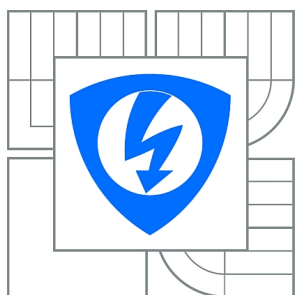


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

POTLAČOVAČ ECHA

ECHO SUPPRESSOR

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

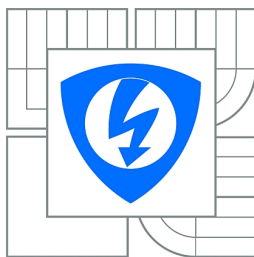
PAVEL KRATOCHVÍL

VEDOUcí PRÁCE

SUPERVISOR

Ing. PETR SYSEL, Ph.D.

BRNO 2012



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Pavel Kratochvíl

ID: 44172

Ročník: 3

Akademický rok: 2011/2012

NÁZEV TÉMATU:

Potlačovač echa

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s číslicovými metodami potlačování echa používanými v telekomunikační technice. Zaměřte se především na metody adaptivní filtrace. V prostředí Matlab implementujte vybranou metodu potlačení echa a otestujte ji na testovacím signálu. Výsledky porovnejte s publikovanými výsledky jiných metod.

DOPORUČENÁ LITERATURA:

- [1] Digital network echo cancellers. Standard ITU-T, 2010.
- [2] Farhang-Boroujeny, B. Adaptive Filters. Theory and Applications. First Edition. New York, Wiley: 1998. ISBN 0-471-98337-3
- [3] Talker echo and its control. Standard ITU-T, 1996.

Termín zadání: 6.2.2012

Termín odevzdání: 31.5.2012

Vedoucí práce: Ing. Petr Sysel, Ph.D.

Konzultanti bakalářské práce:

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 části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

V komunikačních sítích se při spojení vyskytuje echo, což je odražený signál putující zpět sítí a znesnadňující komunikaci. Problém je možné řešit použitím zařízení omezujících odražený signál - potlačovačů echa. Potlačovače by měly splňovat požadavky specifikované v doporučení ITU-T. Algoritmy, které by bylo možné eventuálně pro potlačovače použít je nutné otestovat a poté se rozhodnout o vhodnosti jejich nasazení pro konkrétní použití. V práci jsou otestovány základní algoritmy nejmenší odchylky čtverců LMS, dále její normalizované varianty NLMS a rekurzivní algoritmus RLS. .

KLÍČOVÁ SLOVA

Potlačovač echa, Adaptivní filtr, Wienerův filtr, Algoritmus nejmenší odchylky čtverců, Normalizovaný algoritmus nejmenší odchylky čtverců, Hybridní echo, Ztráty echa, Řeč na blízkém konci.

ABSTRACT

In the communication networks a common problem is a returned copy of an original signal that goes back through the network to the sender, disturbing the communication. This problem is attacked using Echo Cancellors that should comply with the specifications of ITU-T organization. Algorithms suitable for cancellors should be tested under specified conditions, then a decision should be taken regarding their deployment. This work deals with basic Least Mean Squares LMS algorithm, Normalized Least Mean Squares NLMS and Recursive Least Square RLS algorithms.

KEYWORDS

Echo Canceller, Adaptive filter, Wiener Filter, Least Mean Square Algorithm, Normalized Least Mean Square Algorithm, Hybrid Echo, Echo Return Loss, Single Talk.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Potlačovač echa“ 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 bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., a změn některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

(podpis autora)

OBSAH

Úvod	6
1 Potlačovače echa	7
1.1 Historie	7
1.2 Druhy Echa	7
1.3 Typy potlačovačů	9
2 Adaptivní filtrace	10
2.1 Charakteristiky adaptivního filtru	10
2.2 Wienerův filtr	11
2.3 LMS algoritmus	14
2.4 NLMS algoritmus	14
2.5 RLS algoritmus	15
3 Implementace algoritmu do potlačovače echa	20
3.1 Postup při implementaci	20
Implementace NLMS	20
Implementace RLS	21
3.2 Testování algoritmů	22
3.3 Diskuze výsledků	24
4 Závěr	27
Literatura	28
A Soubory na CD	29
B LMS testy	30
C NLMS testy	33
D RLS testy	38

Úvod

Za ozvěnu považujeme odraženou kopii vyslaných dat (hlasu) přijatou po určité době od vyslání. V telefonii může zpoždění na velkých vzdálenostech nabývat značných hodnot (stovky ms) a subjektivně je nepříjemné. Pokud je zpoždění malé (kolem 10 ms), nevadí tolik v telefonii, ale ztěžuje komunikaci modemů [1]. Echo vzniká převážně na straně příjemce hovoru, ale nepříznivě se projevuje na straně volajícího. Protože není možné zabránit vzniku echa, hledají se způsoby pro jeho omezení.

