze zobrazit bodem ve fázovém prostoru. Fázový prostor je lineární prostor, jehož každý rozměr reprezentuje jednu stavovou proměnnou systému. Má tedy tolik rozměrů, kolik má daný systém stavových proměnných. Libovolný dynamický systém lze tedy znázornit a popsat pomocí abstraktního prostoru stavů, který se nazývá fázový prostor. Právě metoda vytvoření vhodného stavového prostoru je v této práci popsána.



Obr. 1.1: Souřadná soustava trojrozměrného fázového prostoru vybraného dynamického systému.

Atraktor

Jestliže se ponechá daný dynamický systém vyvíjet dostatečně dlouho, vzniká ve fázovém prostoru křivka (pokud je čas spojitý) nebo množina bodů stavů (pokud je čas diskrétní), která zvýrazňuje určitou strukturu. Tato struktura se nazývá atraktor, z anglického slova attract - přitahovat. Množina stavů vykreslených ve fázovém prostoru, které vedou ke stejnému atraktoru, se nazývá *oblast přitahování atraktoru*. Zjednodušeně se dá říci, že atraktor je konečný stav systému.

Atraktory se rozdělují do několika tříd:

Atraktorem jsou pevné body. Nejjednodušší případ. Systém se v nekonečném čase ustálil v nějakém stabilním stavu a v podstatě už nejde o dynamický systém. Příkladem může být kyvadlo, které se vlivem odporu vzduchu a odporu ložisek zastaví v nejnižším bodě své dráhy.

Atraktorem jsou periodické, kvaziperiodické body. Jde také o jednoduchý případ. Systém se ustálil tak, že osciluje mezi několika stavy. Příkladem je těleso, které se na své cestě vesmírem dostane do blízkosti velmi hmotného tělesa. Po určitém čase se pohyb tohoto tělesa ustálí na eliptické dráze.

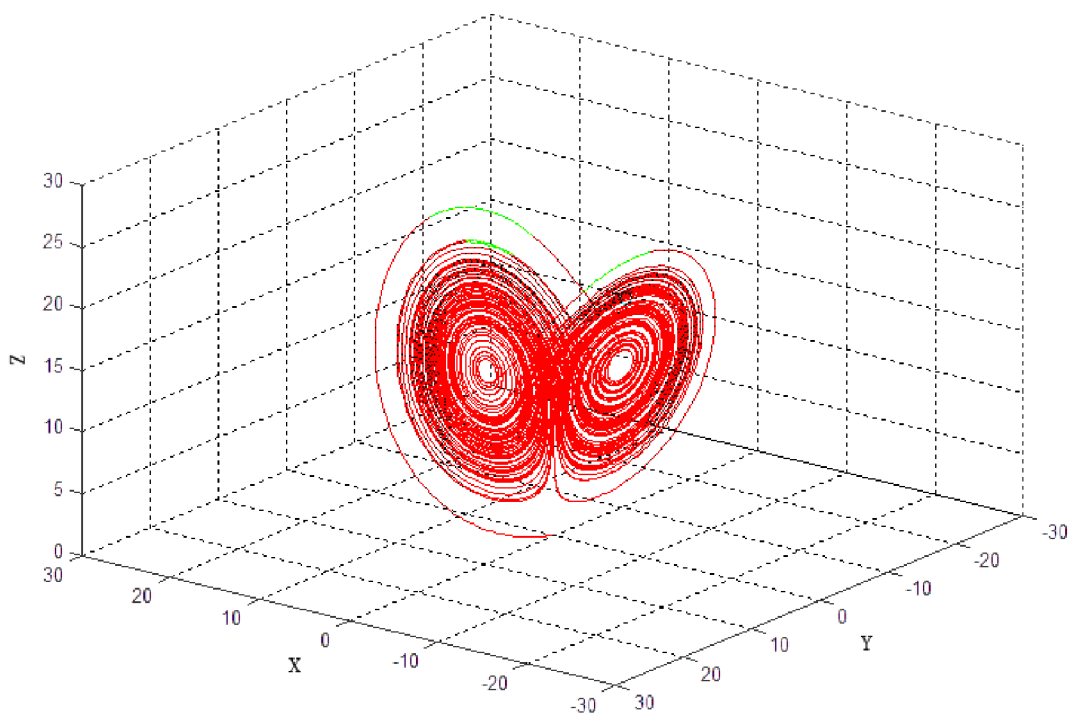
Chaotický atraktor. Tento atraktor nelze v podstatě nijak dopředu předpovědět. To je způsobeno tím, že je systém velmi citlivý na počáteční podmínky. Chaotičnost v tomto případě neznamena náhodnost, ale s ohledem na počáteční podmínky se jedná o deterministický systém. Příkladem může být koule postavená na vrcholku jehlanu. Jakýkoliv vnější podnět způsobí, že koule tento stav opustí a dostane se do některého atraktoru (místo pod jehlanem). Tento atraktor nelze předpovědět, protože bez zásahu do měření nelze zjistit počáteční podmínky.

Podivný atraktor. Nejzajímavější případ atraktoru. Tento typ atraktoru vzniká, je-li systém popsán minimálně třemi diferenciálními rovnicemi. Takový systém může mít velmi komplikovaný atraktor, který sice bude chaotický, ale přesto bude vykazovat určité pravidelnosti. Termín podivný atraktor není ještě přesně matematicky definován, ale považuje se za něj takový atraktor, který vykazuje stejné vlastnosti, jaké mají fraktály (podivný atraktor je tedy fraktálem). Příkladem podivného atraktoru je známý Lorenzův atraktor na obr. 1.2.

Chaotické chování systému vede k atraktoru s neuzavřenou hranicí. Jednotlivé body ve fázovém prostoru od sebe příliš divergují a tedy ani po relativně dlouhé době, po kterou se systém vyvíjí, nejde pozorovat žádná opakující se strukturu atraktoru. Pokud ale po určité, dostatečně dlouhé době (její velikost závisí na počátečních podmínkách) křivka atraktoru začne opisovat strukturu, kterou lze jednoznačně ohraničit, potom lze chování systému předpovídat na libovolně dlouhou dobu. Toto je velice důležité zjištění, umožňující popisovat a odhadovat chování vybraných nelineárních systémů. Zavádí se zde tedy pojem **deterministický nelineární systém**, popřípadě **deterministický chaos**.

Deterministický chaos

Pod pojmem chaos se všeobecně rozumí takové chování, které je projevem absolutní a čisté náhody, není v něm tedy místo pro působení zákonitostí. Takového chování by bylo možno nazvat absolutně nekoherentní, kdy neexistují zákonité vazby mezi



Obr. 1.2: Lorenzův atraktor.

sousedními stavy (jak v prostoru tak v čase). Chaotické chování skutečných systémů v přírodě však charakterizuje termín "deterministický chaos". Jedná se o proces samoorganizace složitých systémů, kdy vznikají soustavy koherentních struktur chovajících se v souladu s přírodními zákony. Z hlediska jedné určité struktury má vývoj systému prvky náhodnosti, systém jako celek se však vyvíjí zcela zákonitě a tedy deterministicky. Tento jev lze nalézt v přírodních systémech všech možných forem - od fyzikálních, chemických (např. chemické reakce), ekonomických (např. Elliotova vlna na kapitálových trzích), tak i v biologických systémech (např. chování kolonie mravenců).

Lineární metody znázorňují všechny pravidelné struktury obsažené v datech. To znamená, že všechny vnitřní procesy systému jsou řízeny lineárním paradigmatem a analýzy systému vedly k lineárním nebo periodicky oscilujícím řešením. Pokud se však narazilo na nějaké nepravidelnosti v analyzovaných datech, byly připisovány náhodným vnějším vstupům do systému. Jinak řečeno nebyly brány jako součást analyzovaných dat, tedy součást zkoumaného systému, ale pouze jako nechtěné, přidané, externí prvky. Pozdější vývoj teorie chaosu ale ukázal, že náhodný vstup není pouze jediný možný zdroj nepravidelností ve výstupu systému: nelineární, chaotické systémy mohou produkovat mnoho nepravidelných dat s čistě deterministickými rovnicemi pohybu a to autonomním způsobem, to jest bez časově závislých vstupů.

Přistupuje-li se k těmto systémům s touto vědomostí, lze potom pomocí nelineárních metod získat deterministickou charakteristiku systému, neodhalitelnou pomocí klasických lineárních metod. Získá-li se aplikováním vybraných vhodných nelineárních metod na konkrétní nelineární systém tato jeho deterministická charakteristika, lze potom rozeznat skutečné nechtěné externí vstupy do systému, tzv. *dynamický šum*, a pomocí dalších nelineárních metod tento odfiltrvat. Ovšem nalézt vhodné nelineární metody pro daný systém bývá mnohdy obtížné a v současné době toto odvětví moderní fyziky zažívá bouřlivý vývoj.

Nehledě na to, jsou v této práci použity již známé a ověřené postupy a metody nelineárního zpracování, experimentálně ovšem použité na specifický typ dat - řečová data. Některé postupy jsou proto přizpůsobeny pro aplikaci na tato data a jejich výstupy nemusí být tedy pro použitá data optimální. Není vyloučeno, že použití jiných vhodnějších metod z celkové rozsáhlé škály současných nelineárních metod analýzy nelineárních dynamických systémů přinese přesnější eventuálně lepší výsledky.

