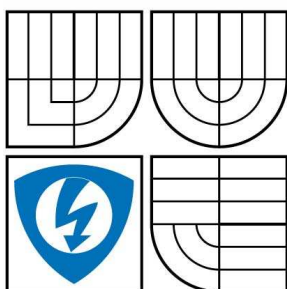


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKACNÍCH
TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

ADAPTIVNÍ POTLAČENÍ ŠUMU V ŘEČI

ADAPTIVE NOISE REJECTION IN SPEECH

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

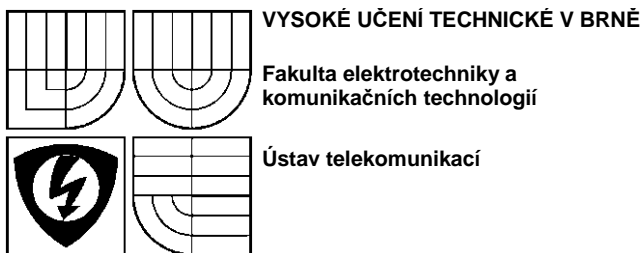
AUTOR PRÁCE
AUTHOR

JAN VOMÁČKA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Petr Sysel, Ph.D.

BRNO 2009



Bakalářská práce

bakalářský studijní obor **Teleinformatika**

Student: Jan Vomáčka

ID: 98091

Ročník: 3

Akademický rok: 2008/2009

NÁZEV TÉMATU:

Adaptivní potlačení šumu v řeči

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s metodami adaptivního potlačení šumu v řečovém signálu. Prostudujte zejména požadavky metod na parametry vstupních signálů. Na základě toho sestavte metodiku pro záznam testovacích signálů a proveďte záznam signálů v reálných prostředích. V simulačním prostředí Matlab implementujte několik metod adaptivního potlačení šumu a zhodnoťte jejich výsledky.

DOPORUČENÁ LITERATURA:

[1] Farhang-Boroujeny, B. Adaptive Filters. Theory and Applications. First Edition. New York, Wiley: 1998. ISBN 0-471-98337-3

[2] Vaseghi, S. V., Advanced Digital Processing and Noise Reduction. Second Edition. New York, Wiley:2000. ISBN 0-471-62692-9

Termín zadání: 9.2.2009

Termín odevzdání: 2.6.2009

Vedoucí práce: Ing. Petr Sysel, Ph.D.

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské 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 § 152 trestního zákona č. 140/1961 Sb.

Bibliografická citace:

VOMÁČKA, J. *Adaptivní potlačení šumu v řeči*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 35 s. Vedoucí bakalářské práce Ing. Petr Sysel, Ph.D.

Abstrakt

Tato bakalářská práce se zabývá metodami adaptivního potlačení šumu ve zvukových nahrávkách, především řečových signálů. Práce se zabývá návrhem vhodného systému pro efektivní odstranění šumu ze zvukové nahrávky. V teoretické části jsou rozebrány základy adaptivních filtrů a metody potlačování šumů ve zvukových nahrávkách. V praktické části je uveden návrh a simulace systému pro adaptivní potlačení šumu v řeči.

Klíčová slova

adaptivní filtr, LMS algoritmus, potlačení šumu,

Abstract

This bachelor's thesis deals with methods adaptive noise rejection in acoustic recordings, above all speech signals. Thesis deal with proposal of suitable system for effective removes noise from acoustic recordings. In theoretic parts are out of print principles of adaptive filters and method of repression noise in acoustic recordings. In practical parts come out proposal and system simulation for adaptive noise rejection in speech.

Keywords

Adaptive filter, LMS algorithm, noise suppression,

Prohlášení

Prohlašuji, že svou bakalářskou práci na téma Adaptivní potlačení šumu v řeči jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této 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 28. května 2009

.....
podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Petru Syslovy, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne 28. května 2009

.....
podpis autora

Obsah

| | |
|--|----|
| Seznam obrázků..... | 7 |
| Seznam tabulek..... | 7 |
| 1 ADAPTIVNÍ ČÍSLICOVÉ FILTRY..... | 8 |
| 1.1 Úvod..... | 8 |
| 1.2 Wienerova optimální filtrace | 11 |
| 1.3 Základní koncepce metod adaptivní filtrace..... | 13 |
| 2 Algoritmy typu LMS, RLS a jejich varianty..... | 15 |
| 2.1 LMS (Least-Mean Square) | 15 |
| 2.2 NLMS (Normalized Least-Mean Square)..... | 15 |
| 2.4 DCT-LMS (Discrete Cosine Transform - Least Mean Square)..... | 17 |
| 2.5 RLS (Recursive Least Square) | 17 |
| 2.6 FTF (Fast Transversal Filter, rychlý rekurzivní)..... | 18 |
| 3 Řečový signál, šum a možnosti jeho potlačení..... | 19 |
| 4 Simulace v programu Matlab | 22 |
| 4.1 Konstrukce filtru v Matlabu | 22 |
| 4.2 Vliv parametrů filtru, porovnání algoritmů | 25 |
| 4.3 Aplikace filtru na reálné nahrávky..... | 30 |
| ZÁVĚR | 34 |
| Literatura | 35 |
| Obsah CD:..... | 35 |

Seznam obrázků

| | | |
|------|--|----|
| 1.1 | Graf signálových toků číslicového filtru typu FIR v 2. kanonické struktuře | 8 |
| 1.2 | Blokové schéma obecného zapojení adaptivního číslicového filtru | 9 |
| 1.3 | Schéma adaptivního filtru s transversální strukturou | 13 |
| 1.4 | Blokové schéma adaptivního systému | 13 |
| 1.5 | Příklad průběhu kritériální funkce | 14 |
| 2.1 | Graf signálových toků procesu váhové adaptace LMS algoritmu | 15 |
| 2.2 | Blokové schéma FLMS algoritmu | 16 |
| 2.3 | Blokové schéma DCT-LMS algoritmu | 17 |
| 2.4 | Blokové schéma FTF algoritmu | 18 |
| 3.1 | Příklad časového průběhu řečového signálu | 19 |
| 3.2 | Jednakanálové systémy | 20 |
| 3.3 | Dvoukanálové systémy | 20 |
| 3.4 | Vícekanálové systémy | 20 |
| 4.1 | Porovnání vstupního a výstupního signálu filtru | 23 |
| 4.2 | Vliv konstanty μ na proces adaptivní filtrace algoritmu LMS | 26 |
| 4.3 | Vliv konstanty μ na proces adaptivní filtrace algoritmu NLMS | 27 |
| 4.4 | Vliv délky filtru na proces adaptivní filtrace algoritmu LMS | 28 |
| 4.5 | Vliv délky filtru μ na proces adaptivní filtrace algoritmu LMS | 29 |
| 4.6 | Filtrace signálu se zvukem vrtačky v pozadí pomocí LMS algoritmu | 30 |
| 4.7 | Filtrace signálu se zvukem vysavače v pozadí pomocí LMS algoritmu | 31 |
| 4.8 | Filtrace signálu se zvukem vrtačky v pozadí pomocí NLMS algoritmu | 31 |
| 4.9 | Filtrace signálu se zvukem vysavače v pozadí pomocí NLMS algoritmu | 32 |
| 4.10 | Blokové schéma upraveného systému za použití spektrálního odečítání | 32 |
| 4.11 | Výsledek filtrování za použití spektrálního odečítání | 33 |

Seznam tabulek

| | | |
|-----|--|----|
| 3.1 | Základní klasifikace algoritmů pro potlačování šumu u řečových signálů | 21 |
|-----|--|----|

1 ADAPTIVNÍ ČÍSLICOVÉ FILTRY

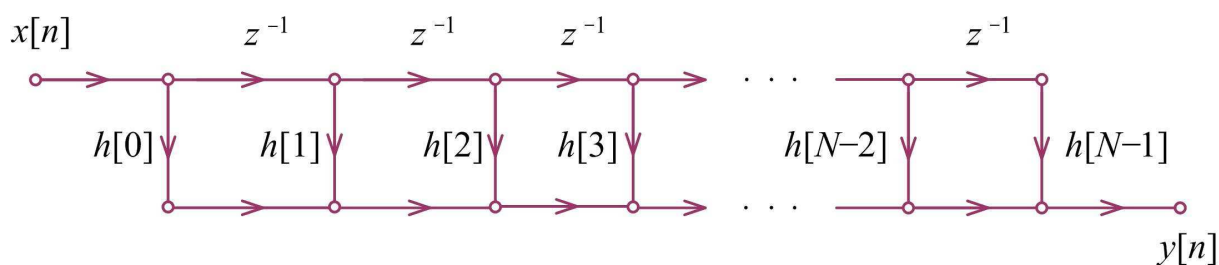
1.1 Úvod

Adaptivní číslicové filtry se velmi často využívají v telekomunikační technice pro potlačení jak elektrických tak i akustických ozvěn. Také se používají pro potlačení šumu a hluku při telefonních hovorech v automobilu, v letadle, v telefonní budce blízko hlučné křižovatky apod. Nejpoužívanější je algoritmus typu LMS, který vychází z optimální Wiesnerovy filtrace.

Pojem adaptabilita vyjadřuje schopnost živého organismu, nějakého orgánu nebo systému přizpůsobit se změněným podmínkám. Jako *adaptivní systém* je označován takový systém, který přizpůsobuje své vlastnosti měnícím se okolním podmínkám, v nichž pracuje, a to tak, aby své vlastnosti udržoval nebo dokonce i zlepšoval. Tato činnost směřuje k dosažení optimálního chování systému. Proto problémy adaptability mohou úzce souviset s problémy optimality.

Jestliže adaptivní systém využívá k udržování nebo zlepšování svých vlastností předchozích zkušeností, nazývá se obvykle *učícím se systémem*. Učící se systém musí při své činnosti zaznamenávat změny podmínek, v nichž pracuje, zvažovat minulé situace a využívat získaných zkušeností k novému rozhodnutí o své další činnosti.

Pojem filtr je používán k označení obvodu nebo systému, který má vybrat užitečnou informaci ze vstupního signálu. Adaptivního filtru můžeme například použít k optimální filtraci nebo k predikci (předpovědi). Většinou předpokládáme, že vstupní signál adaptivního filtru tvoří směs užitečného signálu a šumu a my se filrací snažíme šum odstranit. Jestliže všechny operace filtru jsou lineární, tak hovoříme o lineární filtraci. Pokud použijeme lineární filtr a nežádoucí šum je přičten k užitečnému vstupnímu signálu (jedná se o aditivní šum), pak lze pro optimální filtraci použít *Wienerův filtr*. V tom případě ještě musí platit podmínka, že vstupní signál a šum jsou ze statistického hlediska stacionární.

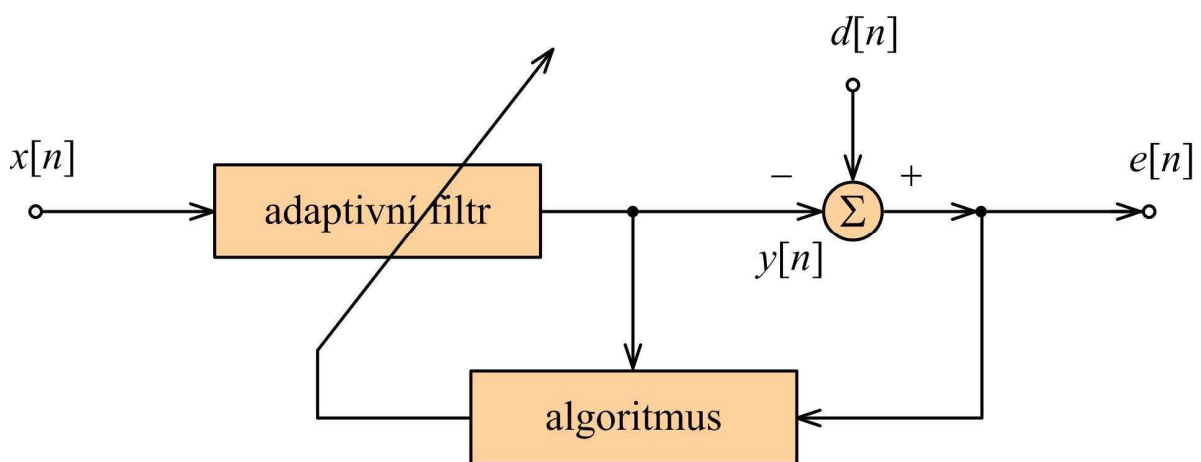


Obrázek 1.1: Graf signálových toků číslicového filtru typu FIR v 2. kanonické struktuře.

Definujme chybový signál $e[n]$ jako rozdíl

$$e[n] = d[n] - y[n], n = 1, 2, 3, \dots \quad (1.1)$$

Signál $d[n]$ je signál, který požadujeme (např. čistý vstupní signál bez šumu) a výstupní signál $y[n]$ získáme na výstupu číslicového filtru. Optimální Wienerův filtr je nastaven tak, že filtruje vstupní signál s minimálním výkonem chybového signálu, neboli s minimální hodnotou střední kvadratické hodnoty chybového signálu $e[n]$. Nejprve je ovšem nutné vypočítat optimální hodnoty impulsní charakteristiky $h_{\text{opt}}[n]$ Wienerova filtru na základě výpočtu statistických vlastností vstupního signálu. Norbert Wiener při odvození optimální filtrace používal spojité signály a systémy. Jeho teorie byla N. Levinsonem upravena i pro diskrétní případ. V této kapitole se budeme zabývat pouze popisem Wienerova filtru pro diskrétní a číslicové signály a systémy. Při optimální filtraci bylo použito číslicového filtru typu FIR v 2. kanonické formě podle obr. 1.1. Po výpočtu hodnot impulsní odezvy je realizována lineární filtrace s tím předpokladem, že statistické vlastnosti vstupního signálu se již nebudou měnit.