V telefonii se potlačení echa používá za účelem zlepšení kvality hlasového přenosu. Kromě subjektivního zlepšení se však navíc zvyšuje kapacita linky protože ozvěna neputuje zpět sítí.

Potlačení echa spočívá v rozpoznání originálního signálu (který se po nějakém čase vrací) v přijatém signálu a odečtení jeho odrazu (echa) tak, aby se echo nešířilo v síti zpět. Protože prostředí, v němž se echo šíří se může měnit, je nutné aby se v čase měnilo i nastavení filtru, který simuluje vlastnosti prostředí - je nutná adaptace filtru. V praxi se pro adaptaci v reálném čase používají DSP procesory.

Tato práce se nejprve stručně zabývá vznikem echa a jeho důsledky, v další části se pak zaměřuje na způsoby jeho potlačení. V kapitole Adaptivní filtrace jsou odvozeny adaptační algoritmy vycházející z Wienerových filtrů - Least Mean square (LMS), Normalized LMS a RLS (Recursive Least Square) jako jedny ze způsobů potlačení echa a poté následuje kapitola se srovnávacími testy těchto algoritmů, v níž jsem tyto algoritmy (v jejich základní podobě) implementoval na řečové signály v prostředí Matlab a otestoval jejich konvergenci. Výsledky srovnání jsou okomentovány v kapitole 3.

1 POTLAČOVAČE ECHA

1.1 Historie

Dříve se problém ozvěny v telefonii řešil pomocí předpokladu poloduplexní komunikace, tedy jeden člověk mluví a druhý naslouchá, rušiče ozvěny pak detekovaly směr komunikace a zcela zrušily vysílání v opačném směru. Funguje to, ovšem je nemálo případů kdy jsou aktivní oba účastníci, případně jeden odpovídá rychleji než je detektor směru schopný určit a směr přehodit. Proto se začaly používat pokročilejší potlačovače ozvěny (přibližně od roku 1970) pracující s adaptivními filtry. Rychlý rozvoj DSP procesorů pak umožnil implementaci potlačovačů přímo do přepínačů hovorů kde je možné je libovolně přepínat linku pro přenos dat nebo hlasu. Dnešní potlačovače echa pracují i v mobilních telefonech které jsou schopny po připojení do PSTN sítě potlačit echo až do zpoždění 64 ms [3].

1.2 Druhy Echa

- **Akustické echo**

Objevuje se hlavně při hlasitém odposlechu např. při konferenčních hovorech, v automobilu, všude tam kde se hlas z reproduktoru ve větší míře přenesou do mikrofonu. Postup vzniku echa a adaptace na akustické echo je následující:

1. Signál ze vzdáleného konce se přijme
2. Je reprodukován do místnosti
3. Zaznamenán mikrofonem včetně odrazů
4. Signál ze vzdáleného konce je zároveň zfiltrován a časově posunut tak, aby se podobal signálu zaznamenanému mikrofonem
5. Zfiltrovaný a časově posunutý signál se odečte od signálu z mikrofonu
6. Výsledný signál by měl zahrnovat všechny zvuky v místnosti s výjimkou přijatého a reprodukováného signálu.