2 POSTUP VYTVOŘENÍ RPS

Po vysvětlení teorie fázového prostoru a jeho výhodného využití při zkoumání systémů s deterministickými vlastnostmi, představím použitou matematickou symboliku. Pro jednoduchost se další popis omezí na případ, kde fázový prostor je dimenzionálně konečný prostor \mathbf{R}^d , kde d je počet rozměrů fázového prostoru. Stav systému je potom popsán vektorem $\mathbf{x} \in \mathbf{R}^d$, který celý leží ve fázovém prostoru. Dynamiku systému potom lze popsat buďto d dimensionální mapou, nebo jako v tomto případě, jednoznačnou soustavou d diferencionálních rovnic prvního řádu. Pro diskrétní čas $t = n\Delta t$ ve formě:

$$\mathbf{x}_{n+1} = \mathbf{F}(\mathbf{x}_n), \quad n \in \mathbf{R}, \quad (2.1)$$

a v druhém případě pro průběžný čas:

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t)), \quad t \in \mathbf{R}. \quad (2.2)$$

Vektorové pole \mathbf{F} v rovnici 2.1 je definováno tak, že nezávisí explicitně na čase, a proto je nazýváno *autonomní*. Posloupnost bodů \mathbf{x}_n nebo $\mathbf{x}(t)$ řešících rovnici 2.1, respektive 2.2, je nazývána *trajektorií* dynamického systému, s \mathbf{x}_0 , respektive $\mathbf{x}(0)$, *počátečními podmínkami*. Typické trajektorie - atraktory, se budou vzdalovat do nekonečna s přibývajícím časem, nebo zůstanou v ohraničené oblasti navždy - zde využitý případ. To, jak se daný atraktor systému bude chovat, závisí jak na podobě \mathbf{F} (nebo, respektive, \mathbf{f}), tak na počátečních podmínkách; u mnoha systémů jsou možná obě řešení.

Změna stavu dynamického systému se potom děje provedením těchto diferencionálních rovnic a nahrazením starého stavového vektoru ve fázovém prostoru vektorem novým. Použití fázového prostoru se tedy jeví jako přirozené pro aproximace těchto rovnic. Takto řešené rovnice potom budou důležité při predikcích atraktoru a filtrování šumu. Časovou řadou je například s_n , která byla získána nějakou měřicí funkcí $s(\cdot)$ ze signálu \mathbf{x}_n :

$$\{s_n = s(\mathbf{x}_n)\}. \quad (2.3)$$

Jelikož ale (obvykle skalární) posloupnost $\{s_n\}$ sama o sobě nepředstavuje (více-rozměrný) fázový prostor dynamického systému, musí se použít nějaká technika, která posloupnost $\{s_n\}$ převede na stavové vektory fázového prostoru. Tato technika se nazývá *rekonstrukce fázového prostoru*. Existuje několik odborných publikací zabývajících se touto problematikou ([4], [5], [7], [8]), tedy volbou vhodných parametrů potřebných k vytvoření RPS z $\{s_n\}$. Ukázalo se však, že volba optimálních parametrů silně závisí na tom, k jakému účelu budeme vytvořený *rekonstruovaný fázový prostor* (z anglického *reconstructed phase space - RPS*) využívat. V této práci se

jedná především o zpracování zvukového (řečového) signálu a proto bude vhodné použít pro rekonstrukci fázového prostoru hlavní a nejpoužívanější techniku - *metodu zpoždování*. Jejím klíčovým parametrem je časové zpoždění signálu τ . Druhým neméně důležitým parametrem je vhodný rozměr dimense vytvářeného fázového prostoru d .

3 METODA ZPOŽĎOVÁNÍ

Metoda zpoždění spočívá v tom, že se zavede časové zpoždění signálu. To může být chápáno jako vícedimensionální zobrazení signálu s opožděnými verzemi sebe sama [3]. Do jednotlivých vektorů rekonstruovaného fázového prostoru jsou potom transformovány opožděné časové posloupnosti signálu pomocí výrazu 3.1:

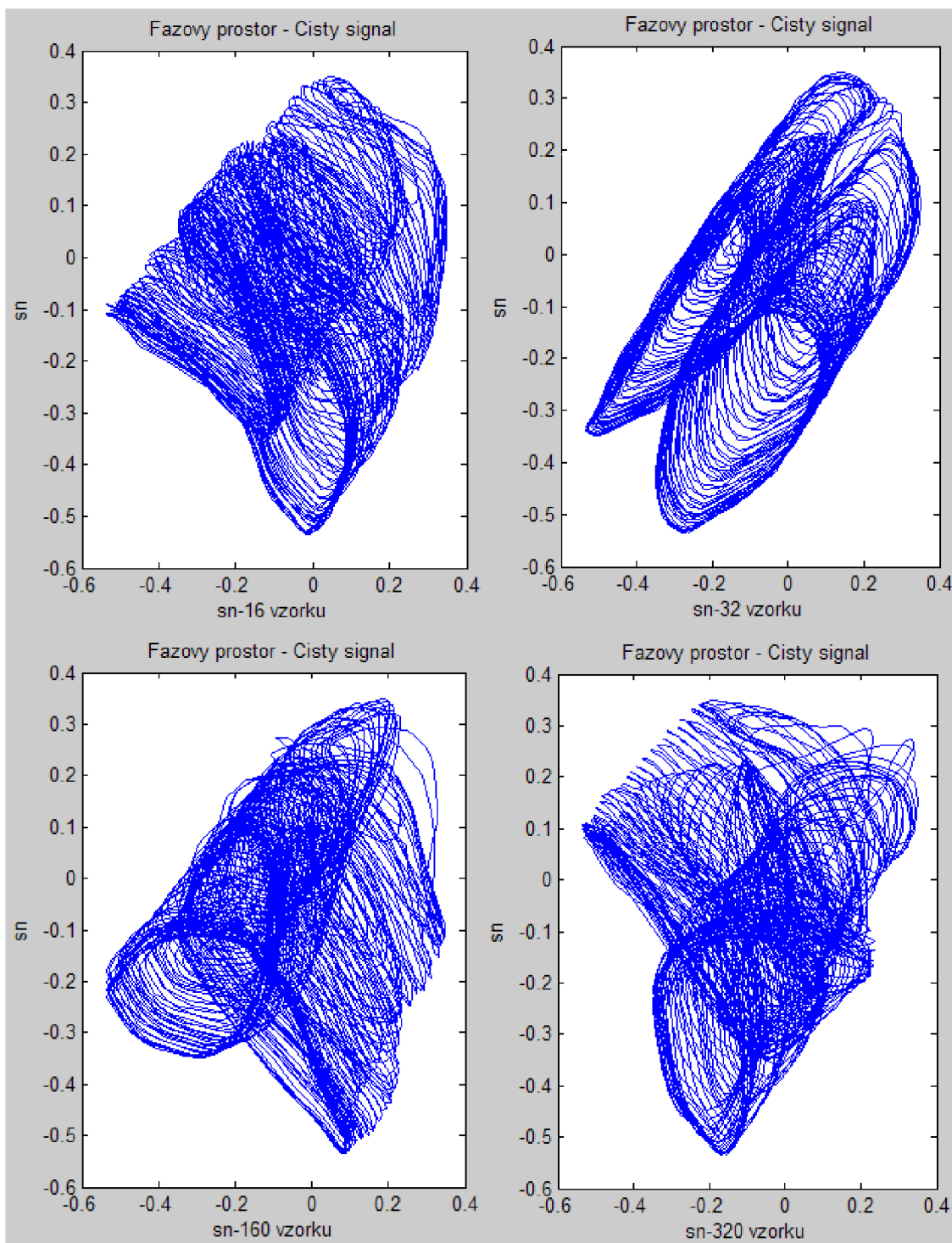
$$\mathbf{x}_n = [x_{n-(d-1)\tau}, x_{n-(d-2)\tau}, \dots, x_{n-\tau}, x_n], \quad (3.1)$$

kde n je časový index, d je rozměr dimenze vhodného fázového prostoru a τ je časové zpoždění signálu \mathbf{x} . Ze známých teorémů Takense [4] a Sauera [5] lze vyvodit, že pokud je sekvence $\{s_n\}$ složena ze skalárních hodnot získaných v určitém stavu dynamického systému, potom, za určitých zobecněných předpokladů, vede vytvoření fázového prostoru pomocí metody zpoždění k vytvoření obrazu původního souboru dat \mathbf{x} v poměru jedna ku jedné (uvažovaný rozměr dimenze d je dostatečně velký).

Vhodná volba časového zpoždění nabývá na důležitosti skrze fakt, že vždy se pracuje s daty určité struktury, signálem který se bude zkoumat. Při volbě malého zpoždění budou v RPS silně korelované prvky vektorů, při použití velkých zpoždění vzniknou vektory jejichž prvky jsou (téměř) nekorelované a data jsou potom (zdánlivě) náhodně rozložena ve fázovém prostoru. Pro lepší pochopení je v tomto případě nasnadě vizuální metoda. Tedy vhodné zobrazení RPS ve 2D nebo 3D prostoru. Této metodě se věnuje následující podkapitola.