Obrázek 1.2: Blokové schéma obecného zapojení adaptivního číslicového filtru.

Cílem nastavení hodnot impulsní odezvy $h[n]$ adaptivního číslicového filtru je dosažení ustáleného stavu, v němž chybový signál $e[n]$ vyhovuje nějaké podmínce (účelové funkci). Na obr. 1.2 je vidět základní schéma adaptivního filtru. Někdy je toto blokové schéma označováno jako zapojení obecného estimatoru [1]. Vstupní signál adaptivního filtru je označen jako $x[n]$ a jeho výstupní signál je $y[n]$. Tento výstupní signál má co nejlépe aproximovat žádaný signál $d[n]$ podle podmínky, která je obsažena v účelové funkci. Přenosovou funkci $H(z)$, která odpovídá impulsní odezvě $h[n]$, lze zvolit jako obecnou racionální lomenou funkci s póly a nulovými body. Adaptivní filtr může například vyhovovat stejné podmínce jako Wienerův filtr [1], tj. dosažení minimálního výkonu signálu $e[n]$. Podstatný rozdíl mezi Wienerovým optimálním filtrem a adaptivním filtrem je v tom, že u adaptivního filtru se neprovádí výpočet statistických vlastností signálu před filtrací, ale adaptivní filtr začíná filtrovat z nějakého počátečního stavu a postupně se pomocí adaptace dostane do stavu optimální filtrace. Adaptivní filtr tedy vychází z vhodně nastavených počátečních podmínek a není u něho požadována předběžná znalost statistických vlastností zpracovávaného signálu jako u Wienerova filtru. Pomocí rekurzivního algoritmu jsou měněny koeficienty jeho impulsní charakteristiky tak, aby byla splněna zadaná podmínka. Tou podmínkou může být dosažení optimální filtrace jako u Wienerova filtru, ale také to být nemusí a může být nastavena podmínka jiným způsobem.

Operace probíhající při číslicové filtraci řízené adaptivní algoritmem zahrnují dva vzájemně svázané procesy:

- Číslicová filtrace, která v případě lineárního číslicového filtru s konstantními koeficienty udává vztah mezi vstupním a výstupním signálem pomocí diskrétní konvoluce

$$\begin{aligned} y[n] &= h[0]x[n] + h[1]x[n-1] + \dots + h[N-1]x[n-N+1] \\ &= \sum_{m=0}^{N-1} h[m]x[n-m] \end{aligned} \quad (1.2)$$

Proces adaptace, jehož cílem je iterativně nastavovat koeficienty adaptivního filtru (např. hodnoty impulsní charakteristiky) tak, aby filtrace splňovala podmínku účelové funkce.

V literatuře je popsáno velké množství adaptivních rekurzivních algoritmů, které byly odvozeny pro různé případy využití adaptivního filtru. Výběr vhodného rekurzivního algoritmu je ovlivněn několika hledisky:

- *Rychlost konvergence.* Rychlost konvergence je určena počtem iterací (opakování), které rekurzivní algoritmus musí provést k tomu, aby bylo dosaženo optimálního stavu adaptivní filtrace. Například v případě adaptivního filtru realizujícího Wienerovu filtraci je dosaženo konečného stavu tehdy, když je minimalizována střední kvadratická chyba.
- *Chyba konvergence.* Chyba je dána rozdílem mezi optimální a skutečnou hodnotou chyby. Např. opět při realizaci Wienerovy filtrace je chyba určena rozdílem mezi hodnotou střední kvadratické chyby při optimální Wienerově filtraci a střední kvadratickou chybou dosaženou pomocí adaptivního filtru.
- *Kvalita sledování.* Když adaptivní filtr zpracovává nestacionární signál, pak tento parametr popisuje míru sledování změn statistických vlastností vstupního signálu. Tento činitel souvisí s rychlostí konvergence a numerickými vlastnostmi adaptivního algoritmu.
- *Robustnost.* Parametr se vztahuje k schopnosti algoritmu pracovat uspokojivě i se špatně zadanými počátečními podmínkami. Robustnost úzce souvisí s numerickými vlastnostmi daného algoritmu.
- *Výpočetní náročnost.* Zde jsou kladeny podmínky na výpočetní výkonnost technického prostředku (počítače, mikroprocesoru apod.), v kterém bude adaptivní algoritmus implementován. Jedná se o počet aritmetických operací (sčítání, násobení apod.), které je nutné realizovat pro výpočet algoritmu při jedné iteraci. Dále je nutné sledovat, jak velkou paměť výpočet zabere a jaká je obtížnost zápisu adaptivního algoritmu v daném programovacím jazyce, který je použit (assembler, jazyk C atd.).
- *Numerické vlastnosti.* Při číslicové realizaci vznikají kvantovací chyby, které mohou způsobit problémy při ustálení adaptivního algoritmu pro optimální řešení. Tyto numerické vlastnosti jsou hlavně hodnoceny pomocí numerické stability a numerické

přesnosti. Algoritmus, který má nulovou chybu konvergence, může vlivem kvantovacích chyb vykazovat nenulovou chybu mezi optimální a skutečnou hodnotou. Hovoříme-li o numerické robustnosti algoritmu, je tím myšlena skutečnost, že jeho vlastnosti nejsou příliš citlivé na změnu délky slova při kvantování.

- *Struktura algoritmu.* Jakou strukturu algoritmus má, nás zajímá zvláště tehdy, když chceme provést hardwarovou implementaci pomocí např. VLSI integrovaných obvodů. Pak se zajímáme o to, zda se algoritmus vyznačuje modulovou strukturou, zda některé jeho části lze zpracovat paralelně apod.

1.2 Wienerova optimální filtrace

Norbert Wiener zvolil pro realizaci transversální filtr, což v naší terminologii v diskrétním případě znamená, že jde o 2. kanonickou strukturu číslicového filtru typu FIR (obr. 1.1). Bylo dokázáno, že v případě statisticky stacionárního signálu na vstupu filtru je střední kvadratická chyba $E [e[n]^2]$ funkcí druhého řádu hodnot impulsní charakteristiky $h [n]$. V teorii pravděpodobnosti pro stacionární diskrétní náhodné procesy je operátor $E[x]$ nazýván střední hodnotou (matematickou nadějí) a je definován takto:

$$E[x] = \sum_k x_k p(x_k)$$

kde $p(x_k)$ představuje pravděpodobnost výskytu hodnoty x_k .

N. Wiener dokázal, že závislost mezi střední kvadratickou chybou a hodnotami impulsní charakteristiky má geometrický tvar vícerozměrného paraboloidu, jehož vrchol definuje optimální hodnoty impulsní charakteristiky $h_{opt} [n]$. Dimenze (počet rozměrů) prostoru, v němž je paraboloid definován, je dán počtem koeficientů impulsní charakteristiky. Definujme vektor vstupního signálu $x [n]$ ve tvaru:

$$x [n] = [x [n] \ x [n - 1] \ x [n - 2] \dots \ x [n - N + 1]]^T \quad (1.3)$$

kde symbol T vyjadřuje transponování matice nebo vektoru. Pro jednoduchost předpokládáme, že všechny signály mají své hodnoty reálné. V obecném případě komplexních signálů by místo prostého transponování matice či vektoru byla použita operace Hermitova transponování (kombinace transponování s komplexním sdružením).

Vektor hodnot impulsní charakteristiky h je :

$$h = [h [0] \ h [1] \ h [2] \dots \ h [N - 1]]^T \quad (1.4)$$

Operaci konvoluce (1.2) lze zapsat v maticovém tvaru následovně:

$$y[n] = [h[0] \ h[1] \ h[2] \dots \ h[N - 1]] \begin{bmatrix} x[n] \\ x[n - 1] \\ x[n - 2] \\ \dots \\ x[n - N + 1] \end{bmatrix} = h^T x(n) = x^T(n) h \quad (1.5)$$

Pro nalezení optima Wienerova filtru je definována účelová funkce:

$$J(\mathbf{h}) = E[|e[n]|^2] = E[|d[n] - \mathbf{h}^T \mathbf{x}[n]|^2] \quad (1.6)$$

Tuto účelovou funkci $J(\mathbf{h})$ minimalizujeme (na nejmenší hodnotu střední kvadratické chyby). Jako výsledek je získána *Wienerova-Hopfova rovnice*

$$\mathbf{R} \cdot \mathbf{h}_{\text{opt}} = \mathbf{p} \quad (1.7)$$

Matrice \mathbf{R} je definována jako autokorelační matice ve tvaru

$$R = E[\mathbf{x}[n]\mathbf{x}^T[n]] = \begin{bmatrix} E[x^2[n]] & E[x[n-1]x[n]] & \dots & E[x[n-N+1]x[n]] \\ E[x[n]x[n-1]] & E[x^2[n-1]] & \dots & E[x[n-N+1]x[n-1]] \\ E[x[n]x[n-2]] & E[x[n-1]x[n-2]] & \dots & E[x[n-N+1]x[n-2]] \\ \dots & \dots & \dots & \dots \\ E[x[n]x[n-N+1]] & E[x[n-1]x[n-N+1]] & \dots & E[x^2[n-N+1]] \end{bmatrix}$$

V případě stacionárních náhodných procesů je tato autokorelační matice \mathbf{R} *Toeplitzovou maticí*, která je regulární a má tedy inverzi \mathbf{R}^{-1} . Sloupcový vektor \mathbf{p} vyjadřuje vzájemnou souvislost mezi hodnotami vstupního signálu $x[n]$ a žádanou hodnotou

$$d[n]\mathbf{p} = E[x[n]d[n]] \quad (1.8)$$

Tato rovnice řeší optimální nastavení koeficientů impulsní charakteristiky číslicového filtru typu FIR v 2. kanonické struktuře po úpravě takto:

$$\mathbf{h}_{\text{opt}} = \mathbf{R}^{-1}\mathbf{p} \quad (1.9)$$

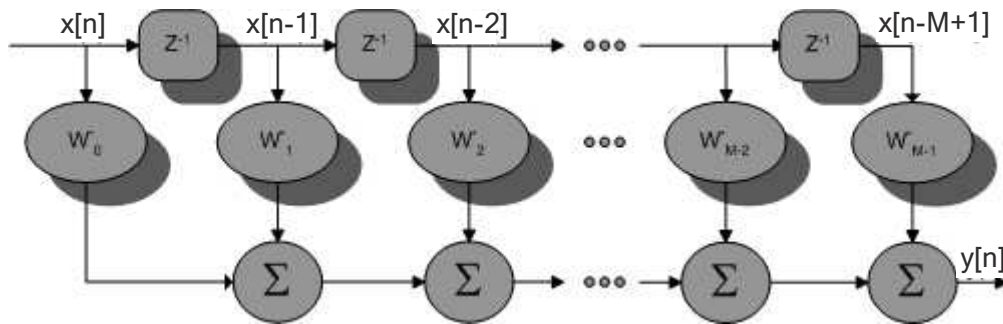
kde \mathbf{R}^{-1} představuje inverzní matici k matici \mathbf{R} .

Výpočtem Wienerovy-Hopfovy rovnice získáme optimální koeficienty impulsní charakteristiky $\mathbf{h}_{\text{opt}}[n]$, které odpovídají v geometrické interpretaci souřadnicím vrcholu paraboloidu. Po výpočtu optimálních koeficientů je nastaven filtr a je realizována číslicová filtrace.

U adaptivního filtru je postup filtrace odlišný. Neprovádí se předběžný výpočet, ale koeficienty filtru jsou nastaveny na počáteční hodnoty a pomocí adaptačního algoritmu je jejich hodnota postupně měněna, až je dosaženo optimálního stavu. Pro aproximaci Wienerova postupu optimální filtrace může být například použit adaptivní algoritmus typu LMS (Least Mean Square).

1.3 Základní koncepce metod adaptivní filtrace

Hlavní myšlenka všech adaptivních filtrů spočívá v kontinuální změně parametrů podle předem stanoveného algoritmu s využitím jak informace ze vstupu tak i pomocné informace přicházející většinou z výstupu celého systému. Adaptivním systémem (obr. 1.3) pak nazýváme systém, jehož jádrem je právě adaptivní filtr spolu se zpětnovazebním mechanismem přenosu informace. Ve většině případů se používají lineární FIR filtry s transverzální strukturou (obr. 1.3) a to z důvodů jednoduchosti implementace a nenáročnosti matematického aparátu.