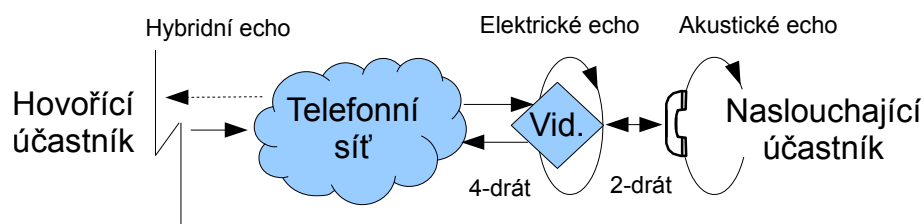
- **Elektrické echo**

Je generováno v PSTN sítích odrazem signálu na vidlici s vyvažovačem sloužící k převodu mezi 2. a 4. drátem (jedné dvousměrné cesty na dvě jednocestné - tedy v místě připojení klasického analogového telefonu - obrázek 1.1). Vyvažovač má v tomto případě odlišnou impedanci. Většina telefonních místní smyček je z důvodu úspory dvoudrátových (v ČR), zatímco k přenosu mezi ústřednami se používá čtyřdrát. Na vyvážení vidlice má vliv délka a vlastnosti připojeného vedení (až 12 km) a impedance telefonního přístroje (300 až 1200 Ω).

Každá vidlice odráží signál v obou směrech, ačkoliv vzdálené echo je pro hlasové přenosy větším problémem. Tento druh echa se vyskytuje poměrně často (15 až 30 %).

Modemy

Výše popsáný způsob odstranění echa může působit problémy modemům; některé zařízení proto zakazují činnost potlačovače echa když detekují 2100 nebo 2225 Hz tóny generované modemy, podle doporučení ITU-T G.164 nebo G.165. V devadesátých letech byla většina potlačovačů echa implementována do modemů typu v.32 a pozdějších. Umožňovalo to použít tytéž frekvence v obou směrech, výrazně tak narostla rychlost přenosu. Při zahájení přenosu každý modem vyslal testovací signál, změřil si echo a mohl tak nastavit zpoždění pomocí parametrů filtru. V tomto případě echo nebylo tvořeno akustickým odrazem, ale jenom impedančním nepřizpůsobením na vidlici. DSL modemy používají potlačovače echa také ve velké míře, testování probíhá v širokém frekvenčním pásmu kvůli častým poruchám a rušení na telefonních linkách které nebyly původně určeny k rychlým přenosům dat.

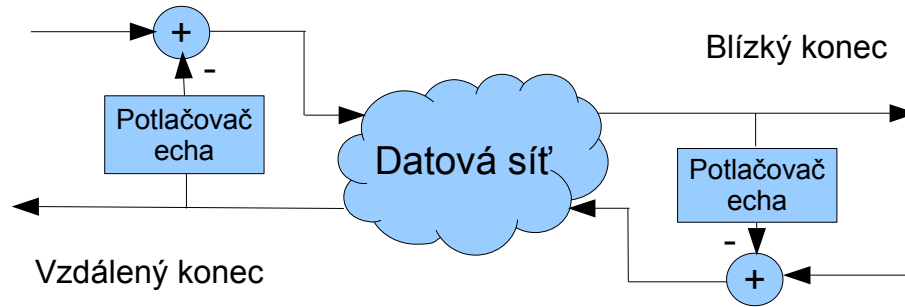


Obr. 1.1: Místa na vedení, kde vzniká elektrické a akustické echo.

1.3 Typy potlačovačů

Potlačovače bývají umístěny ve čtyřdrátové části vedení, pro jeden hovor (výměnu dat) jsou potlačovače dva, každý na jedné straně vedení, co nejbližže zdroji echa.

- Linkový potlačovač - pro odstranění hybridního echa
- Paketový potlačovač - pro odstranění jak hybridního tak i akustického echa



Obr. 1.2: Umístění potlačovačů echa na vedení.

2 ADAPTIVNÍ FILTRACE

2.1 Charakteristiky adaptivního filtru

Pro zpracování signálu jehož charakteristiky se s časem nemění je dostačující návrh fixního filtru, nicméně v praxi existuje mnoho aplikací, kdy okolí ve kterém systém pracuje, se mění s časem. V některých aplikacích (aktivní potlačení šumu, echa..) je spíše pravidlem, že celý systém je při návrhu neznámý. V takových případech je potřeba navrhnout filtr, který sám upraví své charakteristiky podle měnícího se okolí [4].

Adaptivní filtr je filtr, který během filtrace mění přenosovou funkci, oproti neadaptivnímu filtru s konstantními koeficienty se tento filtr aktivně přizpůsobuje proměnnému požadovanému signálu. Cílem přizpůsobení je co nejlepší odhad parametrů filtru tak, aby se signál z filtru co nejvíce podobal signálu požadovanému.