3.1 Volba rozumného časového zpoždění τ pomocí zobrazení atraktoru

Pro samotný odhad vhodného časového zpoždění τ a lepšímu porozumění, co přesně vlastně ovlivňuje, je vhodné si zobrazit atraktor vícekrát ve fázovém prostoru a to vždy s jiným použitým zpožděním. Pokud je totiž časové zpoždění τ zvoleno příliš malé, jsou jednotlivé body x_n a $x_{n+\tau}$ (pro diskrétní čas) příslušného vektoru X_N matice RPS 5.2 velmi podobné a vykreslené vektory atraktoru tvoří téměř čáru - $X_N X_{N+1}$. Naopak, pokud je zpoždění příliš velké, pak uspořádání atraktoru je téměř nezávislé a nelze tak z daného zobrazení získat požadovanou informaci. Toto dokládá obrázek 3.1, kde je názorně vidět, jak se zvolené časové zpoždění projeví na vykresleném atraktoru. Obrázek byl získán jako výstup skriptu `zpozdzeni2D.m`, vytvořený za tímto účelem. Zobrazeným signálem je samohláska "u" a trvá přibližně 0,35 sekundy. Zpoždění je uvedeno ve vzorcích, přičemž 16 vzorků = 1 ms, 32 vzorků = 2ms, 160 vzorků = 10ms a 320 vzorků = 20ms.

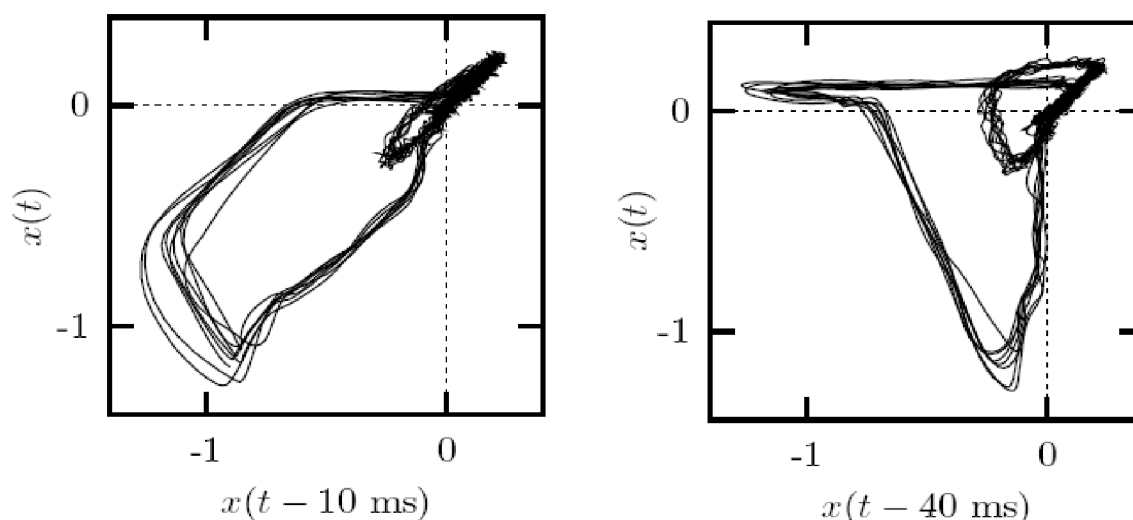


Obr. 3.1: 2D atraktor zobrazený pro různá časová zpoždění opakované hlásky "u".

Souřadnice jednotlivých bodů jsou tvořeny hodnotami jednotlivých vzorků na ose y, přičemž na ose x jim vždy odpovídá hodnota vzorku zpožděná o daný počet vzorků. Nejucelenější struktura atraktoru je získána v tomto konkrétním případě pro zpoždění 32 vzorků. Z uvedeného je patrné, že takovéto atraktory, mající pravidelný,

opakující se tvar, získáme pouze ze vzorků řečového signálu, který se periodicky opakuje (zde několikanásobné opakování samohlásky u). V praxi, kdy se pracuje s celými slovy a větami, je tato metoda zjišťování optimálního časového zpoždění nepoužitelná. Nicméně toto zobrazení vypovídá přinejmenším o důležitosti správné volby časového zpoždění.

Volba optimálního zpoždění je také důležitá z pohledu jeho aplikace. V závislosti na konkrétní časové posloupnosti bude zvoleno konkrétní potřebné zpoždění. Na obrázku 3.2¹ je právě výběr zpoždění pro vhodnou reprezentaci analyzovaných dat vidět.



Obr. 3.2: Znázornění časového zpoždění na lidském magneto-kardiogramu.

V levém bloku na tomto obrázku je použito malé časové zpoždění 10 ms k vykreslení průběhu systoly (kontrakce) srdeční komory. V pravém bloku je zase dobře rozlišitelná pomalejší dilatace (návrat) komory (malá smyčka) díky použitému lehce většímu zpoždění 40 ms.

Přestože v určitých případech může být použité časové zpoždění aplikované na srdeční kardiogram motivováno teoreticky [6], v tomto případě slouží pouze jako vizuální pomůcka. Volba optimálního zpoždění proto není jednoduchá a opět je zde vidět, že závisí vždy na kontextu daného použití.

¹obrázek je zachycuje výstup skriptu *delay*, který je součástí programové části projektu TISEAN [1]. Tento projekt je autory umístěn na internet k volnému stažení s prosbou o jeho využití pro vědecké práce.

3.2 Volba rozumného časového zpoždění τ pomocí vzájemné informace

Časově zpožděná vzájemná informace (souvztažnost) byla navržena Fraserem a Swinneyem [7], jako nástroj pro určení vhodného zpoždění. Pro zjištění vzájemné informace používají nelineární korelační funkci 3.2:

$$S = - \sum_{ij} p_{ij}(\tau) \ln \frac{p_{ij}(\tau)}{p_i p_j}, \quad (3.2)$$

kde pro některé segmenty je p_i pravděpodobnost nalezení hodnoty z časové řady v i -tém intervalu a $p_{ij}(\tau)$ je sdružená pravděpodobnost, že sledování bude v i -tém intervalu a sledovaný čas τ bude v j -tém intervalu. Teoreticky toto vyjádření nemá žádnou systematickou souvislost s velikostí segmentů signálu a může být jednoduše vypočítáno. Pokud při použití určitého zpoždění τ , časově zpožděné vzájemné informace ukážou výrazné minimum S , je toto dobrý kandidát na vhodné časové zpoždění.

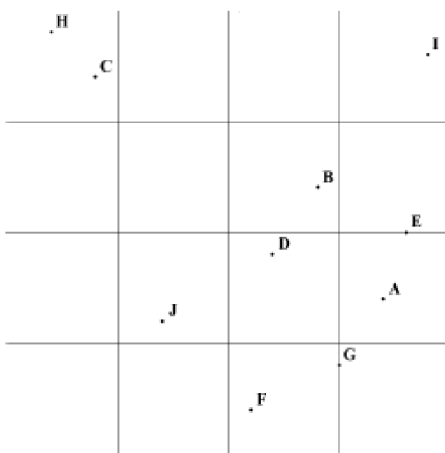
Při stanovení časového zpoždění τ je důležité, aby jeho hodnota byla zvolená vhodně. Pokud by totiž bylo toto zpoždění příliš malé, pak při vykreslení grafu atraktoru zůstanou křivky blízko sebe a budou tvořit téměř čáru.

4 VOLBA VHODNÉHO ROZMĚRU DIMENSE

Pro samotnou volbu vhodného rozměru dimense RPS je použita *Metoda falešných nejbližších sousedů*. Než se však použije, musí se zajistit vhodná vstupní množina prvků této metody - kandidáti pro nejbližší sousedy. Ty lze získat pomocí metody *Box-assisted approach*.

4.1 Box-assisted approach

Box-assisted approach, volně přeloženo jako "*Boxová metoda*", byla popsána Kantzem a Schreiberem [2] jako jednoduchá metoda vícerozměrného prohledávání pro nelineární analýzu časových řad. Nehledě na to, s jakou dimensí RPS se bude pracovat, budou se *kandidáti* na nejbližší sousedy definovat ve 2D prostoru. Použije se k tomu čtvercová mřížka, představující rovnoměrně rozmístěné bloky-boxy (odtud box-assisted approach). Interval této mřížky je ϵ . Všechny vektory (jejich hodnoty) jsou roztrženy do jednotlivých čtverců-boxů v mřížce. Na obrázku 4.1 je vytvořena dvourozměrná mřížka pro demonstrační data, která jsou do ní rozmístěna. Vybere se bod, pro který se hledá nejbližší souseď(i). Prohledávání potom spočívá v nalezení boxu, ve kterém se daný bod nachází a prohledání tohoto boxu a všech přilehlých boxů. Pokud není daný nejbližší souseď nalezen, potom se hledání rozšíří na další přilehlé boxy. Hledání pokračuje dokud nejsou nalezeni všichni požadovaní sousedé. Seznam těchto souseďů se potom dále zpracovává pomocí *Metody falešných nejbližších souseďů* 4.2.



Obr. 4.1: Vzorčky roztrženy do 16 boxů ve dvojdimensionální mřížce, za použití box-assisted metody.

Jedním z problémů této metody je určení vhodného rozměru boxů ϵ . Jejich rozměr je vždy přizpůsoben konkrétní aplikaci a její prohledávací preferenci. Proto určité ϵ zvolené "na míru" danému prohledávání, nemusí být vhodné pro jiné prohledávání (aplikaci). Jestliže je rozměr boxů zvolen malý, potom mnoho boxů zůstane nenaplněných a mnoho se jich bude muset prohledat. To vede k nadměrným požadavkům na paměť a dlouhým výpočtovým operacím.