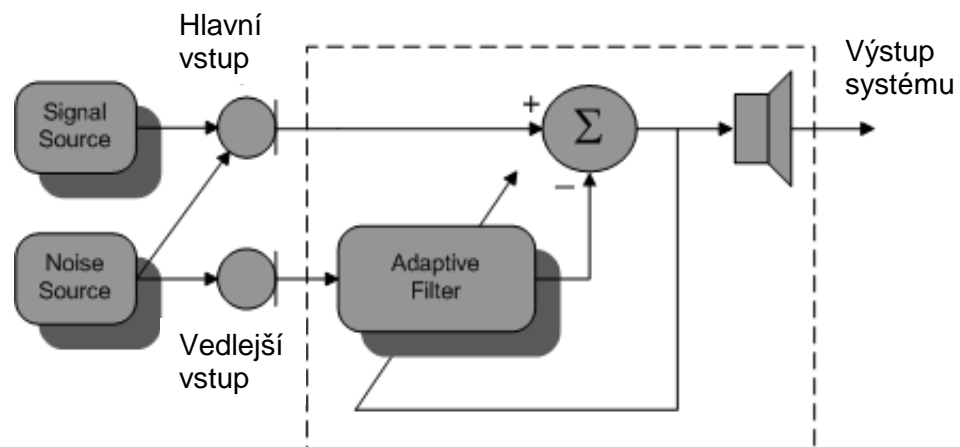
Obrázek 1.3: Schéma adaptivního filtru s transverzální strukturou

Základní výhodou FIR filtrů je jejich neomezená stabilita a konečná doba trvání impulzní odezvy. A samozřejmě již dříve zmiňovaná jednoduchost implementace. Kromě transverzální filtrační struktury se u adaptivních systémů často používají také struktury typu Lattice a Systolická pole.

Výstupní signál adaptivního transverzálního filtru z obr. 1.3 lze zapsat jako

$$y[n] = \sum_{k=0}^{M-1} w_k u[n-k], n = 1, 2, 3, \dots \quad (1.10)$$

kde w_k jsou tzv. váhy filtru, $x[n]$ je označení pro vektor vstupního signálu a $y[n]$ je výstup. Jedná se o filtr M -tého řádu, proto rozsah indexace od 0 do $M-1$.



Obrázek 1.4: Blokové schéma adaptivního systému

Matematický popis adaptivního systému pro potlačování šumu tvoří soubor následujících vztahů (v maticovém tvaru):

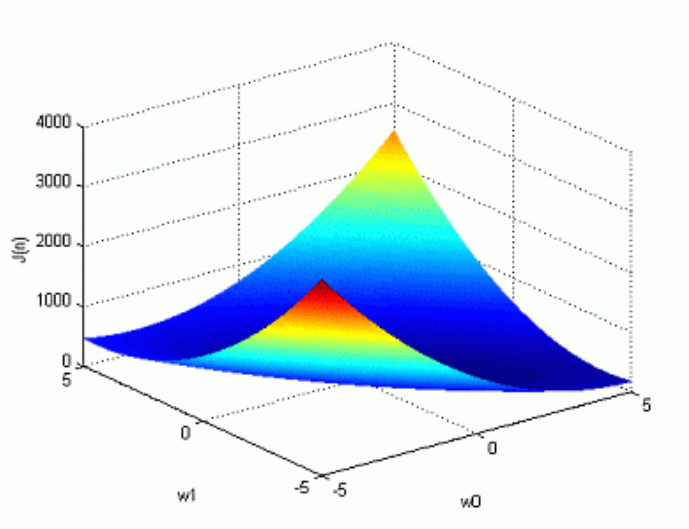
$$w[n] = [w_0(n), w_1(n), \dots, w_{M-1}(n)]^T \quad - \text{vektor vah transverzálního filtru} \quad (1.11)$$

$$y[n] = w[n]^H u[n] \quad - \text{výstupní řečový signál} \quad (1.12)$$

$$e[n] = d[n] - y[n] = d[n] - w[n]^H u[n] \quad - \text{chybový signál} \quad (1.13)$$

$$J[n] = E\{e[n]e^*[n]\} = E\{|e[n]|^2\} \quad - \text{kriteriální funkce} \quad (1.14)$$

kde $d[n]$ je žádaná odezva a $e[n]$ je chybový signál. Operátor $E\{\ast\}$ označuje střední hodnotu



Obrázek 1.5: Příklad průběhu kriteriální funkce

Cílem celého procesu adaptace vah je postupné snižování hodnoty kriteriální funkce až na její minimum. Kriteriální funkce je závislá pouze na hodnotách chybové funkce, tedy je závislá na rozdílu mezi žádanou a skutečnou hodnotou. Čím více se podaří snížit energii šumového signálu obsaženého v řeči, tím menší bude energie výstupního signálu a tím pádem i hodnota kriteriální funkce. Na obr.1.5 je znázorněna závislost kriteriální funkce na vahách adaptivního transverzálního filtru 2. řádu.

2 Algoritmy typu LMS, RLS a jejich varianty

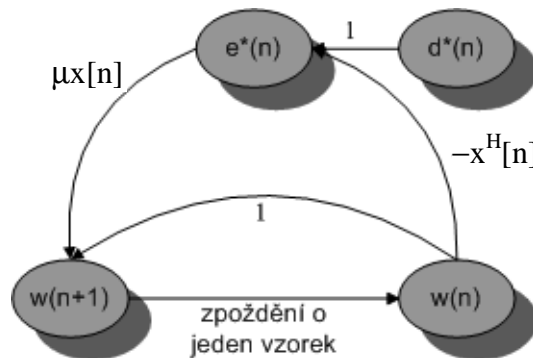
Mezi nejpopulárnější a nejpoužívanější adaptivní algoritmy používané v současné praxi patří algoritmy spadající buď do třídy LMS algoritmů založených na teorii Wienerovy filtrace nebo RLS algoritmů založených na teorii Kalmanovy filtrace.

2.1 LMS (Least-Mean Square)

Síla algoritmu LMS spočívá hlavně v jednoduchosti a matematické nenáročnosti. Adaptivní proces je popsán následujícím vztahem:

$$\hat{w}[n+1] = \hat{w}[n] + \mu x[n]e^*[n] \quad (2.1)$$

Na následujícím obrázku je zobrazen mechanismus adaptace vah ve formě grafu signálových toků:



Obrázek 2.1: Graf signálových toků procesu váhové adaptace LMS algoritmu

2.2 NLMS (Normalized Least-Mean Square)

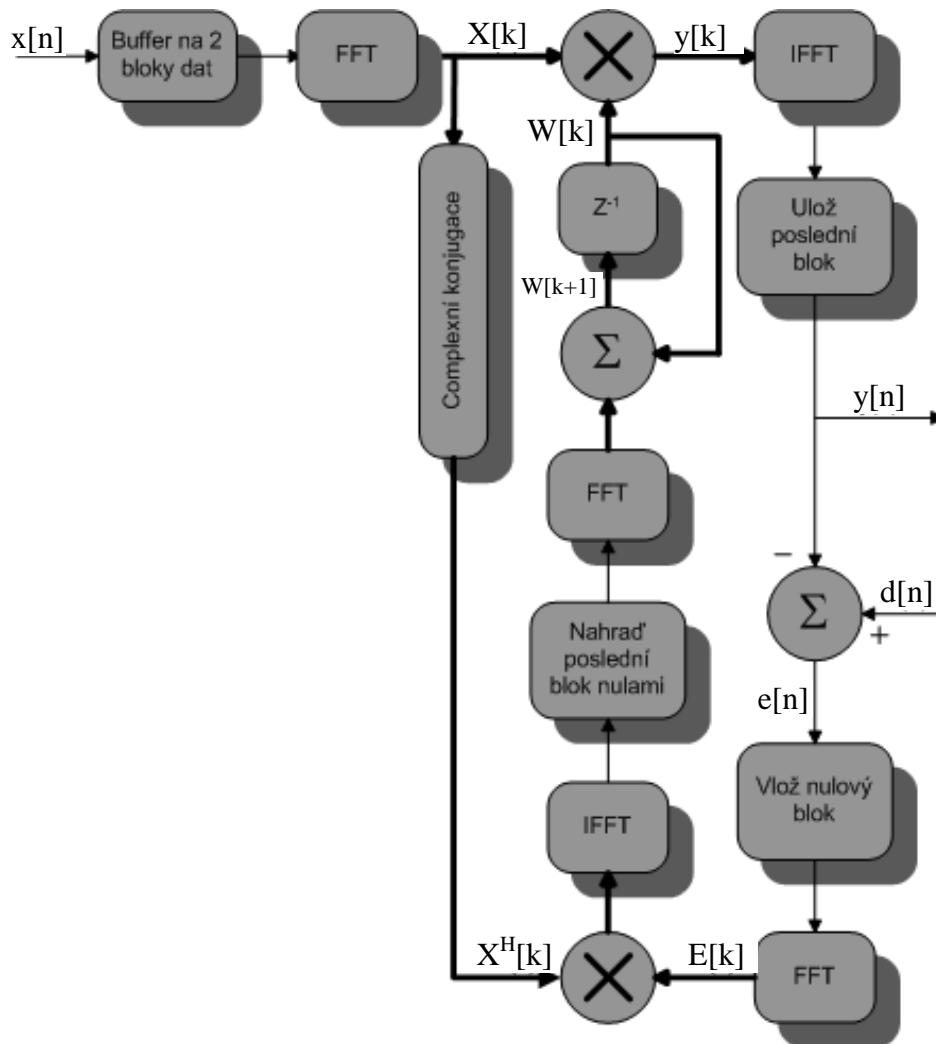
Pokud vstupní řečový signál $x[n]$ nabývá relativně vysokých hodnot, má algoritmus LMS tendenci zesilovat šum. Je to dáno především vztahem pro korekční člen váhové adaptace, resp. jeho přímé závislosti na $x[n]$. Tato závislost se dá jednoduše překonat zavedením tzv. normalizace, z čehož těží právě algoritmus NLMS:

$$\hat{w}[n+1] = \hat{w}[n] + \frac{\mu}{\|x[n]\|^2} x[n]e^*[n] \quad (2.2)$$

kde $\|x[n]\| = \sqrt{\sum_{k=0}^{M-1} |x[n-k]|^2}$ je Euklidovská norma vektoru .

2.3 FLMS (Fast Least-Mean Square)

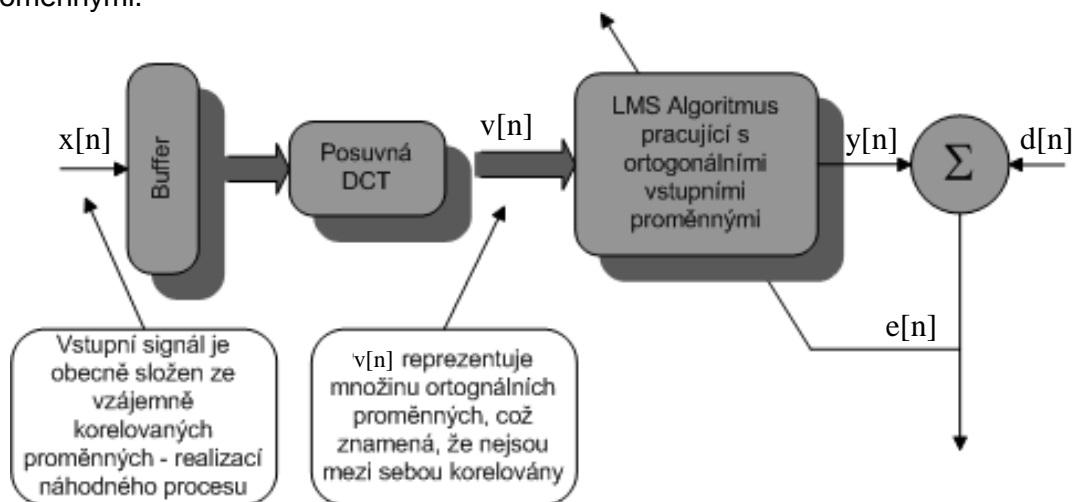
Myšlenkou algoritmu FLMS je vytvoření takové struktury LMS algoritmu, aby se dosáhlo co možná nejefektivnějšího využití DSP technologie při implementaci. Proto jsou při všech operacích se signály používány bloky dat (frames) namísto pouhých vzorků. Blokové schéma algoritmu FLMS (obr. 2.2) je značně odlišné od schématu klasického LMS algoritmu (obr. 1.3). Navíc se zde provádí tzv. "rychlá konvoluce s využitím overlap-save metody", kterou lze velmi efektivně provádět na signálových procesorech.



Obrázek 2.2: Blokové schéma FLMS algoritmu

2.4 DCT-LMS (Discrete Cosine Transform - Least Mean Square)

Název DCT- naznačuje, že klíčovým operátorem tohoto algoritmu je diskrétní kosinová transformace - DCT. Tomuto algoritmu se často říká také samo-se-organizující filtr. Hlavní myšlenkou je transformovat vstupní vektor $x[n]$, který je obecně složen ze statisticky závislých proměnných na jiný vektor, jehož jednotlivé proměnné na sobě budou statisticky nezávislé. Tuto transformaci realizuje právě DCT. Takto upravený vektor se následně použije jako vstup klasického LMS adaptivního algoritmu - viz obr. 2.4. Cílem je dosažení lepších konvergenčních vlastností, než při použití vektoru s vzájemně korelovanými proměnnými.