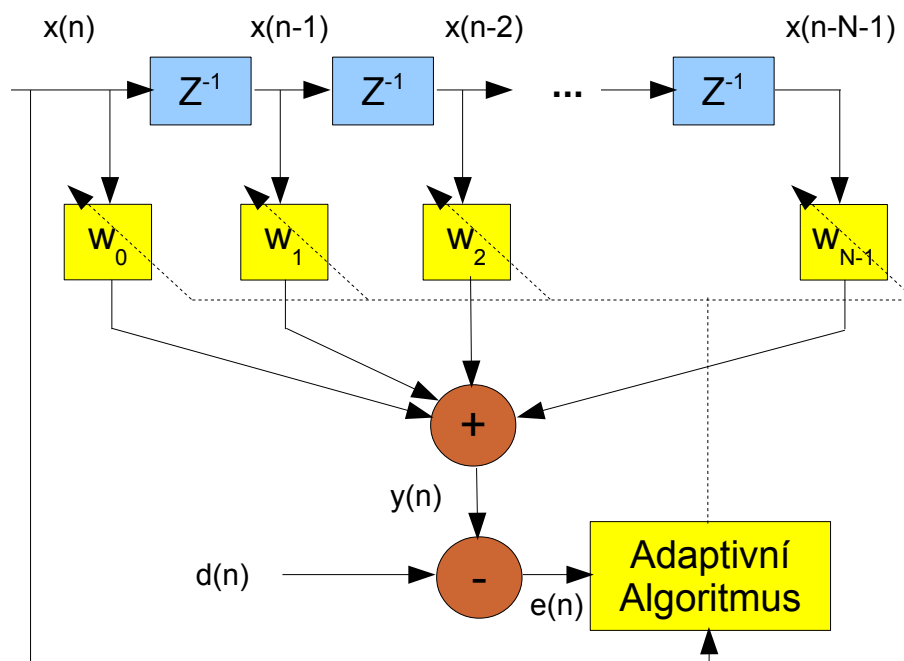
Proces nalezení parametrů filtru se řeší optimalizací chybové funkce, která může obecně být podmíněna deterministicky nebo stochasticky (statisticky). Statistický přístup vyžaduje velké množství realizací (měření) ke spočtení statistiky (což může být problém), tento přístup vede např. k LMS (Least Mean Squares) algoritmu a jeho klonům. Při deterministickém přístupu návrh filtru vyžaduje spočtení charakteristik z většího počtu vzorků, vede k RLS algoritmu (Recursive Least Squares) a jeho odvozeninám.

Většina algoritmů používá dále opakování (iterací), kdy se vezme výstupní chyba filtru, zkoreluje se nějakým způsobem se vstupem a výstup se použije ke zpřesnění koeficientů filtru. Důvodem pro to je úspora paměti, zavedení menšího zpoždění výstupu a jednodušší implementace.

Nejčastější způsob implementace adaptivního filtru je zobrazen na obrázku 2.1. Adaptivní filtr zde má jeden vstup $x(n)$ a jeden výstup $y(n)$, požadovaný signál (desired) je značený $d(n)$. Filtr lze rozdělit do dvou částí- část filtrační (spočte výstup $y(n)$ a část adaptační (nastavuje N koeficientů filtru) w_i . Výstup $y(n)$ se spočte jako lineární kombinace vstupní sekvence $x(n-i)$, $i=0,1,\dots,N-1$, tedy zpožděných vstupních vzorků.

$$y(n) = \sum_0^{N-1} w_i(n) \cdot x(n-i) \quad (2.1)$$

Váhy w_i které se mohou v čase měnit jsou kontrolovány adaptačním algoritmem.

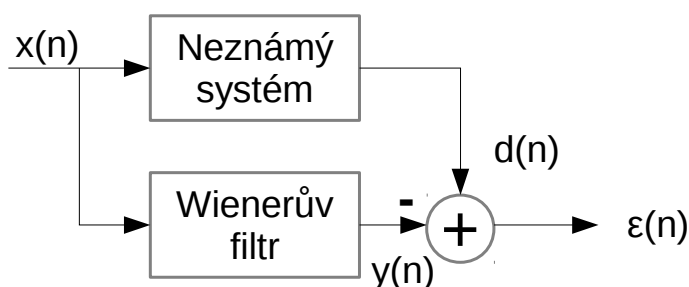


Obr. 2.1: Struktura adaptivního transversálního filtru

2.2 Wienerův filtr

Řada adaptačních algoritmů je založena na Wienerově filtru, nejdříve bude proto uveden tento koncept, v další části pak na něj naváže LMS algoritmus.