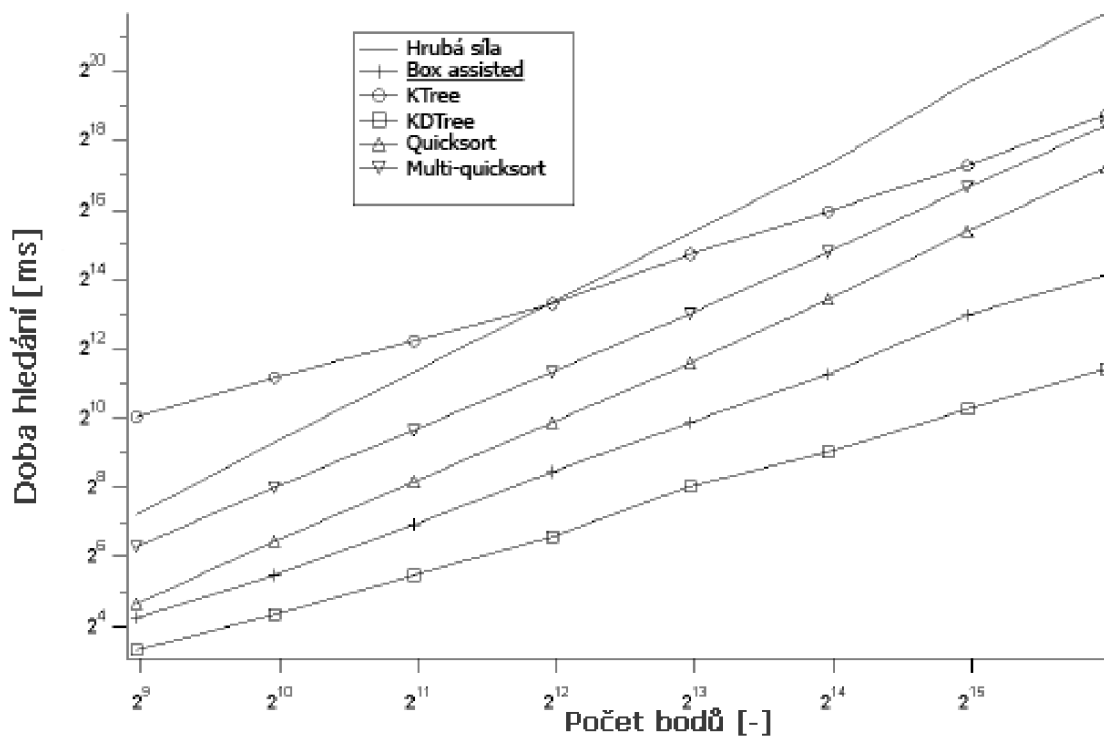
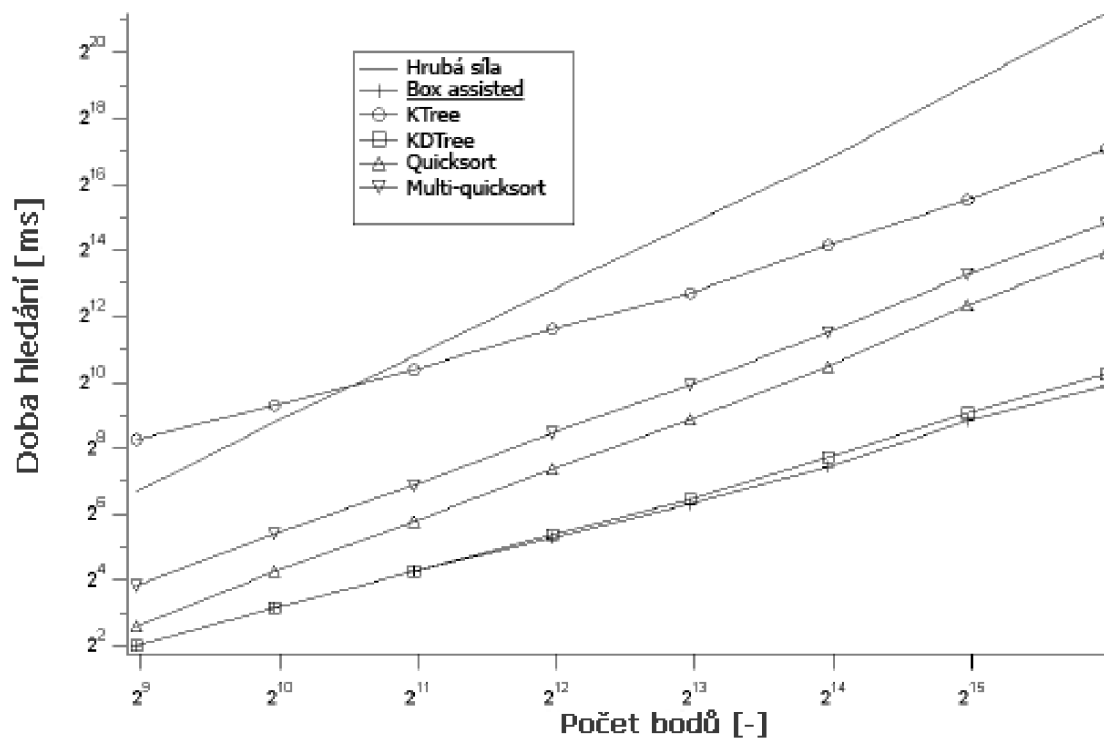
Volba dimensionalit může být také problematická. Schreiber [2] doporučoval dvojdimensionální mřížku. Nicméně to může vést k neúčinnému vyhledávání pro vícerozměrná data. Ta mohou být stále prohledávána, přestože pro nalezení nejbližšího souseda je potřeba daleko většího množství boxů. Na druhé straně, použití vícedimensionálních boxů neekonomicky zvedá nároky na paměť. Na obrázku 4.2 je vidět závislost doby hledání na množství procházených dat metody box-assisted approach, v porovnání s jinými metodami. Křivka je lineární, ovšem s rostoucí dimensí výpočetní náročnost rychle roste a prohledávací čas se exponenciálně zvyšuje. Například pro množství bodů 2^{12} je pro dimenzi $d = 2$ vyhledávací čas roven přibližně $2^{4,5}$, pro dimenzi $d = 3$ je už přibližně 2^9 .

4.2 False nearest neighbours (FNN)

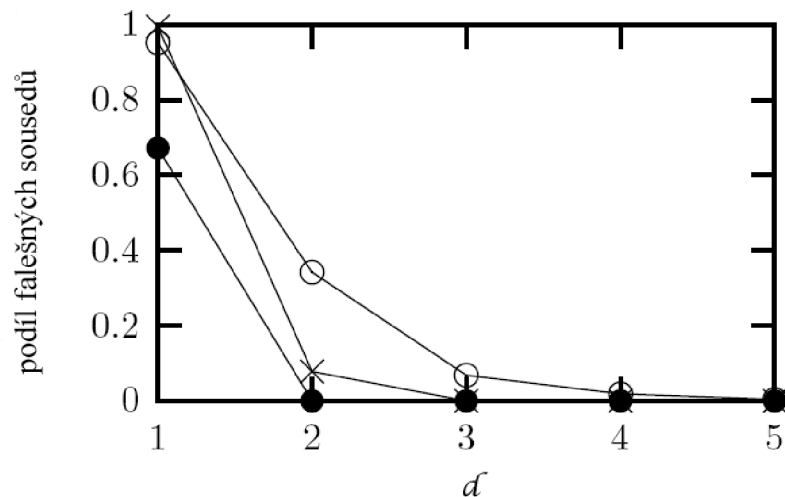
Neboli také Metoda falešných nejbližších sousedů, představena Kennelem [8], slouží k určení minimální dostatečné dimenze d , jenž je zapotřebí k vytvoření ekvivalentního fázového prostoru k prostoru originálnímu. Ekvivalencí se zde myslí zachování topologických vlastností originálního prostoru v nově vytvořeném prostoru.

Předpokládaný minimální dostatečný rozměr vytvářené dimenze pro danou časovou řadu $\{s_i\}$ bude d_0 . To znamená, že v d_0 -dimensionálním zpožděném prostoru je rekonstruovaný atraktor obrazem 1:1 atraktoru v originálním fázovém prostoru. Topologické vlastnosti jsou tedy zachovány a sousedi určitého bodu jsou správně namapováni na sousedy tohoto bodu i v tomto d_0 -dimensionálním prostoru. Při pohybu tohoto dynamického systému jsou na ně stále mapováni znovu a znovu. Nyní se vytvoří d -dimensionální zpožděný prostor, kde $d < d_0$. Potom při vytvoření fázového prostoru pomocí d není topologie zachována. Sousedi daného bodu jsou namapováni na cizí(falešné) sousedy jiných bodů v tomto d -dimensionálním prostoru. Označují se jako tzv. *falešní sousedé*. Pokud je nyní systém v pohybu, sousedi daného bodu jsou stále mapováni na falešné sousedy, cyklus se opakuje, sousedé nejsou namapováni správně a proto nemůže být zachována originální topologie systému a struktura atraktoru se změní.

Vlastní výpočet je následující. Pro každý bod \vec{s}_i v časové řadě se hledá jeho nejbližší soused \vec{s}_j v d -dimensionálním prostoru. Vypočítá se jejich vzdálenost po-



Obr. 4.2: Srovnání doby vyhledávání pro různé metody. Nahoře jsou použita dvojdimensionální data, dole trojdimensionální.