Obrázek 2.3: Blokové schéma DCT-LMS algoritmu

2.5 RLS (Recursive Least Square)

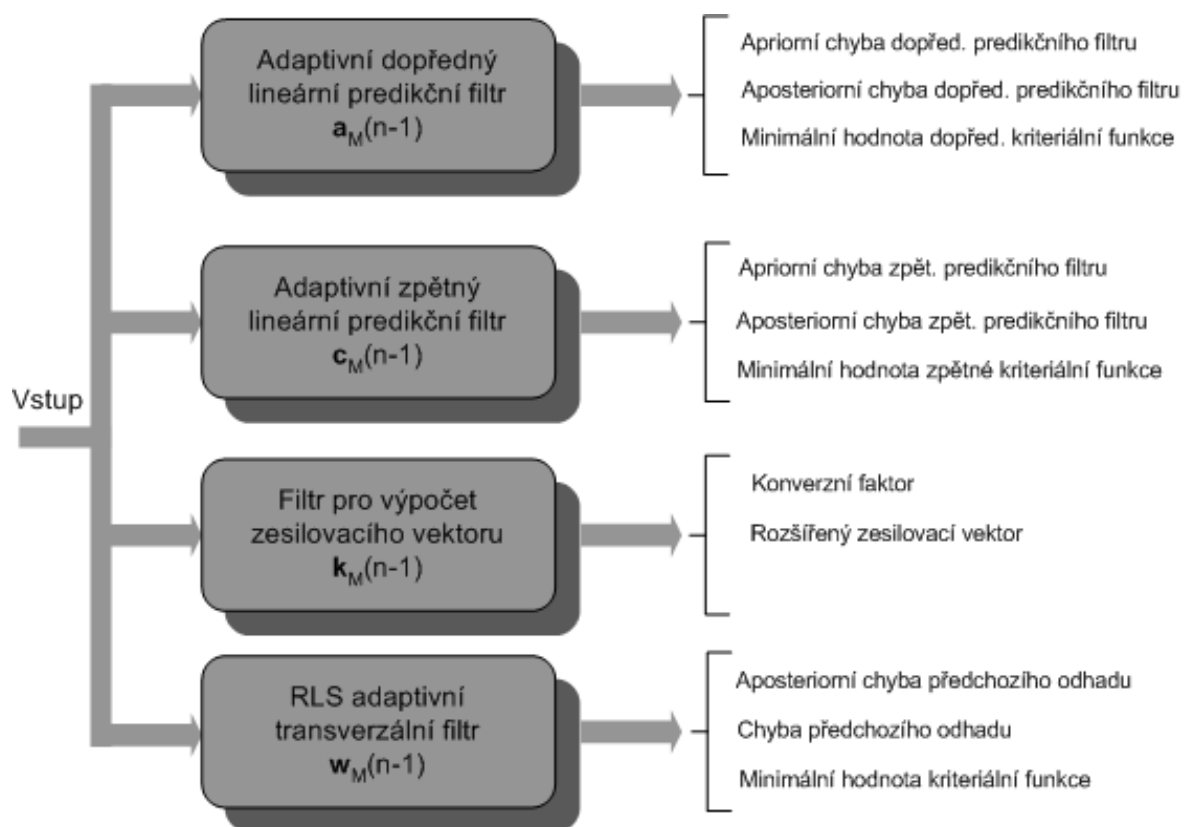
RLS je základním představitelem druhé třídy adaptivních algoritmů - algoritmů vystavených na teorii Kalmanovy filtrace. Základním rozdílem proti rodině LMS algoritmů je vlastní statistické pojetí. Zde se pracuje s průměrnými hodnotami veličin počítanými z časových vývoji namísto vzorkových průměrů počítaných z několika realizací stejného náhodného procesu. Tak např. vztah pro kritériální funkci je následující:

$$J[n] = \sum_{i=M}^N |e[i]|^2 = \sum_{i=M}^N e[i]e^*[i] \quad (2.3)$$

Z tohoto vztahu je patrný princip časového průměrování. Struktura filtru zůstává stejná jako u LMS algoritmů, jen adaptivní proces je odlišný, vzhledem k použití průměrů. Z toho rovněž plyne větší výpočetní náročnost než u LMS algoritmů. Tento rozdíl je ovšem natolik podstatný (RLS úlohy jsou o řád vyšší než LMS úlohy), že často vede k závěrům, že RLS algoritmy nemají praktický význam. Na druhou stranu, pokud se podíváme na rychlost konvergence jednotlivých algoritmů, zjistíme, že u RLS je konvergence několikanásobně rychlejší než u LMS. To plyne z použití časového průměrování, které predikuje velmi přesné hodnoty.

2.6 FTF (Fast Transversal Filter, rychlý rekurzivní)

Jediným důvodem pro návrh tohoto algoritmu je řádové snížení výpočetní náročnosti klasického RLS při zachování rychlosti konvergence. Koncepce vychází strukturálně ze čtyř odlišných filtrů pracujících souběžně na jedné úloze (viz obr. 2.4).



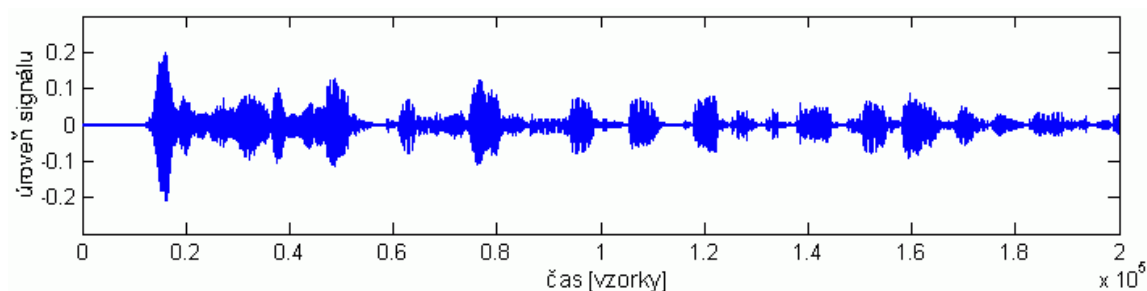
Obrázek 2.4: Blokové schéma FTF algoritmu

3 Řečový signál, šum a možnosti jeho potlačení

Signál na obrázku 3.1 je řečový signál v diskrétní podobě vzorkovaný frekvencí 44,1 kHz. Jedná se o asi 5sekundový úryvek konverzace, pořízený v kabině automobilu. Pro zpracování takových signálů pomocí DSP technologie [2] hraje klíčovou roli jejich vlastní "kvalita". Co to je kvalitní řečový signál, záleží na volbě hodnotícího kritéria. Univerzální kritérium samozřejmě neexistuje, protože řeč je možno posuzovat z několika pohledů (srozumitelnost, přirozenost, rozpoznatelnost, atd..). Hodnotící kritéria lze rozdělit do dvou oblastí - subjektivní (posouzení skupinou posluchačů a nezávislé ohodnocení kvality řečového signálu) a objektivní (kritérium je definováno nejčastěji matematickou rovnicí, příp. algoritmem).

Zde je pouze několik kritérií a testů pro subjektivní a objektivní posouzení kvality řečových signálů.

- subjektivní:
 - MTR (Modifikovaný rýmovací test)
 - DRT (Diagnostický rýmovací test)
 - QUART (Kvalitativní řečový test)
 - DAM (Diagnostický test přijatelnosti)
- objektivní:
 - SNR (Odstup signál/šum)
 - AI (Artikulační index)



Obrázek 3.1: Příklad časového průběhu řečového signálu

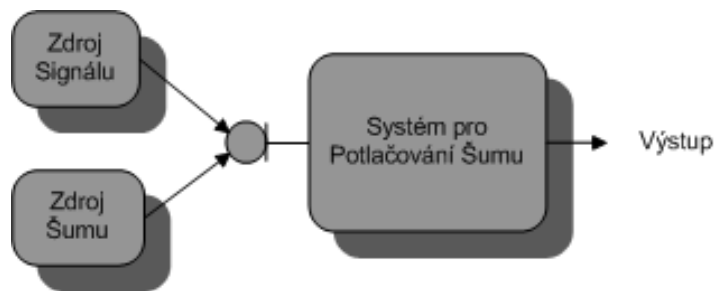
Pro účel této práce se budeme dále zabývat pouze objektivními kritérii, která vedou k jejich přímé aplikaci a jedním z takových kritérií je např. SNR (Signal-to-Noise Ratio),

$$SNR = 10 \log_{10} \frac{s_a[n]}{n_a[n]} \quad (3.1)$$

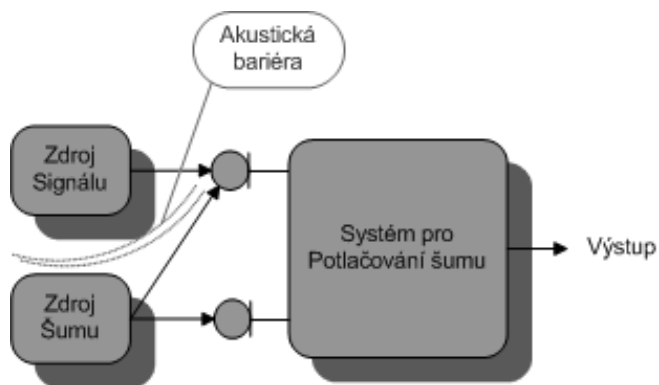
kde s_a je energie užitečného signálu a n_a je energie šumu.

Principů ke zlepšení kvality řečových signálů je několik. Na počátku volby vhodné metody je nutno zvážit o jakou aplikaci se jedná. Rozlišujeme následující systémy:

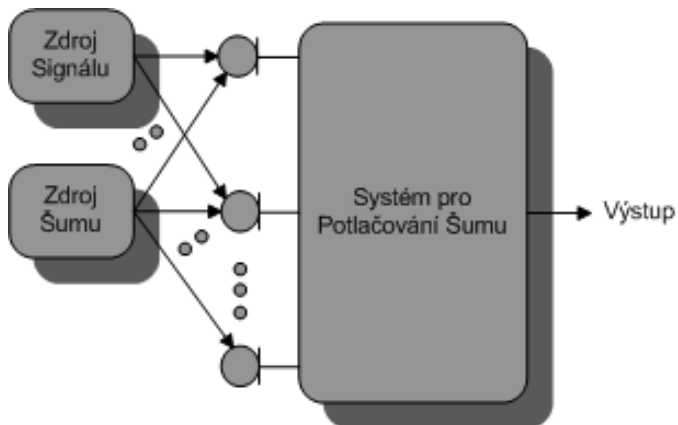
- Jednakanálové systémy
- Dvoukanálové systémy
- Vícekanálové systémy



Obrázek 3.2: Jednokanálové systémy



Obrázek 3.3: Dvoukanálové systémy



Obrázek 3.4: Vícekanálové systémy

Architektura vícekanálových systémů předpokládá prostorové rozložení tří a více senzorů (mikrofonů) a metodika pro zpracování signálů se souhrnně nazývá Array Signal Processing [2]. Jako základní klasifikace algoritmů pro potlačování šumu u řečových signálů může posloužit následující tabulka:

| Jméno třídy algoritmu | Popis | Jedno-/Dvou-kanálové uspořádání |
|---|--|--|
| Krátkodobé spektrálně amplitudové techniky | Pro potlačení šumu se pouze odečte matematicky odhadnutá hodnota šumového prahu v doméně buď Fourierovské, spektrální výkonové nebo autokorelační | jednokanálové dvoukanálové |
| Algoritmy užívající modelování řeči pomocí iterativních principů | Tyto systémy používají k určení řečového modelu ze zašuměného signálu AR, vázaný AR nebo ARMA modely. Následně se provádí tzv. resyntéza signálu zbaveného šumu a to pomocí nekauzální Wienerovy filtrace. | jednokanálové dvoukanálové |
| Adaptivní potlačení šumu (ANC) | ANC je formulován v časové nebo frekvenční doméně a je založen na teoriích Wienerovy a Kalmanovy filtrace. Tyto algoritmy vykonávají rekurzivní aktualizaci svých parametrů. | dvoukanálové |
| Periodicita řečového signálu | Tyto metody sledující základní frekvenci v harmonickém amplitudovém řečovém spektru pomocí buď speciálního typu ANC nebo adaptivního hřebenového filtru. | jednokanálové |

Tabulka 3.1: Základní klasifikace algoritmů pro potlačování šumu u řečových signálů

Pro naše účely jsme zvolili dvou-kanálový systém s adaptivním potlačením šumu. Záznam tedy bude pořizován pomocí 2 mikrofonů pro snímání šumivého a řečového signálu zvlášť a potlačení šumu bude prováděno pomocí adaptivního filtru založeného na algoritmech Wienerovy filtrace

4 Simulace v programu Matlab

Pro simulaci používáme program MATLAB verze R2008a. MATLAB je integrované prostředí pro vědeckotechnické výpočty, modelování, návrhy algoritmů, simulace, analýzu a prezentaci dat, paralelní výpočty, měření a zpracování signálů, návrhy řídicích a komunikačních systémů. Pro tuto simulaci využíváme především následující aplikační knihovny:

Signal Processing Toolbox

Stejně jako ostatní moderní knihovny MATLABu je vybaven grafickým rozhraním, které zjednodušuje jak frekvenční analýzu signálů nebo posloupností, tak jejich úpravy, např. filtraci, modulaci nebo demodulaci. Návrh a analýza filtrů včetně interaktivního zobrazování amplitudových a fázových charakteristik, skupinového zpoždění, nul a pólů, přechodových a impulsních charakteristik patří mezi nejdůležitější schopnosti tohoto nástroje.