Na obrázku 2.2 je blokové schéma lineárního diskrétního filtru který se snaží výstupní funkci $y(n)$ přizpůsobit požadovanému signálu $d(n)$, který je funkcí vstupního signálu $x(n)$. Odečtením $y(n)$ a $d(n)$ dostaneme chybový signál $e(n)$ který určuje úspěšnost odhadu filtru (pro malé hodnoty filtr pracuje dobře).



Obr. 2.2: Blokové schéma problému filtrace.

Chyba $e(n)$ podmiňuje výběr koeficientů w_i a funkce, která tento výběr provádí by měla splňovat požadavek na jednoduše vystopovatelné minimum (nebo maximum) aby nedocházelo k nejednoznačnostem při konvergenci filtru [6].

Tyto podmínky splňují FIR filtry s kritériem chyby

$$\xi = E[|e(n)|^2], \quad (2.2)$$

kde $E[.]$ vyjadřuje střední očekávanou hodnotu chyby. Tato funkce je nejjednodušší, má jednoduché minimum (hyperparaboloid) a jde snadno spočítat.

Označíme vektor koeficientů filtru jako:

$$\mathbf{w} = [w_0 \ w_1 \ \dots \ w_{N-1}]^T, \quad (2.3)$$

a vektor vstupních hodnot jako:

$$\mathbf{x}(n) = [x(n) \ x(n-1) \ \dots \ x(N-n+1)]^T, \quad (2.4)$$

pak výstup filtru lze zapsat jako:

$$y(n) = \sum_{i=0}^{N-1} w_i x(n-i) = \mathbf{x}^T(n) \cdot \mathbf{w} = \mathbf{w}^T \cdot \mathbf{x}(n). \quad (2.5)$$

Chybový vektor je pak:

$$e(n) = d(n) - y(n) = d(n) - \mathbf{x}^T(n) \cdot \mathbf{w} = d(n) - \mathbf{w}^T \cdot \mathbf{x}(n). \quad (2.6)$$

Pro uvedené kritérium chyby ξ platí:

$$\begin{aligned} \xi &= E[|e^2(n)|] \\ &= E[(d(n) - y(n))^2] \\ &= E[(d(n) - \mathbf{x}^T(n) \cdot \mathbf{w}) \cdot (d(n) - \mathbf{w}^T \cdot \mathbf{x}(n))] \\ &= E[d^2(n) - d(n) \cdot (\mathbf{x}^T(n) \cdot \mathbf{w}) + \mathbf{w}^T \cdot \mathbf{x}(n) + \mathbf{x}^T(n) \cdot \mathbf{w}) \cdot \mathbf{w}^T \cdot \mathbf{x}(n)] \\ &= E[d^2(n)] - E[d(n)\mathbf{x}^T(n) \cdot \mathbf{w}] - E[d(n)\mathbf{w}^T \cdot \mathbf{x}(n)] + E[\mathbf{x}^T(n) \cdot \mathbf{w}) \cdot \mathbf{w}^T \cdot \mathbf{x}(n)] \\ &= E[d^2(n)] - E[d(n)\mathbf{x}^T(n)] \cdot \mathbf{w} - \mathbf{w}^T \cdot E[d(n)\mathbf{x}(n)] + \mathbf{w}E[\mathbf{x}^T(n) \cdot \mathbf{x}(n)]\mathbf{w}^T \quad (2.7) \end{aligned}$$

V předchozím vztahu se mohl vytknout vektor koeficientů w_i z očekávané střední hodnoty E poněvadž w není statistickou veličinou. Ve vztahu (2.7) se pro další úpravy označí vektor p :

$$\mathbf{p} = E[d(n)\mathbf{x}^T(n)] \quad (2.8)$$

Vektor \mathbf{p} má význam kroskorelační funkce a má rozměr $N - 1$ prvků. Dále se pak ve vztahu (2.7) označuje výraz $E[\mathbf{x}^T(n) \cdot \mathbf{x}(n)]$ jako funkce \mathbf{R} , je to matice o rozměrech $(N - 1) \times (N - 1)$ a má význam autokorelační funkce:

$$\mathbf{R} = \begin{bmatrix} r_{00} & r_{01} & r_{02} & \dots & r_{0,N-1} \\ r_{10} & r_{11} & r_{12} & \dots & r_{1,N-1} \\ r_{20} & r_{21} & r_{22} & \dots & r_{2,N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{N-1,0} & r_{N-1,1} & r_{N-1,2} & \dots & r_{N-1,N-1} \end{bmatrix}. \quad (2.9)$$