Obr. 4.3: Podíl falešných nejbližších sousedů jako funkce rozměru vytvářené dimense.

mocí 4.1:

$$\| \vec{s}_i - \vec{s}_j \| . \quad (4.1)$$

Následně se iterují oba body a vypočte se hodnota podle 4.2:

$$R_i = \frac{|s_{i+1} - s_{j+1}|}{\| \vec{s}_i - \vec{s}_j \|} . \quad (4.2)$$

Pokud R_i překročí daný heuristický práh R_t (hodnota blízká nule), pak je tento bod označen za bod mající falešného nejbližšího souseda [8]. Pokud procento výskytu falešných nejbližších sousedů je malé, hodnota d , při které byl test proveden, je považována za dostatečnou dimenzi prostoru RPS. Na obrázku 4.3 jsou zobrazeny tři příklady (obrázek je převzat z teoretické části projektu TISEAN [1], se svolením autorů a jejich uvedených licenčních podmínek). Jeden je pro Lorenzův systém (křížky), druhý pro Hénonův systém (plná kolečka) a třetí pro Hénonovu časovou řadu (prázdná kolečka) zašuměnou 10% bílým šumem s Gaussovým rozložením. Je zřejmé že pro Hénonovu časovou řadu je dostatečná dimense $d = 2$ a pro Lorenzovu dimense $d = 3$. Data, která obsahují šum, potřebují větší rozměr dimense d , v tomto případě je dostatečná dimense až $d = 5$.

5 REKONSTRUKCE VÍCEDIMENSIONÁLNÍHO FÁZOVÉHO PROSTORU

Pokud je připraven optimální parametr zpoždění τ a minimální dostačující rozměr dimense m , může se rekonstruovat fázový prostor. Rovnice 3.1 se upraví na tento tvar:

$$\mathbf{x}_n = [x_n, x_{n-\tau}, x_{n-2\tau}, \dots, x_{n-(d-1)\tau}] \text{ pro } n = (1 + (d-1)\tau \dots N), \quad (5.1)$$

kde n je opět časový index, d je rozměr dimense vhodného fázového prostoru a τ je časové zpoždění signálu \mathbf{x} . Prvky ve vektoru \mathbf{x}_n jsou uspořádány oproti hodnotám v časové řadě 3.1 opačně. Nově přibyl parametr N , který odpovídá celkovému počtu hodnot v rámci signálu, tedy všem hodnotám analyzované časové řady. Kompletní popis rekonstruovaného fázového prostoru signálu je dán maticí atraktoru 5.2:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_{1+(d-1)\tau} \\ \mathbf{x}_{2+(d-1)\tau} \\ \vdots \\ \mathbf{x}_N \end{bmatrix} = \begin{bmatrix} x_{1+(d-1)\tau} & \dots & x_{1+\tau} & x_1 \\ x_{2+(d-1)\tau} & \dots & x_{2+\tau} & x_2 \\ \vdots & & \ddots & \\ x_N & \dots & x_{N-(d-2)\tau} & x_{N-(d-1)\tau} \end{bmatrix} \quad (5.2)$$

Řádky této matice tvoří vektory vytvořené pomocí rovnice 5.1. Pokud je k dispozici N hodnot v rámci signálu, pak počet vektorů (řádků matice atraktoru) je roven $N - (d-1)\tau$. Konkrétní implementace a vytváření matice RPS, získávání parametrů pro vybudování RPS τ a d , a další potřebné výpočty v programu Matlab jsou popsány v kapitole 6.

6 IMPLEMENTACE V PROSTŘEDÍ MATLAB

V programovacím prostředí Matlab jsem implementoval metodu pro získání matice rekonstruovaného stavového prostoru. V hlavním skriptu, který jsem nazval `RPS.m` je nejprve zavolána segmentace signálu pomocí `segmentation.m`. Vhodné časové zpoždění je počítáno pomocí skriptu `delay.m` a dostatečný rozměr dimense pomocí skriptu `FNN.m`. Následuje vytvoření samotného fázového prostoru a zavolání filtrace pomocí funkce `filtrovani.m`. Zpětně je potom poskládán signál z vyfiltrovaných dat a zavolány funkce `SNRg.m` a `...` pro měření odstupů signálu od šumu. Všechny skripty obsahují komentáře pro snadnější porozumění prováděných operací a delší skripty jsou logicky členěny pro snadnější orientaci. Všechny zde popisované skripty a funkce jsou uloženy na přiloženém CD-R.

6.1 Skript RPS.m

Tento skript je hlavním skriptem, který využívá a volá všechny zde popsané ostatní skripty a funkce. Je logicky rozdělen do několika částí podle prováděných výpočtů. V první části uživatel vybere řečový signál, který je načten a navzorkován ($f_{vz} = 16\text{kHz}$). Signál je dále zarušen aditivním Gaussovým bílým šumem pomocí funkce `awgn()` a následně nasegmentován pomocí skriptu `segmentation.m` 6.2 na jednotlivé rámce. V druhé části je zavolána funkce `delay.m` 6.3, která vypočítá jednotlivá zpoždění pro všechny rámce a uloží je do vektoru *tau*. Zpoždění je navýšeno o 1, což se lehce kladněji projevilo na výsledcích. Obdobně je potom vytvořen vektor *d*, který obsahuje vypočítané rozměry dimense fázového prostoru pomocí `FNN.m` 6.4. Jednotlivé řádky sloupcových vektorů *tau* a *d* vždy obsahují vypočtené zpoždění, respektive rozměr dimense, pro příslušný rámeček (tedy sloupec v matici *signal_segments*, kde každý sloupec odpovídá jednomu rámečku nasegmentovaného signálu). V třetím kroku je sestavena vlastní matice RPS podle vzorce 5.2 ze získaných parametrů *tau* a *d* a to vždy pro každý rámeček. Vlastní postup vytváření matic RPS je detailně komentován uvnitř tohoto skriptu v odpovídajícím, tedy třetím kroku, a pro názornost je zachycen na obrázku 6.1. Jednotlivé matice RPS jsou uloženy pro každý rámeček na odpovídající řádek matice polí *RPSarray*.

Ve čtvrtém kroku je zavolán vlastní filtrační algoritmus, implementovaný ve skriptu `filtrovani.m`. Z jeho výstupu je získána matice polí *RPSarray_filtr* 6.5. Z této matice se potom v pátém kroku zpětně rekonstruují jednotlivé rámce. Postup je inverzní k tomu, který je popsán ve třetím kroku, ale s jistou změnou. Pro každý rámeček je nejprve vytvořena a vyplněna nulami matice *RPStemp2* o rozměrech příslušné RPS z *RPSarray_filtr* - *RPStemp*. Ovšem počet řádků *RPStemp2* je 320 (počet vzorků v každém rámečku při délce rámečku 20ms a $f_{vz} = 16\text{kHz}$) a ne tedy

```

% a) Pro kazdy ramec (segment) je vytvorena nova matice RPS.
% b) Jsou vytvareny jednotlivé sloupce docasne matice "RPStemp". Pomoci
% parametru "tau" a "d" (vztahujicim se k prislusnemu segmentu) jsou
% vyseknuty prislusne hodnoty z daneho segmentu
% nasegmentovaneho signalu "y" - [DP vzorec 5.2].
% Pozn.: V Matlabu je toto reseni rychlejsi nez prochazeni jednotlivych
% radku a vybirani o "tau" zposdenych hodnot pomoci prikazu for.
% c) Docasna matice "RPStemp" je vsky ulozena na prislusny radek matice
% poli "RPSarray" (cislo radku odpovida cislu segmentu).
% d) Docasna matice "RPStemp" je vsky smazana, aby mohla byt znovu naplnena
% hodnotami z dalsiho segmentu.

for j = 1:poc_segmentu %Ad.a)
    for i = 1:d(j,1) %Ad.b)
        RPStemp(:,i) = signal_segmenty((1+(d(j,1)-i)*(tau(j,1))):...
            (Delka_segmentu_vzorky - ((i-1)*tau(j,1))),j);
    end
    RPSarray(j,1) = (RPStemp); %Ad.c)
    clear RPStemp %Ad.d)
end

```

Obr. 6.1: Algoritmus sestavování matic RPS pro jednotlivé segmenty. Je součástí skriptu RPS.m.

$320 - x_{N-(d-1)\tau}$, jak je uvedeno ve vzorci 5.2. Do prvního sloupce v matici $RPStemp2$ je potom zapsán příslušný sloupec z $RPStemp$, ale je uložen až od řádku s hodnotou $(d-a)\tau + 1$, přičemž pro první sloupec má a hodnotu 1, pro druhý 2 a pro poslední $a = d$. Hodnoty jsou pak v jednotlivých řádcích sečteny a vyděleny počtem nenulových hodnot. Jde tak v podstatě o vyprůměrování vyfiltrovaných hodnot. Teprve z nich je vytvořen rekonstruovaný rámec a následně celý signál. Snáze je to pochopitelné z následujícího obrázku 6.2, kde vstupní matice RPS jej vytvořena z reprezentativní řady $s_n = [1, 2, 3, 4, 5, 6, 7, 8]$, přičemž $d = 3$ a $\tau = 2$:

$$\begin{bmatrix} 5 & 3 & 1 \\ 6 & 4 & 2 \\ 7 & 5 & 3 \\ 8 & 6 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 2 \\ 0 & 3 & 3 \\ 0 & 4 & 4 \\ 5 & 5 & 0 \\ 6 & 6 & 0 \\ 7 & 0 & 0 \\ 8 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0+0+1 \\ 0+0+2 \\ 0+3+3 \\ 0+4+4 \\ 5+5+0 \\ 6+6+0 \\ 7+0+0 \\ 8+0+0 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 6 \\ 8 \\ 10 \\ 12 \\ 7 \\ 8 \end{bmatrix} : \begin{bmatrix} 1 \\ 1 \\ 2 \\ 2 \\ 2 \\ 2 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{bmatrix}$$

Obr. 6.2: Příklad sestavení rámce z vyfiltrovaných hodnot matice RPS.