Filter Design Toolbox

Sbírka nástrojů, která poskytuje pokročilé techniky pro projektování, simulace a analyzování číslicových filtrů. Rozšiřuje Signal Processing Toolbox architekturami filtrů a metodami výpočtu pro komplexní real-time DSP aplikace, včetně adaptivního a multirate filtrování.

4.1 Konstrukce filtru v Matlabu

Pro konstrukci adaptivních filtrů má Matlab již připravený skript pojmenovaný *adaptfilt*. Implementace skriptu je následující:

```
H = ADAPTFILT.LMS(L, STEPSIZE, LEAKAGE, COEFFS, STATES)
```

V tomto případě je v proměnné H vytvořen objekt adaptivního filtru, který je založen na algoritmu LMS. Parametry v závorkách upřesňují samotné nastavení filtru.

| | |
|----------|--------------------------------|
| L | - řád filtru |
| STEPSIZE | - krok filtrace |
| LEAKAGE | - rozptyl filtru |
| COEFFS | - počáteční koeficienty filtru |
| STATES | - počáteční stavy filtru |

Pokud bychom chtěli filtr založený na jiném algoritmu, jednoduše nahradíme řetězec znaku LMS za jiný algoritmus, např. RLS. Zápis pak bude vypadat následovně:

```
H = ADAPTFILT.RLS(...)
```

Pro samotný výpočet filtrace použijeme funkci pojmenovanou *filter*. Implementace je následující:

```
[Y,E] = FILTER(H,X,D)
```

Tato funkce aplikuje adaptivní filtr H na vstupní signál X (v našem případě samostatný šumový signál) a na užitečný signál D (v našem případě signál + šum).

Vyfiltrovaný signál nalezneme ve výstupní proměnné E.

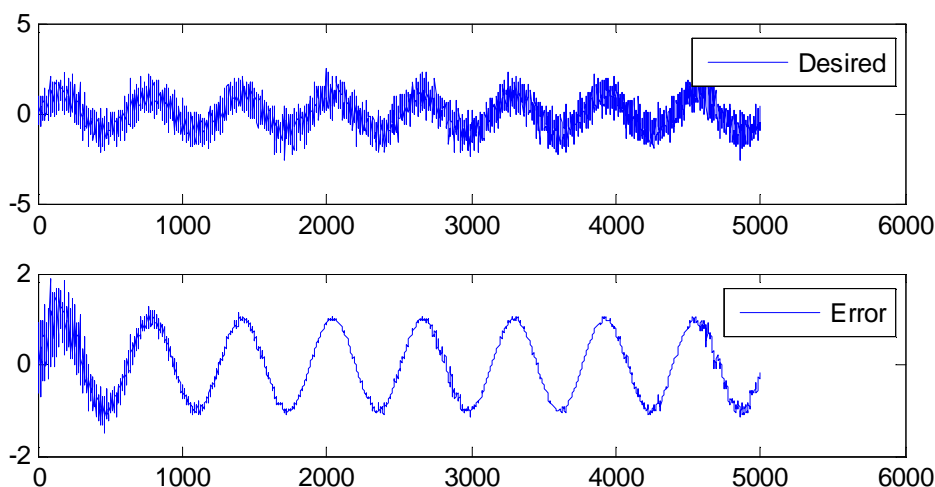
Takto vypadá nejjednodušší zápis v Matlabu:

```
clear all; %Vymaže paměť proměnných
x = sin(0:0.01:50); %Originální signál - sinusový průběh
n = randn(1,5001); %Náhodný šumový signál
d = x + 0.5*n; %Sinusový průběh + šum

step = 0.002; %Nastavení StepSize
l = 10; %Řád filtru
H = adaptfilt.lms(10,step); %Vytvoření objektu filtru
[y,e] = filter(H,n,d); %Aplikace adaptivního filtru
```

Jako užitečný signál (Desired) určený k filtraci jsme použili sinusový průběh, který jsme následně znehodnotili náhodným šumem (Noise). Tyto dva signály slouží jako vstup adaptivního filtru založeného na algoritmu LMS. Po provedení výpočtu dostáváme vyfiltrovaný signál (Error). Na obrázku 4.1 je na začátku patrné zpoždění, než se filtr přizpůsobí, ale potom již odstraňuje šum velice dobře.

Nutno ovšem podotknout, že v tomto případě je šumový signál identický s šumem, který je součástí užitečného signálu (není nijak zpožděný ani fázově posunutý), takže filtr ve své podstatě nedělá nic jiného, než že odečítá šum od pozorovaného signálu. U reálných signálu Desire a Noise se již projevuje vzájemné zpoždění a fázový posuv vlivem metody zaznamenávání signálů.



Obrázek 4.1: Porovnání vstupního a výstupního signálu filtru

Pro lepší pochopení jsme ale sestavili vlastní funkci adaptivního filtru pro algoritmus LMS a NLMS. Zdrojový kód funkcí je následující:

Adaptivní filtr s algoritmem LMS:

```
function [y,e] = lms(d,n,FilterSize,mu)

% %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% % [y,e] = lms(d,n,FilterSize,mu)
% %
% % Funkce adaptivního filtru postaveného na algoritmu LMS
% % -----
-----
% % Vstupní parametry:
% % d - signál obsahující užitečný signál znehodnocený šumem
(hlukem)
% % n - referenční signál obsahující pouze šum (hluk)
% % FilterSize - udává řád (velikost filtru)
% % mu - konstanta "mí" pro výpočet
% %
% % Výstupem této funkce jsou proměnné "y" a "e" které obsahují
odfiltrovaný
% % a chybový signál.
% %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

N = length(d); %zjištění délky vstupního
signálu
w = zeros( FilterSize , 1 ) ; %inicializace proměnných pro
rychlejší zpracování
y = zeros( N , 1 );
e = zeros( N , 1 );
for i = FilterSize : N
    u = n(i:-1:i-FilterSize+1) ;
    y(i)= w' * u; %vypočet aktuálního vzorku
    e(i) = d(i) - y(i) ; %vypočet chyby pro aktuální
vzorek
    w = w + mu * u * e(i) ; %přepočet koeficientů filtru
end
end
```


Adaptivní filtr s algoritmem NLMS:

```
function [y,e] = lms(d,n,FilterSize,mu)

% %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %
% % [y,e] = nlms(d,n,FilterSize,mu)
% %
% % Funkce adaptivního filtru postaveného na algoritmu NLMS
% % -----
-----
% % Vstupní parametry:
% % d - signál obsahující užitečný signál znehodnocený šumem
% %      (hlukem)
% % n - referenční signál obsahující pouze šum (hluk)
% % FilterSize - udává řád (velikost filtru)
% % mu - konstanta "mí" pro výpočet
% %
% % Výstupem této funkce jsou proměnné "y" a "e" které obsahují
% % odfiltrovaný
% % a chybový signál.
% %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

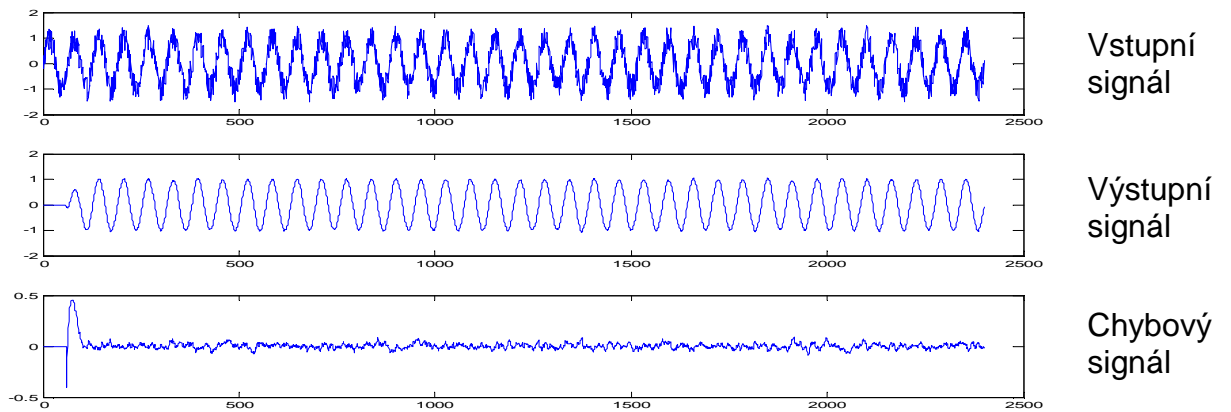
N = length(d); %zjištění délky vstupního
signálu
w = zeros( FilterSize , 1 ) ; %inicializace proměnných pro
rychlejší zpracování
y = zeros( N , 1 );
e = zeros( N , 1 );
for i = FilterSize : N
    u = n(i:-1:i-FilterSize+1) ;
    y(i)= w' * u; %vypočet aktuálního vzorku
    e(i) = d(i) - y(i) ; %vypočet chyby pro aktuální
vzorek
    w = w + (mu/(u'*u) ) * u * e(i) ; %přepočet koeficientů filtru
end
end
```

4.2 Vliv parametrů filtru, porovnání algoritmů

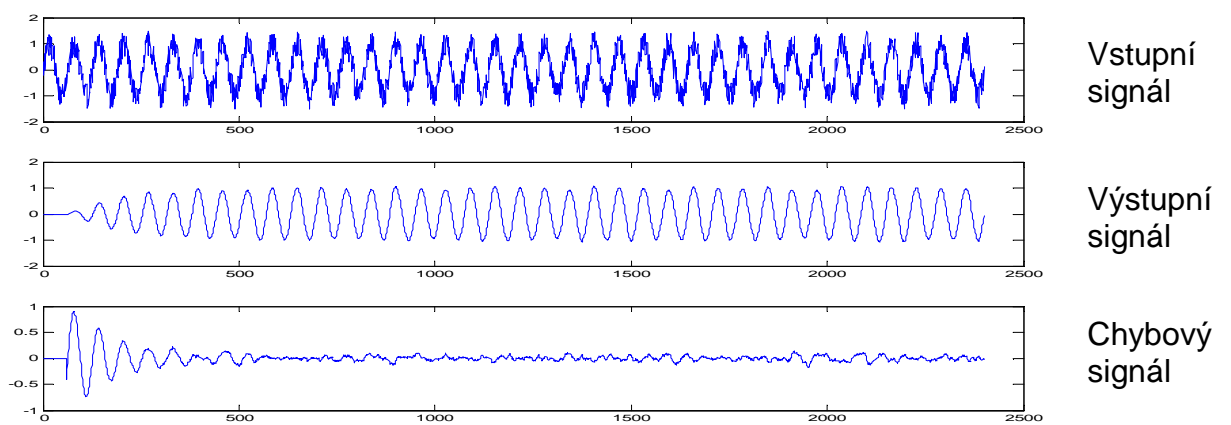
Důležitými vstupními parametry jsou délka filtru (FilterSize), která udává počet koeficientů filtru a konstanta "μ" (mu), která určuje rychlost konvergence. Na následujících případech si ukážeme jejich vliv na proces adaptivní filtrace u dvou algoritmů.

Jako vstupní signál nám bude sloužit sinusový průběh znehodnocený náhodně generovaným šumem. Porovnávat budeme algoritmy LMS a NLMS, délka filtru zůstane konstantní (FilterSize = 60) a budeme měnit konstantu μ (mu).