Po dosazení do rovnice (2.7) pak dostaneme:

$$\xi = E[d^2(n)] - \mathbf{p} \cdot \mathbf{w} - \mathbf{w}^T \cdot \mathbf{p}^T + w \cdot \mathbf{R} \cdot \mathbf{w}^T = E[d^2(n)] - 2\mathbf{p} \cdot \mathbf{w} + \mathbf{w} \cdot \mathbf{R} \cdot \mathbf{w}^T. \quad (2.10)$$

Funkce ξ je kvadratickou funkcí \mathbf{w} a má jedno globální minimum. Polohu minima zjistíme pomocí parciálních derivací funkce ξ podle koeficientů \mathbf{w} a položením zderivovaných funkcí =0:

$$\nabla \xi = \mathbf{0}, \quad (2.11)$$

kde ∇ je operátor gradientu:

$$\nabla \xi = \left[\frac{\partial}{\partial w_0} + \frac{\partial}{\partial w_1} + \dots + \frac{\partial}{\partial w_{N-1}} \right]^T. \quad (2.12)$$

S využitím vztahu platného pro symetrické matice:

$$\frac{d}{d\mathbf{w}}(\mathbf{A}\mathbf{w}) = \mathbf{A}, \quad \frac{d}{d\mathbf{w}}(\mathbf{w}M\mathbf{w}^T) = 2M\mathbf{w}, \quad (2.13)$$

je hledaný vztah:

$$\nabla \xi = -2\mathbf{R}\mathbf{w} - 2\mathbf{p} = \mathbf{0}, \quad (2.14)$$

odkud:

$$\mathbf{w}_{\text{opt}}\mathbf{R} = \mathbf{p} \Rightarrow \mathbf{w}_{\text{opt}} = \mathbf{R}^{-1}\mathbf{p}. \quad (2.15)$$

Rovnice (2.15) tedy umožňuje spočítat optimální váhy filtru a je známá pod jménem *Wiener-Hopf* rovnice.

2.3 LMS algoritmus

LMS algoritmus je nejpoužívanější algoritmus pro adaptivní filtraci, široké uplatnění si našel především kvůli jednoduchosti a robustnosti. Byl použit i v diplomové práci [3], a navazuje na něj i modifikace NLMS (Normalized) s výsledky uvedenými dále. LMS algoritmus je implementací algoritmu pro největší spád - gradientu chyby, pomocí kterého zajistíme pohyb ke globálnímu minimu v prostoru koeficientů \mathbf{w} . Algoritmus vezme předchozí hodnotu $\mathbf{w}(n)$ a připočte k ní novou hodnotu zjištěnou z \mathbf{x} a ze spočteného gradientu:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \nabla e^2(n), \quad (2.16)$$

kde μ je konvergenční konstanta, která určuje velikost kroku při pohybu směrem k minimu a tedy rychlost konvergence. Na druhé straně se však pro větší krok snižuje přesnost konvergence.

Po provedení naznačené derivace a rozepsání dostáváme:

$$w_k(n+1) = w_k(n) + 2\mu e(n)x(n-k), \quad (2.17)$$

kde k je délka filtru.

Velikost konvergenční konstanty je nastavena na počátku výpočtu a měla by se pohybovat v mezích:

$$\frac{0,01}{KE[y^2(i)]} < \mu < \frac{0,1}{KE[y^2(i)]}, \quad (2.18)$$

kde K je délka FIR filtru,
 $y^2(i)$ je energie signálu $y(i)$.

2.4 NLMS algoritmus

V normalizovaném LMS je krok pro gradient μ normalizován energií vektoru \mathbf{x} , tedy

$$\mu_{NLMS} = \frac{\alpha}{x^T(n)x(n) + \sigma}. \quad (2.19)$$

Znamená to, že při velkých hodnotách vstupu $x(n)$ se bude postupovat k minimu pomaleji a naopak. Koeficient α je nutné otestovat pro konkrétní aplikaci, pro testy 2A potlačovače echa byla zvolena hodnota od 0,03 do 1,4. Hodnota σ je nějaké malé číslo, které zajistí aby se ve vztahu (2.19) nedělilo nulou (po několika testech jsem zvolil hodnotu $\sigma = 10^{-14}$).