V posledním kroku jsou zavolány vyhodnocovací algoritmy v podobě funkcí `SNRg.m` 6.6. Výstupem skriptu jsou potom soubory `zaruseny.wav` obsahující zarušený signál a `filtrovany.wav` obsahující vyfiltrovaný signál. Od začátku spuštění skriptu je uživatel informován o aktuálně probíhajícím výpočtu pomocí ukazatele průběhu volaného ze skriptu `waitbar.m`.

6.2 Skript `segmnetation.m`

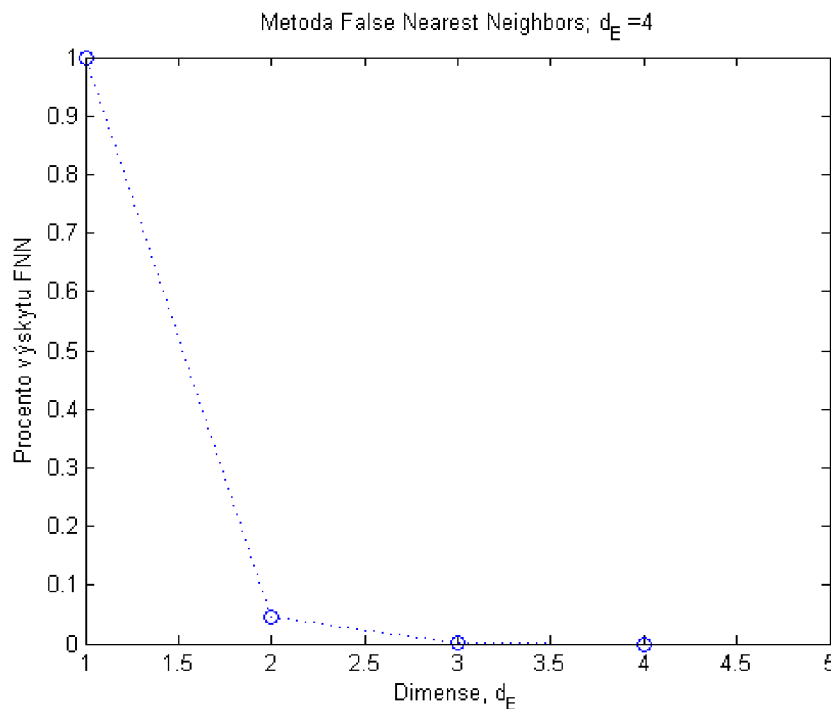
Provede segmentaci vstupního signálu podle zadaných parametrů. Funkce `[out]=segmentation(signal, delka, posun, okno)` vrátí matici, jejíž sloupce tvoří segmenty vstupního signálu zadané délky a s daným posuvem mezi jednotlivými segmenty. Segmenty mohou být váhovány zadaným oknem (obdélníkové, trojúhelníkové, Hammingovo, Bartlettovo, Blackmanovo).

6.3 Skript `delay.m`

Autorem je R. Moddemeijer a je výstupem jeho disertační práce [9]. Skript odhaduje zpoždění signálu y v závislosti na x za použití informačního zpoždovacího kritéria. Zápis funkce je `[lag,criterion] = delay(x,y,shift,order,base)`. Vstupní proměnné jsou signály x a y stejné délky. Dále jsou zde volitelné parametry. *Shift* je absolutní hodnota maximálního posunutí obou signálů. Výstupní proměnná *lag* potom leží v mezích $-shift \leq lag \leq shift$. Parametr *order* určuje počet prováděných výpočtů, které rostou přibližně kvadraticky v závislosti na hodnotě parametru *order* a lineárně v závislosti na délce x . Pro *order* = 1 je metoda přibližně ekvivalentní ke křížové korelační analýze. Pro jeho vyšší hodnoty (2,3,4) je metoda schopna zahrnout do odhadu dodatečné vzorky signálu. Je vytvořena kovariační matice a počítáno *informační kritérium*, které měří vzájemné informace segmentů obou signálů x a y . Parametr *base* je základ logaritmu, kterým je *kritérium* děleno, tedy jeho jednotku. Žádný z volitelných parametrů není nutný pro tuto práci a není tedy použit. Jako testovací signály (časové řady) mohou být zvoleny dva vektory sta náhodně vygenerovaných čísel s normálním rozložením pomocí příkazu `normrnd`, k nalezení v souboru `delay_example.m`. Výsledné zpoždění pro tyto signály vychází typicky kolem 2 až 4, v závislosti na vygenerovaných hodnotách. Podobné zpoždění vychází také pro reálné řečové nahrávky.

6.4 Skript FNN.m

Tento skript odhaduje minimální dostatečnou dimenzi dE vytvářeného stavového prostoru. Zápis funkce je $[dE] = FNN(dataStream, tau)$. Parametr $dataStream$ je vstupní nasegmentovaný signál a parametr τ je výstupní hodnota proměnné lag ze skriptu `delay.m` 6.3 - tedy optimální zpoždění pro daný segment. Skript postupně zvyšuje dimenzi dE až do dané limitní hodnoty, potom podle rovnice 4.2 odhadne optimální rozměr dimense. Příklad výstupu skriptu je na obrázku 6.3, kde optimální dimense vyšla 4, což je typická velikost pro danou vstupní časovou řadu. Rozměry dimense vytvářeného atraktoru pro reálné řečové nahrávky vycházejí zpravidla v rozsahu od 3 do 5.



Obr. 6.3: Výstup skriptu `FNN.m`. Podíl falešných nejbližších sousedů jako funkce rozměru vytvářené dimense.

6.5 Skript `filtrvani.m`

V tomto skriptu je naprogramován vlastní filtrační algoritmus. Je volán jako funkce $[RPSarray_filtr] = filtrvani(RPSarray, poc_segmentu, d, tau)$, která vypočítá výstupní matici polí $RPSarray_filtr$ obsahující filtrované RPS matice ze vstupních parametrů: $RPSarray$ - matice polí obsahující vytvořené RPS ze zarušeného signálu, $poc_segmentu$

- proměnná obsahující počet segmentů nasegmentovaného signálu, d a τ - vektory, obsahující vytvářecí parametry jednotlivých RPS.

V první části skriptu se nejprve pro každý (referenční) řádek každé RPS matice z $RPSarray$ změří Euklidovská vzdálenost D_E ke všem ostatním řádkům v dané RPS pomocí vzorce 6.1

$$D_E((x_1, x_2), (y_1, y_2)) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}, \quad (6.1)$$

kde (x_1, x_2) a (y_1, y_2) jsou souřadnice dvou bodů X a Y . Pro každý referenční řádek se potom vybere d (odpovídající hodnotě rozměru dimense pro danou RPS) vektorů s nejnižší Euklidovskou vzdáleností a uloží spolu s referenčním vektorem do dočasné matice. V druhé části se potom vypočítá aritmetický průměr pro jednotlivé sloupce v dočasné matici a touto hodnotou se přepíší všechny hodnoty v daném sloupci. Popisovaná metoda je odvozena z [2] (kapitola 4.5). Tato filtrační metoda je funkční, ale ne optimální, což zmiňuju v kapitole 6. Skript je podrobně komentován a jeho zkrácený zápis je uveden na obrázku 6.4.

```

for j=1:poc_segmentu %Cyklus pro jednotlivé segmenty.
    RPStemp = RPSarray(j,1); %Dočasna matice pro kazdy segment.
    [radky,sloupce]=size(RPStemp);
    RPStemp2 = zeros(size(RPStemp));

    for i=1:radky %Cyklus pro vybrani kazdeho radku jako referencniho v danem segmentu.
        for k=1:radky %Cyklus pro Euklidovskou vzdalenost kazdeho radku k referencnimu.
            for l=1:sloupce
                Euklid(k,l)= (RPStemp(k,l)-RPStemp(i,l))^2;
            end
            Euklid2(k,l) = (sum(Euklid(k,:)))^0.5;
        end
        MVV(i,j) = (RPStemp(Ix(1:(d(j,l)+1)),:));
        %+-----+
        %% VLASTNÍ FILTRÁČNÍ PROCES
        %+-----+
        for m=1:(d(j,l)+1)
            RPStemp2(Ix(m,l),:) = (sum(MVV(i,j) (:, :)))/(d(j,l)+1);
        end
        RPSarray_filtr(j,1) = {RPStemp2};
end

```

Obr. 6.4: Ukázka ze skriptu `filtrrovani.m`

6.6 Skript SNRg.m

V průběhu načtení, zašumění, filtrování řečového signálu po spuštění skriptu `RPS.m`, jsou výstupy jednotlivých operací ukládány takto: `signal1` = originální signál, `signal2` = zarušený signál, `signal3` = filtrovaný signál. V tomto skriptu - funkci, jsou tyto proměnné načteny, a je počítáno *globální* SNR, tedy odstup signálu od šumu přes celý signál na základě poměru energií. *Globální* SNR je vypočítáno podle vzorce 6.2

$$SNRg = 10 \log_{10} \frac{E_s}{E_\epsilon} = 10 \log_{10} \frac{\sum_{n=0}^{l-1} s^2[n]}{\sum_{n=0}^{l-1} [\tilde{s}[n] - s[n]]^2}, \quad (6.2)$$

kde E_s je energie nezarušeného signálu, E_ϵ je energie šumu, $s[n]$ je nezarušený signál ve vzorcích n , l je délka signálů ve vzorcích a $\tilde{s}[n]$ je zarušený signál. Za $s[n]$ jsou dosazeny vzorky ze signálu `signal1` a za $\tilde{s}[n]$ vzorky ze `signal2` a potom ze `signal3`. Vlastní výpočet je vidět na obrázku 6.5.

```
%Vypocet celkove energie originalniho signalu.  
En = sum(X.^2);  
%Vypocet energetickeho rozdlilu Ee1 (mezi filtrovanou nahravkou F  
% a originalnim signalem X).  
Er1 = (F - X);  
Ee1 = sum(Er1.^2);  
%Vypocet Globaniho SNR pro filtrovany-originalni signal.  
SNRglobal = 10*(log10(En/Ee1));  
disp( sprintf('SNRglobalni pro originalni-filtrovaný signal je %g dB',SNRglobal) )
```

Obr. 6.5: Výpočet *globálního* SNR ve skriptu `SNRg.m`.