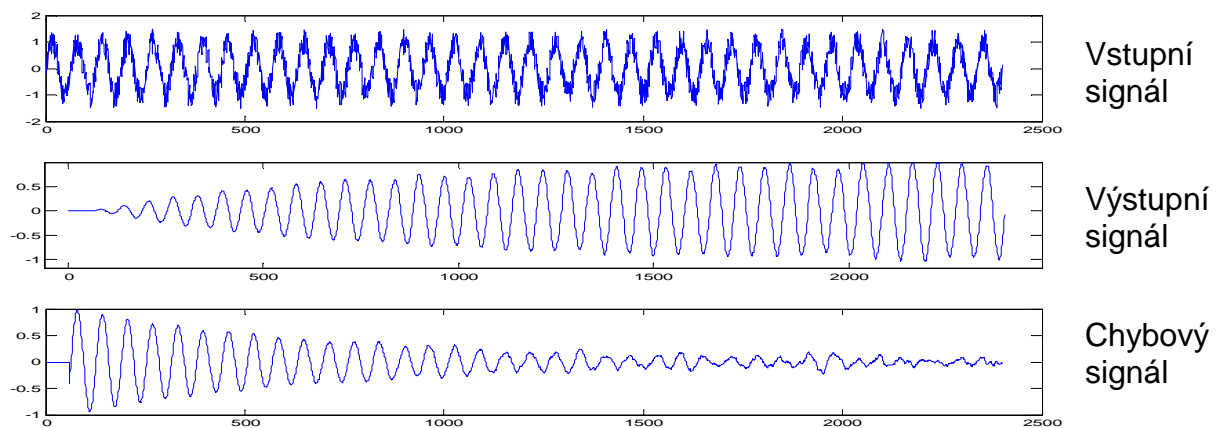
Algoritmus: LMS
 $\mu = 0.005$



$\mu = 0.0005$

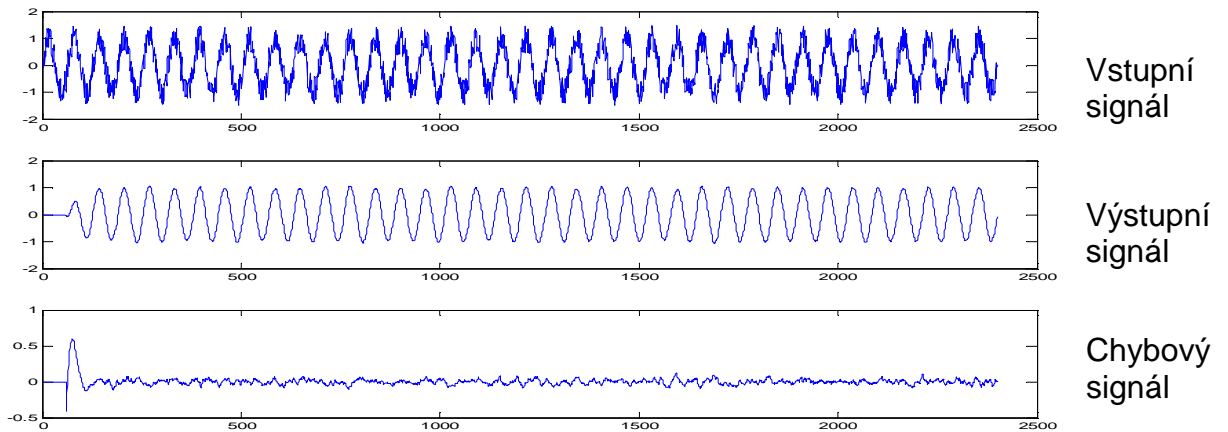


$\mu = 0.0001$

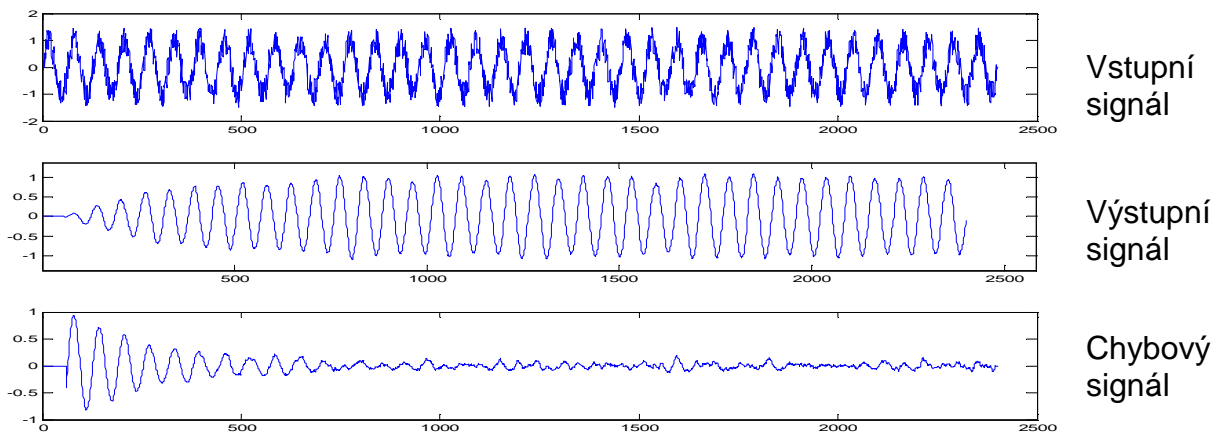


Obrázek 4.2: Vliv konstanty μ na proces adaptivní filtrace algoritmu LMS

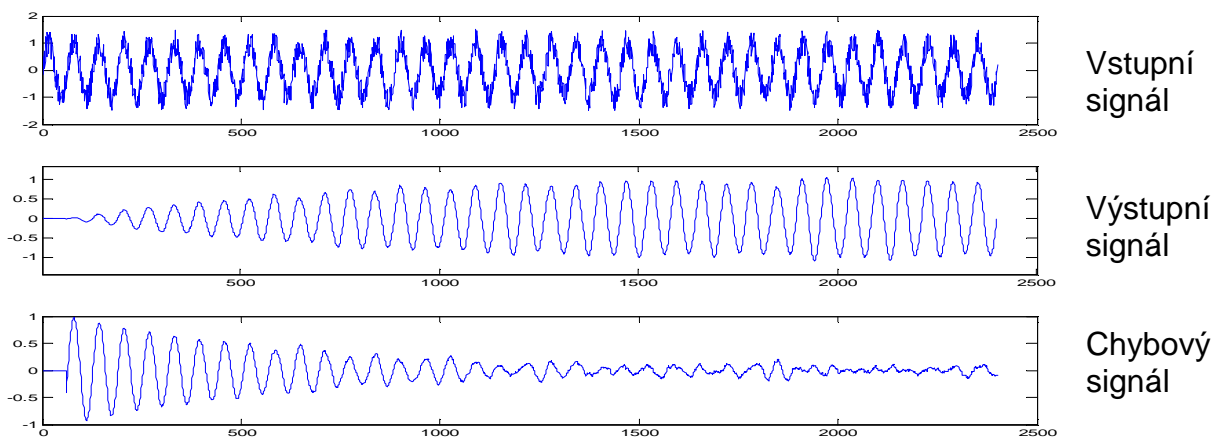
Algoritmus: NLMS
 $\mu = 0.1$



$\mu = 0.01$



$\mu = 0.004$



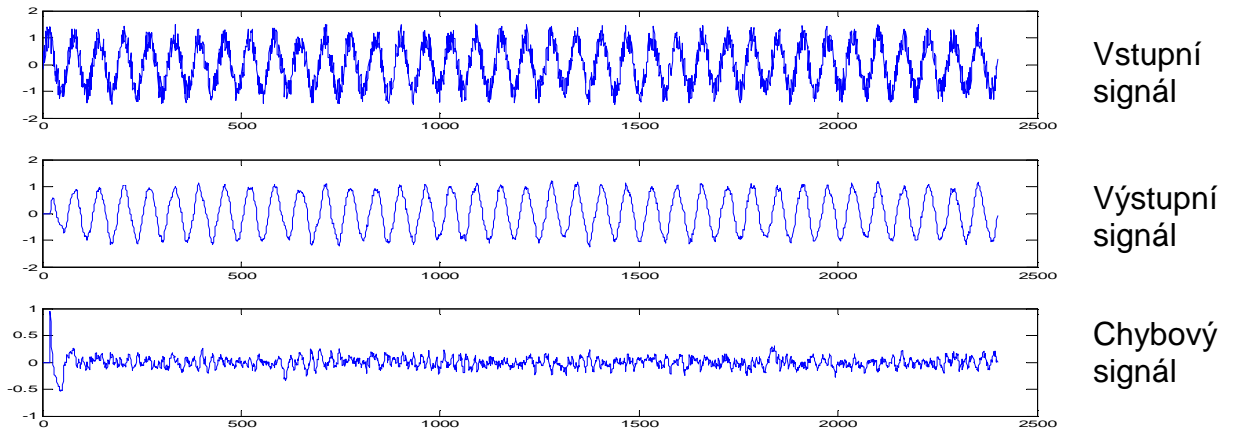
Obrázek 4.3: Vliv konstanty μ na proces adaptivní filtrace algoritmu NLMS

Jak je z předcházejících průběhů vidět, konstanta " μ " (μ) má vliv na rychlost adaptace filtru. Čím je menší, tím pomaleji se filtr přizpůsobuje. Dalším parametrem je délka filtru, která udává počet koeficientů filtru, Opět budeme porovnávat u algoritmů LMS

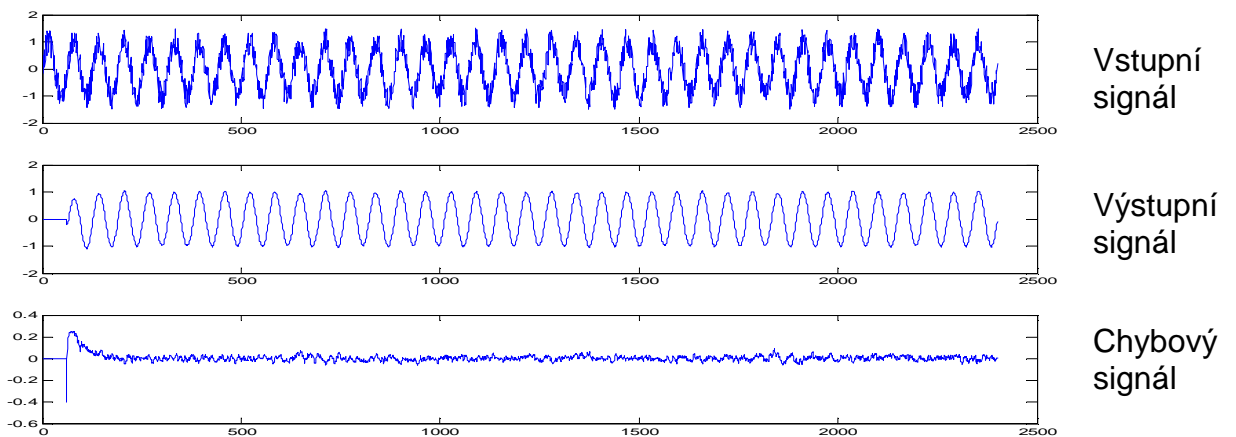
a NLMS, konstanta " μ " (μ) zůstane ve všech případech stejná a bude se měnit délka filtru.

Algoritmus: LMS

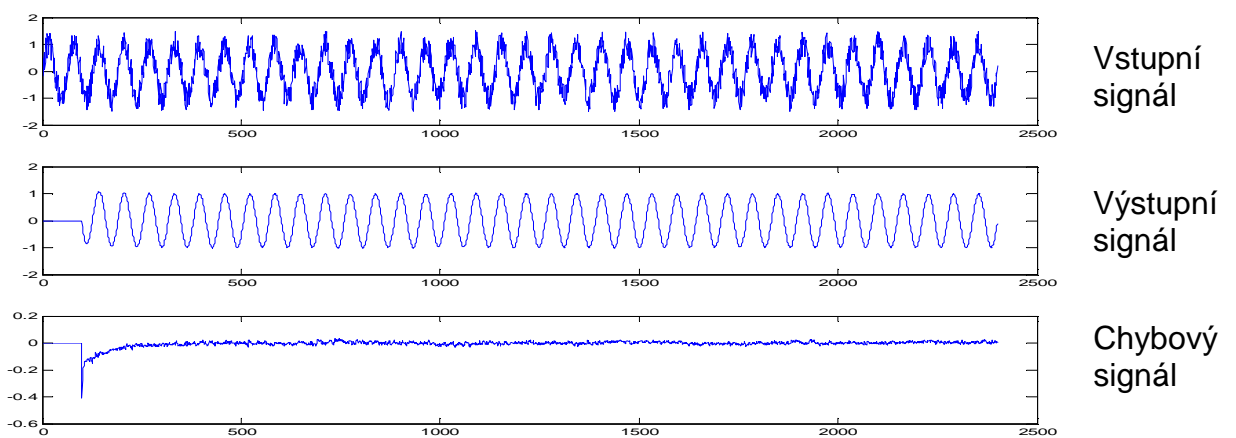
FilterSize = 20



FilterSize = 60

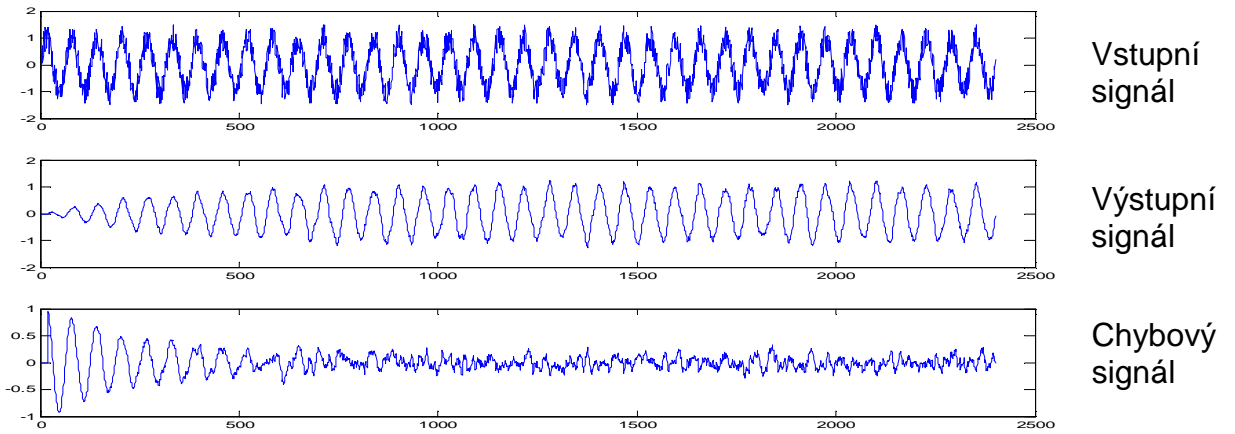


FilterSize = 100

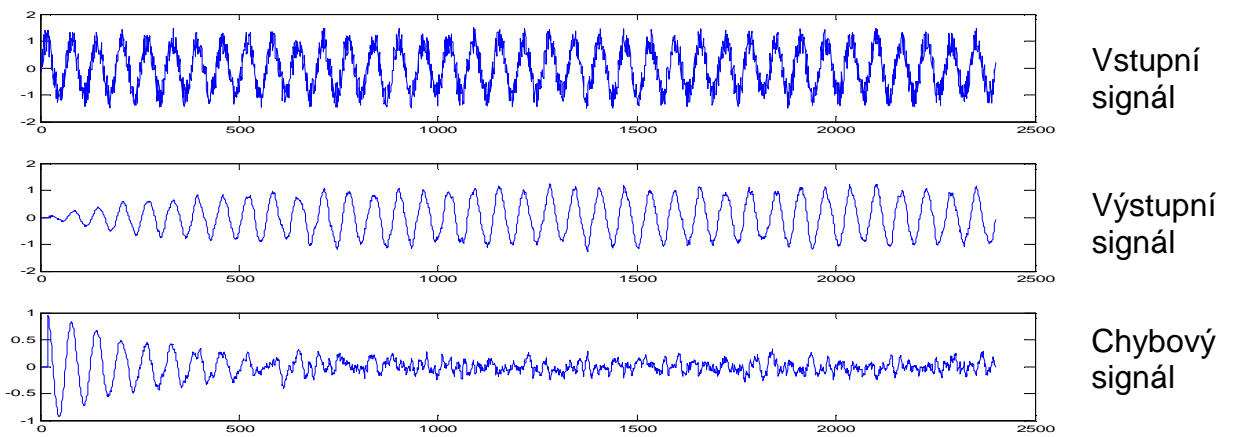


Obrázek 4.4: Vliv délky filtru na proces adaptivní filtrace algoritmu LMS

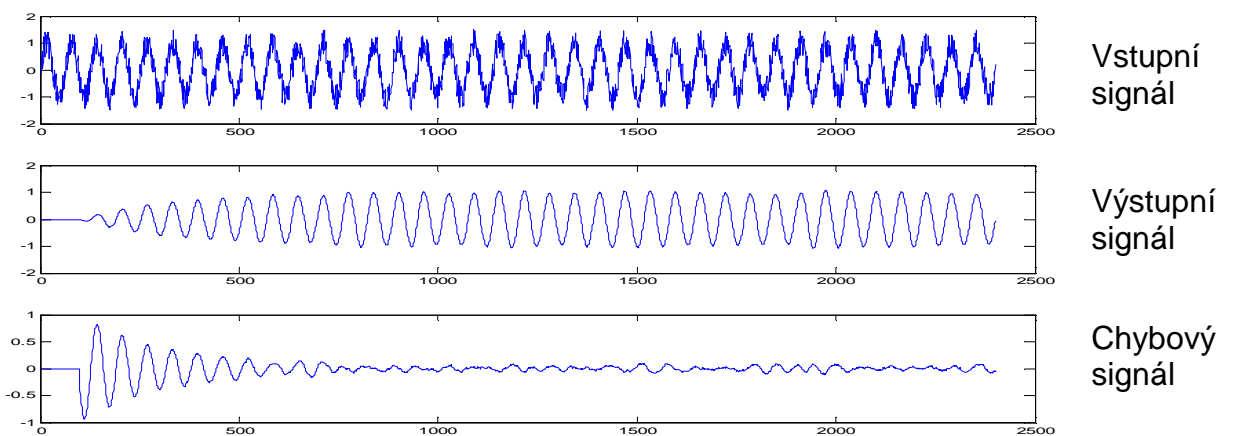
Algoritmus: NLMS
FilterSize = 20



FilterSize = 60



FilterSize = 100



Obrázek 4.5: Vliv délky filtru na proces adaptivní filtrace algoritmu NLMS

Zde je patrné, že délka filtru má vliv na kvalitu filtrování. Čím je filtr delší, tím lépe je schopen rekonstruovat původní signál. Ovšem delší filtr je také náročnější na výpočty, tedy i na hardware.

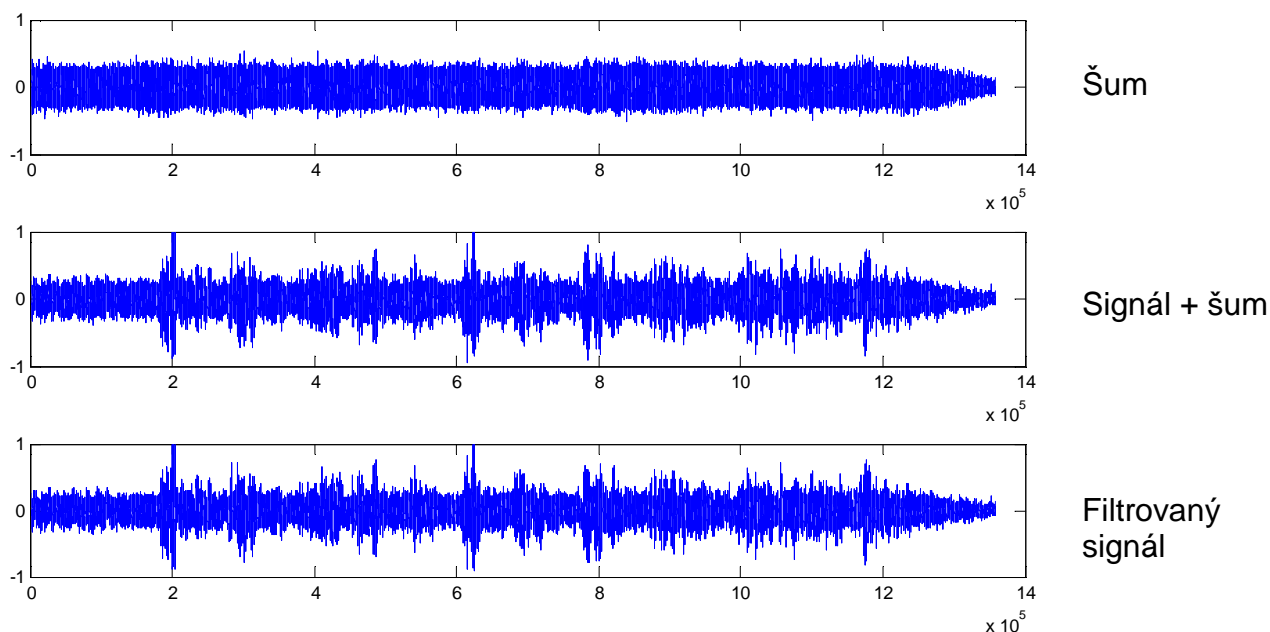
4.3 Aplikace filtru na reálné nahrávky

Abychom otestovali účinnost filtru na reálných nahrávkách, pořídili jsme dva zvukové záznamy v bezodrazové komoře. Jednalo se o čtený souvislý text o délce asi 20 sekund. Jako zdroj rušivého šumu jsme v prvním případě použili vrtačku, v druhém případě vysavač. Pro záznam zvuku jsme použili dva mikrofony, směrový pro zachycení samostatného šumu a všesměrový pro zachycení mluveného slova a šumu. Směrový mikrofon byl umístěn tak, aby snímal pouze okolní šum, tedy v horizontální poloze tak, aby jeho charakteristika směřovala od mluvčího. Všesměrový mikrofon byl umístěn vertikálně, co nejbližší mikrofonu směrovému, abychom co nejvíce eliminovali vzájemné zpoždění obou signálů. Signál z obou mikrofonů byl zaznamenáván na HDD Recorder TASCAM HD-P2 s vzorkovací frekvencí 48kHz jako stereo nahrávka, kde jeden kanál obsahoval mluvené slovo a šum, a druhý kanál samostatný šum.

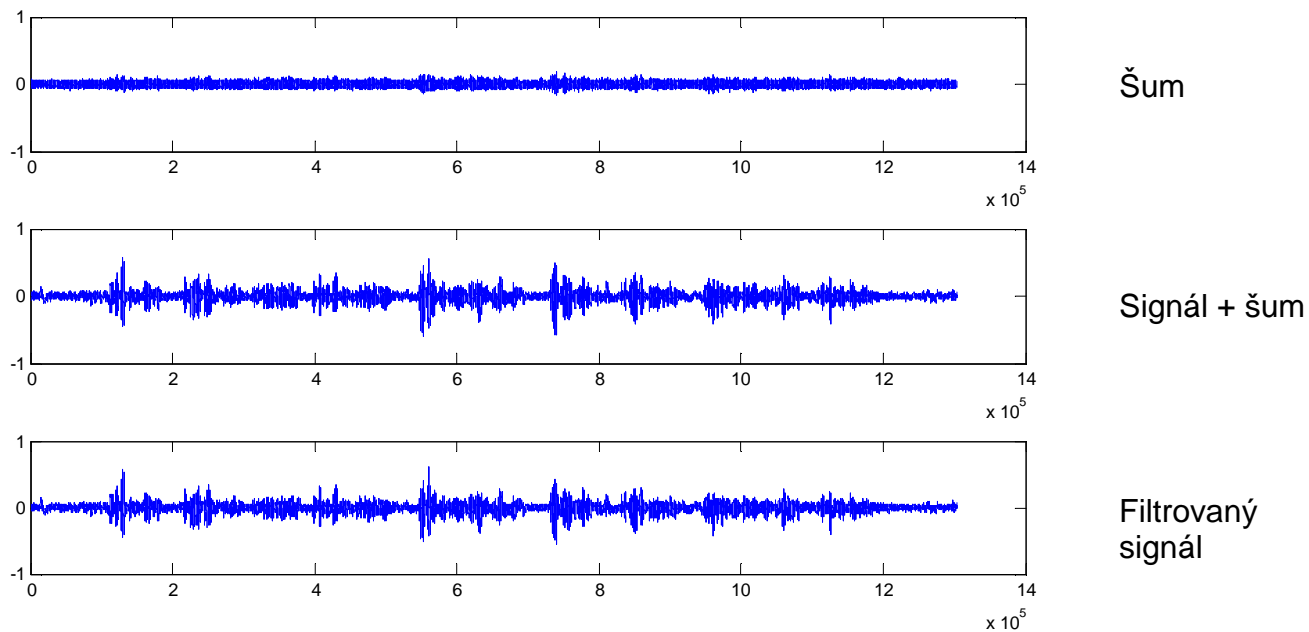
V prvním případě jsme na pořizené nahrávky aplikovali filtr založený na LMS algoritmu s následující konfigurací:

Řád filtru: 60
Krok: 0.002

Na obrázku 4.2 jsou znázorněny průběhy signálů se zvukem vrtačky v pozadí před a po filtraci, na obrázku 4.3 je tentýž signál se zvukem vysavače v pozadí. Je patrné, že po filtraci zůstává signál silně narušený šumem.