Je počítán odstup originálního k filtrovanému signálu a také odstup originálního k zarušenému signálu. Výsledky v dB jsou zobrazeny v příkazovém okně Matlabu.

6.7 Skript SNRseg.m

Pro *globální* SNR je charakteristické, že s analyzovaným signálem pracuje jako s celkem. V tom spočívá jeho matematická jednoduchost - měření ukazuje průměrnou chybovost zpracovaného signálu v závislosti na čase. Ale právě slovo průměrný je bohužel také indikátorem nepoužitelnosti SNR v některých případech. SNR totiž velmi špatně odhaduje kvalitu řečového signálu pro velmi velkou oblast různých

typů zkreslení řečového signálu. Je to proto, že SNR měření není spjato s žádným konkrétním subjektivním parametrem kvality řeči, ale také proto, že SNR váží všechny chyby v časovém průběhu řeči stejně. Jinými slovy, *globální* SNR je zatíženo chybou, protože do tohoto výpočtu výkonu řeči jsou zahrnuty i části signálu bez řečové aktivity, které snižují celkový výkon řečového signálu. V řečových pauzách je teoreticky nulový výkon signálu. Ve skutečnosti je i zde zbytkový výkon generovaný hlasovým ústrojím člověka. Korektní výpočet SNR lze získat při vážení jednotlivých segmentů ve vzorci 6.2 jejich řečovou aktivitou. Pro detekci a vyhodnocení segmentu jako segmentu obsahující řeč nebo šum je použit jednoduchý detektor řečové aktivity `vad_energy.m` (fungující na základě sledování krátkodobé energie signálu). Výsledný vzorec 6.3 pro výpočet *segmentálního* SNR je potom definován takto:

$$SNR_{seg} = \frac{1}{K} \sum_{i=0}^{L-1} \left(10 \log_{10} \frac{\sum_{n=0}^{M-1} s_i^2[n]}{\sum_{n=0}^{M-1} [\tilde{s}[n] - s[n]]^2} \cdot VAD_i \right), \quad (6.3)$$

kde L je celkový počet analyzovaných segmentů, M je délka segmentu a K je počet segmentů s řečovou aktivitou. Hodnota VAD_i nese informaci o řečové aktivitě v i -tém segmentu (1=řeč, 0=pauza). Pro každý rámeček (typicky 15-25 milisekund - doba nutná pro přechod lidského hlasového ústrojí z jednoho fyzického stavu do druhého) je počítáno SNR, výsledky jsou zlogaritmovány a proveden jejich součet. Konečné vyhodnocení se získá jako jejich průměr (děleno K - počtem segmentů obsahujících řeč). Tato metoda umožňuje přiřadit odpovídající váhu hlasitým a tichým částem řeči. Příklad vlastní implementace je znázorněn na obrázku 7.2.

```

%Ziskani rozdilu filtrovani-originalni signal.
G = (F - X);
%Tento sum nasledne segmentujeme jako originalni nezaruseny i zaruseny signal.
[Bsum_seg]=segmentation(G, (Delka_segmentu_cas*Fs), (Delka_segmentu_cas*Fs));
%Celkova energie (Esb) obsazena v jednotlivych segmentech(ne vzorcich) nasegmentovaného
%rozdilu G.
Esb = sum(Bsum_seg.^2);
%Rozdil energii (Er2) originalni nezarusne nahravky a z filtrovaného signalu pres
%jednotlive segmenty. Eo obsahuje energie segmentu nezaruseneho signalu.
Er2 = (Eo./Esb);
%Vlastni vypocet SSNR (K=pocet recovych segmentu, detection1=vektor
%obsahujici 1(rec) nebo 0(sum)), K = pocet segmentu obsahujicich rec.
SNRseg_vysledek2 = (1/K)*(sum( (10*log10 (Er2)).*detection1)) ;
disp( sprintf('SNRsegmentalni pro originalni-filtrovani signal je %g dB',SNRseg_vysledek2))

```

Obr. 6.6: Výpočet *segmentálního* SNR ve skriptu SNRseg.m.

7 VYHODNOCENÍ VÝSLEDKŮ

Implementovaný algoritmus jsem zkoušel na více vzorcích řečových nahrávek a pro prezentaci výsledků jsem vybral tyto 4 následující, které jsou uloženy na připojeném CD-ROMu A.1 ve složce **Nahravky**. Jedná se o tyto nahrávky:

1. Vzorek A: *"Přišel jsem včera večer pozdě"*
2. Vzorek B: *"Vlak z Lelekovic do Brna dnes nejede"*
3. Vzorek C: *"Hoří hospoda, zavolejte hasiče"*
4. Vzorek D: hláska "U"

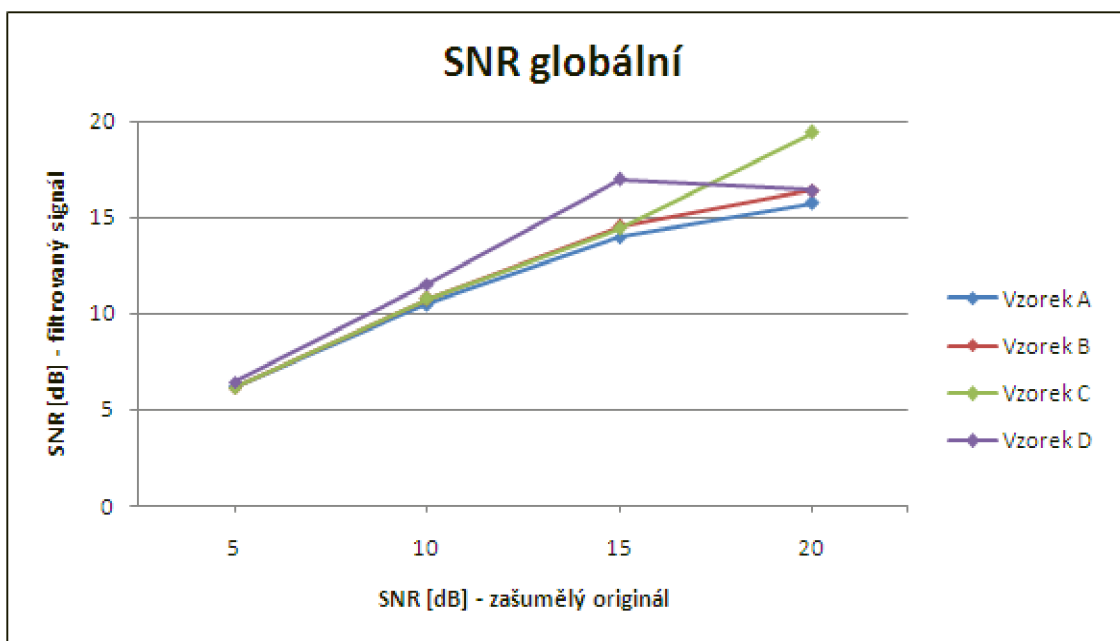
Signál byl vždy segmentován s délkou segmentu 20ms. Výsledky pro všechny 4 vzorky jsou uvedeny v tabulkách 7.1 a 7.2. Originální vzorek byl vždy zarušen aditivním Gaussovým bílým šumem pomocí funkce `awgn()` postupně pro SNR 5, 10, 15 a 20 dB. V tabulkách jsou uvedeny pro porovnání a ověření funkčnosti vyhodnocovacích algoritmů i hodnoty SNR pro originální - zarušený signál. Na obrázcích 7.1 a 7.2 jsou potom graficky znázorněny hodnoty SNR pro vyfiltrované signály v závislosti na daném SNR zarušení. Z výsledků vyplývá, že metoda je účinnější pro nižší hodnoty SNR. I přestože samotný filtrační algoritmus není optimální, v některých případech dochází ke zlepšení subjektivní kvality signálu. Lepších výsledků pro nižší hodnoty SNR je dosahováno také u kratších signálů, zde vzorek D.

Tab. 7.1: Naměřené výsledky při použití skriptu SNRg.m.