Obrázek 4.6: Filtrace signálu se zvukem vrtačky v pozadí pomocí LMS algoritmu

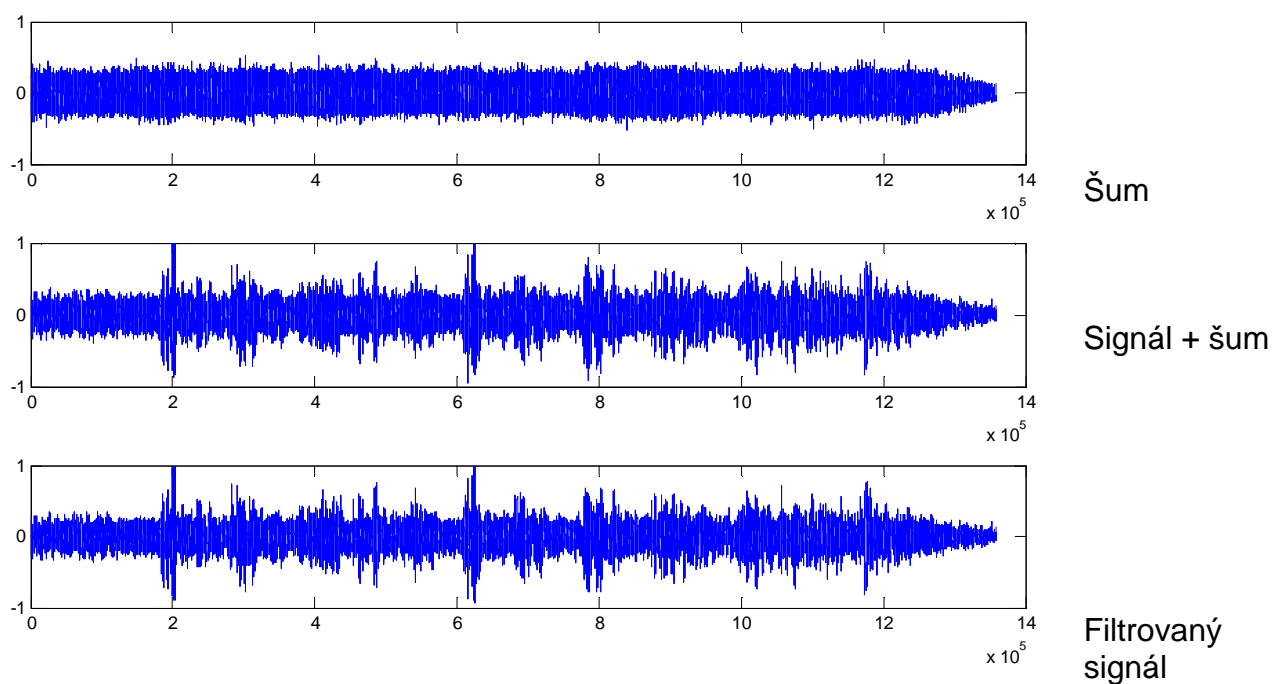


Obrázek 4.7: Filtrace signálu se zvukem vysavače v pozadí LMS algoritmu

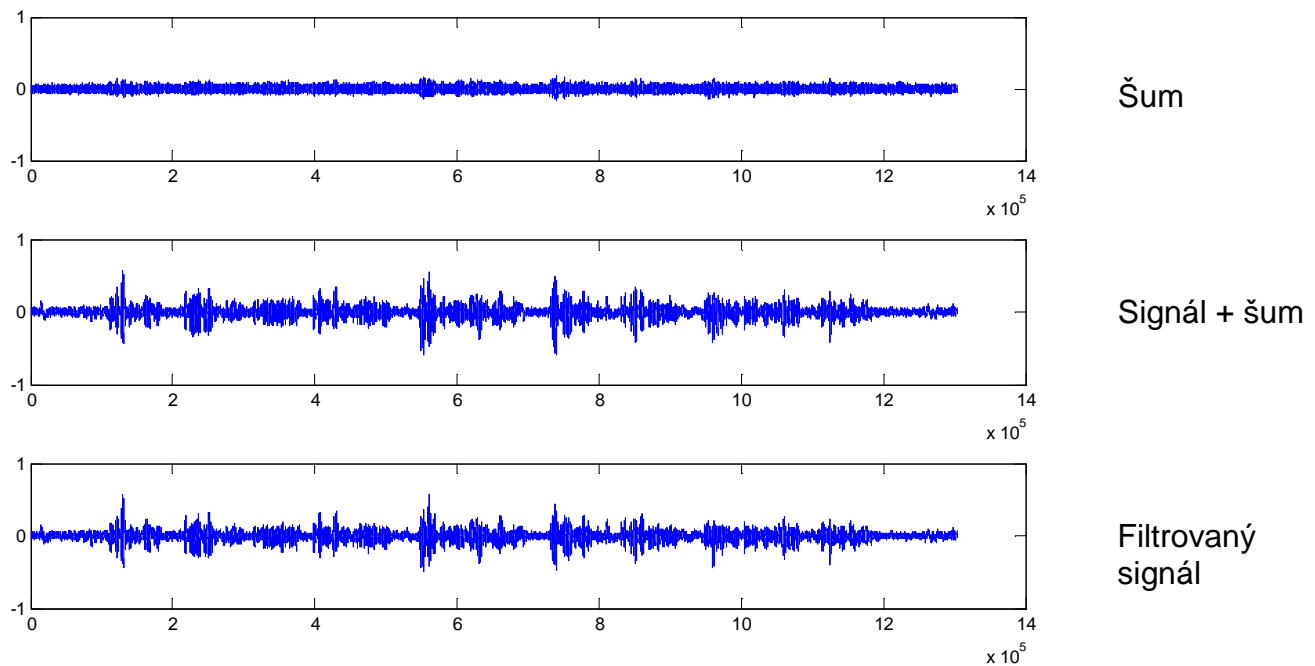
V druhém případě jsme použili filtr založený na NLMS algoritmu o následující konfiguraci:

Řád filtru: 60
Krok: 0.002

Na obrázku 4.4 jsou opět znázorněny průběhy signálů se zvukem vrtačky v pozadí před a po filtraci, na obrázku 4.3 je tentýž signál se zvukem vysavače v pozadí. I v tomto případě je patrné, že filtr nedokázal téměř vůbec odstranit daný šum.

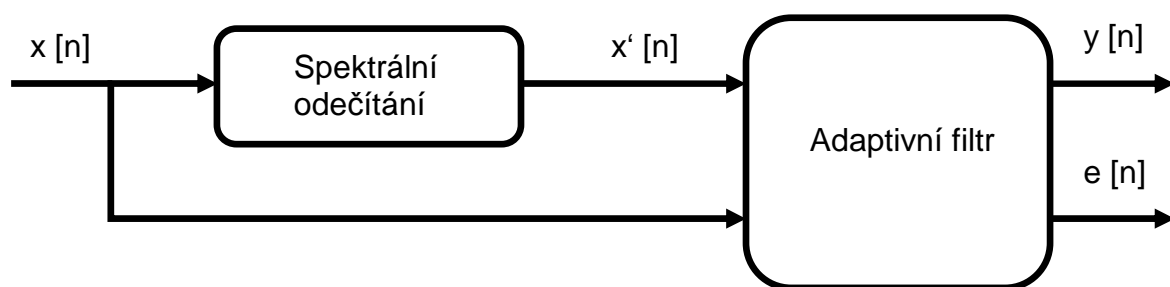


Obrázek 4.8: Filtrace signálu se zvukem vrtačky v pozadí pomocí NLMS algoritmu



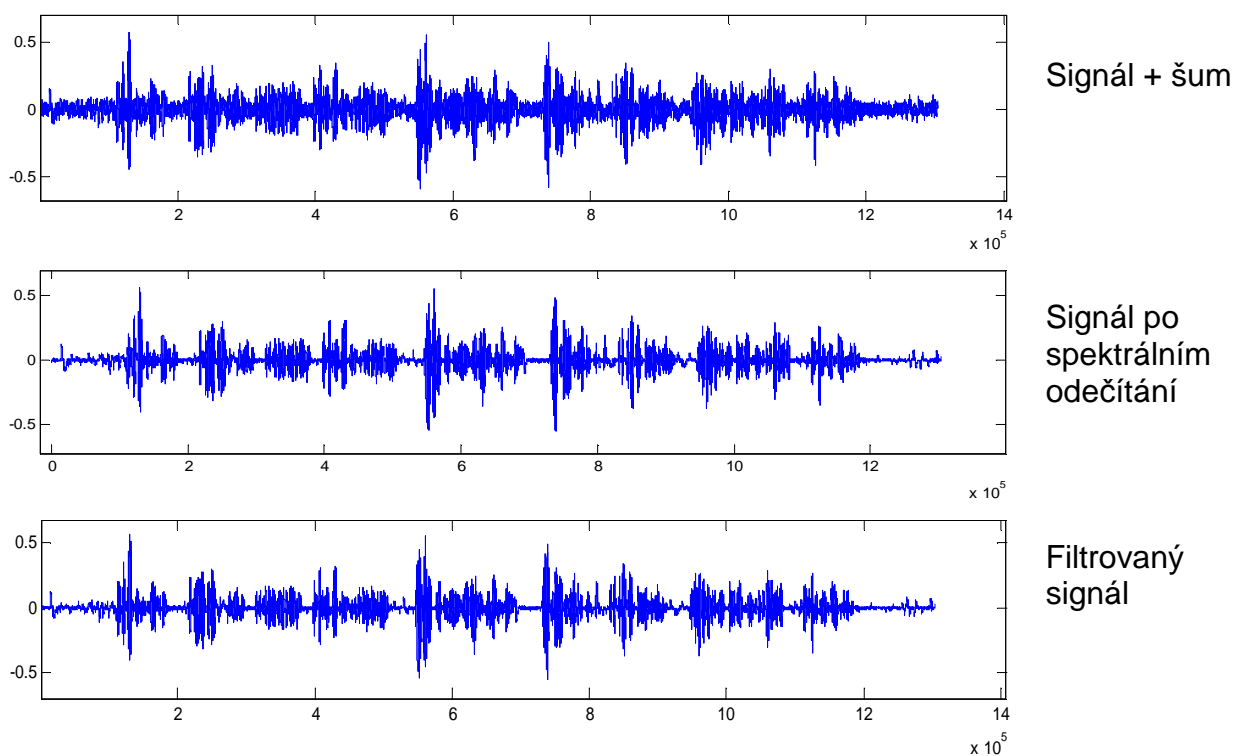
Obrázek 4.9: Filtrace signálu se zvukem vysavače v pozadí pomocí NLMS algoritmu

Jak je vidět, filtr není schopný efektivně odfiltrout šum z nahrávky. Pro jeho ideální funkci by bylo nejlepší, kdyby jako referenční signál byl použit nezkraslený, a šumem neovlivněný signál. Vzhledem k tomu že nejsme schopni v rušivém prostředí tento signál získat, musíme se tomuto signálu alespoň přiblížit. Pro tento účel jsme experimentálně použili algoritmus spektrálního odečítání [3], jehož umístění v systému filtru je na obrázku 4.10. U signálu, který obsahoval řeč znehodnocenou šumem, jsme nejprve provedli spektrální odečítání, abychom se co nejlíže přiblížily ideálnímu, neznehodnocenému řečovému signálu, a spolu s původním znehodnoceným signálem jsme jej přivedli na adaptivní filtr.



Obrázek 4.10: Blokové schéma upraveného systému za použití spektrálního odečítání.

Skript pro spektrální odečítání poskytl Ing. Petr Sysel, Ph.D. Jedná se o skript vytvořený Dr. Grahamem Eatwellem. Pro jeho rozsáhlost zde nebudeme uvádět jeho zdrojový kód. Skript je k dispozici na přiloženém CD.



Obrázek 4.11: Výsledek filtrování za použití spektrálního odečítání

Na obr.4.11 je vidět, že už po spektrálním odečítání je u signálu relativně dobře potlačeno rušení. Po filtraci pomocí adaptivního filtru došlo k nepatrnému zkreslení řeči, ale srozumitelnost zůstala dobrá.

ZÁVĚR

Cílem bakalářské práce bylo seznámit se s adaptivními filtry zaměřenými na adaptivní potlačování šumu v řečovém signálu. V teoretickém úvodu byly uvedeny základní informace o adaptivních filtrech, algoritmech používaných při filtraci a různých metodách potlačování šumu v řečovém signálu.

Simulací těchto metod jsme zjistili, že navržený systém pro potlačení šumu založený pouze na adaptivních filtrech není schopný dostatečně potlačit rušení. Podstatou adaptivního filtru je totiž snaha udržet chybový signál nulový, což znamená, že výstupní signál $y[n]$ musí být takový, aby po odečtení od referenčního signálu $d[n]$ vznikl nulový signál $e[n]$ (viz obr. 1.2). Tedy signál výstupní se přizpůsobuje podle signálu referenčního. Ten by tedy neměl obsahovat řečový signál znehodnocený šumem, ale signál samostatné řeči, který by sloužil jako referenční signál pro adaptivní filtr. Jelikož ale nejsme schopni tento signál získat, protože při pořizování záznamu v hlučném prostředí nejsme schopni při záznamu separovat pouze řeč, musíme tento referenční signál napodobit. Pro tento účel jsme experimentálně vyzkoušeli metodu spektrálního odečítání, která se projevila jako velice efektivní.

Metoda spektrálního odečítání je podle výsledku testování jako efektivnější pro odstranění nežádoucího rušení ale je o dost složitější než metoda adaptivního filtrování.

Literatura

[1] SMÉKAL, Z., SYSEL, P, *Číslicové filtry*, Academia, Skriptum. Brno: FEKT VUT v Brně, 2005.

[2] MALENOVSKÝ, V. *Adaptivní filtrace zašuměných řečových signálů*, 2002, roč. 63, s. 1-6, ISSN 1213-1539
<http://www.elektrorevue.cz/clanky/02063/index.html>

[3] OPRŠAL, P. *Spektrální odečítání ve wavelet oblasti, založené na statistice minim*, 2004, roč. 23, s. 1-11, ISSN 1213-1539
<http://www.elektrorevue.cz/clanky/04023/index.html>

[4] FARHANG-BOROJENY, B., *Adaptive Filters – Theory and Application*, 1998, Wiley, National University of Singapore

Obsah CD:

- | | |
|--|--|
| TEXT | - obsahuje bakalářskou práci |
| - Adaptivní potlačení šumu – Jan Vomáčka.pdf | |
| LMS | - obsahuje skript a zpracované signály pro algoritmus LMS |
| - original.wav | |
| - filtered.wav | |
| - error.wav | |
| - lms.m | |
| NLMS | - obsahuje skript a zpracované signály pro algoritmus NLMS |
| - original.wav | |
| - filtered.wav | |
| - error.wav | |
| - nlms.m | |
| SpecSubt | - obsahuje skript a zpracované signály pro algoritmus LMS a spektrálního odečítání |
| - original.wav | |
| - SpecSub_filtered.wav | |
| - filtered.wav | |
| - error.wav | |
| - SPECSUBT.m | |
| Noises | - obsahuje nahrávky různých hluků a rušení |