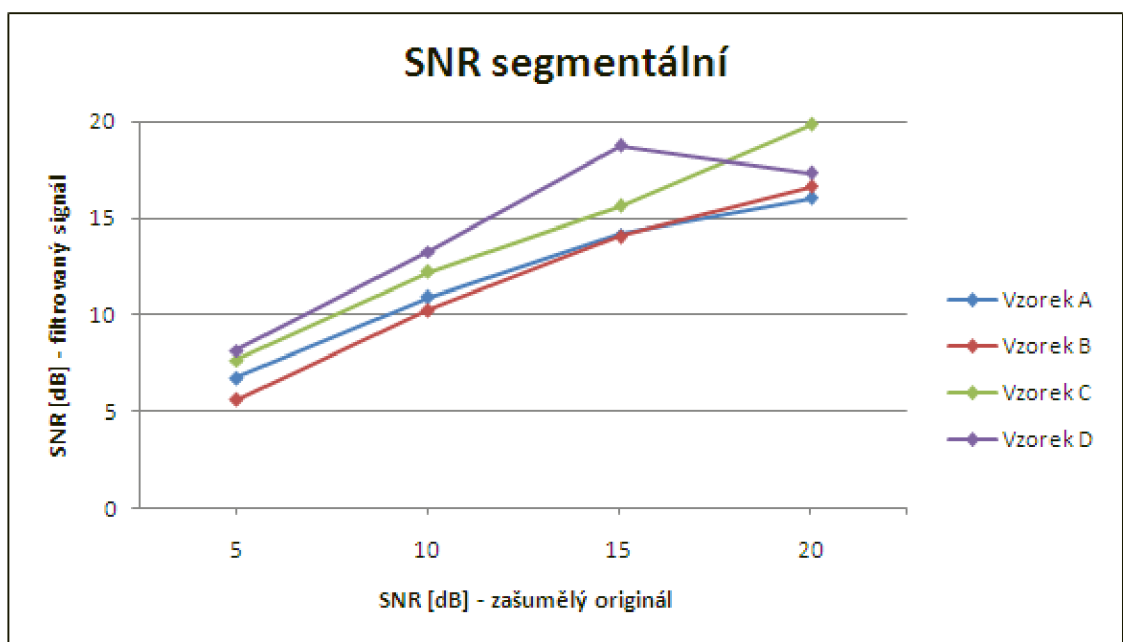
Vzorek:	SNRg [dB]							
	originální - zarušený				originální - filtrovaný			
zarušení [dB]	5	10	15	20	5	10	15	20
A	5,106	9,923	14,970	20,012	6,164	10,502	13,970	15,763
B	4,975	10,017	15,039	20,047	6,155	10,772	14,517	16,432
C	4,966	9,977	15,003	20,234	6,148	10,757	14,438	19,435
D	5,162	9,960	15,1125	20,029	6,446	11,552	17,0013	16,432

Tab. 7.2: Naměřené výsledky při použití skriptu SNRseg.m.

Vzorek:	SNR _g [dB]							
	originální - zarušený				originální - filtrovaný			
zarušení [dB]	5	10	15	20	5	10	15	20
A	5,717	10,495	15,629	20,667	6,724	10,886	14,158	16,004
B	4,344	9,432	14,419	19,367	5,569	10,231	14,044	16,622
C	6,445	11,571	16, 517	20,667	7,644	12,220	15,598	19,816
D	6,740	11,568	16,707	21,538	8,150	13,264	18,719	17,326



Obr. 7.1: Výsledky SNR pro aditivní Gaussův bílý šum. Počítáno pomocí skriptu SNRg.m 6.6.



Obr. 7.2: Výsledky SNR pro aditivní Gaussův bílý šum. Počítáno pomocí skriptu SNRseg.m 6.7.

8 ZÁVĚR

Práce se zabývá nelineární analýzou časových řad, v dnešní době populární, ovšem stále široké a v mnoha směrech nepopsané téma moderní vědy. Cílem bylo objasnit získání rekonstruovaného stavového prostoru RPS, pomocí něhož je možné popsat signály (časové řady) zcela jiným způsobem než je běžné při časové nebo frekvenční analýze, a to aproximovanou geometrickou strukturou ve vyšší dimenzi, tedy atraktorem.

Byly představeny metody pro výpočet parametrů potřebných pro rekonstrukci fázového prostoru. Pomocí metody zpoždění a hledání vzájemné informace se připraví vhodné časové zpoždění τ . Vhodný rozměr dimense vytvářeného fázového prostoru je získán pomocí metody box-assisted approach a falešných nejbližších sousedů. Popis těchto metod a samotná rekonstrukce fázového prostoru, spolu s uvedenou teorií nutnou pro pochopení popisované problematiky, tvoří stěžejní část této práce. Hlavním informačním zdrojem, ze kterého jsem čerpal byla kniha *Nonlinear Time Series Analysis* [2] autorů H. Kanze and T. Schreiber.

Všechny popisované metody byly naprogramovány v prostředí Matlab. Jednotlivé skripty obsahují uvedené metody, skriptem hlavním je `RPS.m`. Opět jsem větší důraz kladl, na samotnou rekonstrukci RPS matice a její rozložení. Použitá filtrační metoda vychází z jednodušší metody potlačování šumu a v tomto směru by mohla být tato práce rozšířena. Presentace výsledků je uvedena v kapitole 7. Výsledné filtrované signály jsou vyhodnocovány pomocí dvou metod měřících SNR, globální SNR a segmentální SNR. Všechny skripty jsou okomentovány a jejich bližší popis a vysvětlení použitých algoritmů uživatel nalezne v kapitole 6.

Závěrem této práce je teoreticky zdokumentovaná a v Matlabu implementovaná nelineární metoda analýzy řečových signálů s obsaženým filtračním algoritmem pro částečné potlačení šumu.

LITERATURA

- [1] R. Hegger, H. Kantz, and T. Schreiber, *Practical implementation of nonlinear time series methods: The TISEAN package* <http://www.mpipks-dresden.mpg.de/~tisean/Tisean_3.0.1/index.html> 9, 413 (1999).
- [2] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis*, Cambridge University Press, Cambridge (2004).
- [3] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw, *Geometry from a time series*, Physical Review Letters, vol. 45, pp. 712-716, 1980.
- [4] F. Takens, *Strange Attractors in Turbulence*, Lecture Notes in Math. Vol. 898, Springer, New York (1981).
- [5] T. Sauer, J. Yorke, and M. Casdagli, *Embedology*, J. Stat.Phys. 65, 579 (1991).
- [6] M. Richter and T. Schreiber, *Phase space embedding of electrocardiograms*, to appear in Phys. Rev. E (1998).
- [7] A. M. Fraser and H. L. Swinney, *Independent coordinates for strange attractors from mutual information*, Phys. Rev. A 33, 1134 (1986).
- [8] M. B. Kennel, R. Brown, and H. D. I. Abarbanel, *Determining embedding dimension for phase-space reconstruction using a geometrical construction*, Phys. Rev. A 45,3403 (1992).
- [9] R. Moddemeijer, *Delay-Estimation with Application to Electroencephalograms in Epilepsy (Phd-thesis)*, Universiteit Twente, 1989, Enschede (NL), ISBN: 90-9002668-1.
- [10] Ing. Radek Zezula, Ph.D., Ing. Ivan Koula, Prof. Ing. Zdeněk Smékal, CSc., *Využití RPS pro potlačování šumu v řečových signálech*, <<http://www.elektrorevue.cz/clanky/07002/index.html>>
- [11] <<http://hungry-lord.wz.cz/data/chaos.php>>
- [12] <<http://www.fit.vutbr.cz/~tisnovpa/fract/clanky/2.htm>>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

RPS rekonstruovaný fázový prostor – Reconstructed Phase Space

FNN falešní nejbližší sousedé – False Nearest Neighbors

τ časové zpoždění

ϵ interval mřížky metody Box-assisted approach

d rozměr dimense

n časový index

x_n signál

$\{s_n\}$ časová řada

N délka signálu

f_{vz} vzorkovací kmitočet

SNR odstup signálu od šumu – Signal to Noise Ratio

dB decibel

SEZNAM PŘÍLOH

A PŘÍLOHA

A.1 Obsah příloženého CD-ROM

Strukturu příloženého CD-ROMu je na obrázku A.1.



Obr. A.1: Struktura příloženého CD.

V adresáři **Matlab** jsou uloženy tyto skripty:

<i>RPS.m</i>	<i>segmentation.m</i>
<i>delay.m</i>	<i>FNN.m</i>
<i>filtrvani.m</i>	<i>SNRg.m</i>
<i>SNRseg.m</i>	<i>vad_energy.m</i>
<i>delay_example.m</i>	<i>zpozdeni2D.m</i>
<i>waitbar.m</i>	

Adresář **Nahravky** obsahuje čisté nezarušené řečové nahrávky pro testování. Všechny nahrávky jsou uloženy ve formátu **.wav*.

Adresář **Text_diplomove_prace** obsahuje soubory *diplomka.pdf* a *diplomka.ps*, ve kterých je uložen vlastní text diplomové práce. Dále obsahuje jeden podadresář **Latex_zdrojovy_kod**, ve kterém je umístěn zdrojový kód textu této diplomové práce ve formátu \LaTeX . Hlavním souborem je *diplomka.tex*. Je zde uložen i soubor *diplomka.ppr* jehož spuštěním v programu PSPad jsou načteny všechny soubory s příponou **.tex*, tedy celého projektu. V kořenovém adresáři CD-ROMu je uložen soubor *Readme.txt*, ve kterém uživatel nalezne stručné shrnutí diplomové práce, její zaměření a popis obsahu tohoto CD-ROMu tak jak je vypsán v této příloze.