2.5 RLS algoritmus

Jak již bylo uvedeno v kapitole 2.1, je rekurzivní algoritmus nejmenších čtverců (RLS) zástupcem deterministického přístupu k výpočtu koeficientů adaptivního filtru.

Pro časově neměnné signály RLS konverguje k těm samým koeficientům jako Wienerův filtr, pro signály časově proměnné RLS sleduje časové variace signálu. RLS má poměrně rychlou konvergenci k optimálním parametrům filtru, což je v potlačovačích echa důležité, na druhou stranu je však v jeho základní podobě výpočetně hodně náročný. RLS startuje s nějak nastavenými koeficienty filtru (ve vytvořených skriptech potlačovače echa jsou vynulované) a algoritmus se následně snaží tyto koeficienty adaptovat. Následující odvození je podle [4], avšak doplněno komentáři, které souvisí s potlačovačem echa.

V úvodu 2.1 byl uveden vztah pro výstup adaptivního filtru:

$$y(n) = \sum_0^{N-1} w_i(n) \cdot x(n-i). \quad (2.20)$$

Rozdíl oproti normální filtraci s koeficienty určenými z řešení rovnice (2.15) je v tom, že váhy w_i se mění v čase, určujeme je v daném časovém okamžiku n .

Chybový vektor je pak určený jako:

$$e_n(k) = d(k) - y_n(k), \quad (2.21)$$

Index k určuje pořadové číslo výpočtu, $k=1$ v čase startu algoritmu a maximální hodnota k je rovna n . Hledáme tedy koeficienty filtru $\mathbf{w}(n)$ v každém časovém okamžiku n . Pro jednodušší zápis problému se opět volí zápis formou matic a vektorů, požadovaný signál značíme jako:

$$\mathbf{d}(n) = [d(1) \ d(2) \ d(3) \ \dots \ d(n)]^T, \quad (2.22)$$

výstupní hodnoty z filtru jako:

$$\mathbf{y}(n) = [y_n(1) \ y_n(2) \ y_n(3) \ \dots \ y_n(n)]^T, \quad (2.23)$$

a chybový vektor jako:

$$\mathbf{e}(n) = [e_n(1) \ e_n(2) \ e_n(3) \ \dots \ e_n(n)]^T. \quad (2.24)$$

Matici vstupních vzorků si označíme jako:

$$\mathbf{X}(n) = [\mathbf{x}(1) \ \mathbf{x}(2) \ \mathbf{x}(3) \ \dots \ \mathbf{x}(n)], \quad (2.25)$$

tedy v případě nějaké časové posloupnosti vzorků budou jednotlivé sloupce obsahovat vždy o jeden interval vzájemně posunuté vzorky.

Výstup z filtru je možné zapsat maticově jako:

$$\mathbf{y}(n) = \mathbf{X}(n)^T \mathbf{w}(n). \quad (2.26)$$

Podobně chybový vektor:

$$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{y}(n) \quad (2.27)$$

V následujícím kroku je nadefinována chybová funkce, která má tvar daný jednoduchostí následujícího odvození, je sice možné definovat chybovou funkci např. jako sumu absolutních hodnot jednotlivých hodnot $\mathbf{e}(n)$, ale následující funkce má výhodu jednoduše lokalizovatelného minima:

$$\xi(n) = \mathbf{e}^T(n) \mathbf{e}(n) \quad (2.28)$$

Dosazením (2.26) a (2.27) do (2.28) dostaneme:

$$\xi(n) = \mathbf{d}^T(n) \mathbf{d}(n) - 2\mathbf{X}^T(n) \mathbf{d}(n) \mathbf{w}(n) + \mathbf{w}^T(n) \mathbf{X}(n) \mathbf{X}^T(n) \mathbf{w}(n) \quad (2.29)$$

a pak označíme:

$$\mathbf{\Psi}(n) = \mathbf{X}(n) \mathbf{X}^T(n), \quad (2.30)$$

$$\boldsymbol{\theta}(n) = \mathbf{X}(n) \mathbf{d}(n), \quad (2.31)$$

tedy:

$$\xi(n) = \mathbf{d}^T(n) \mathbf{d}(n) - 2\boldsymbol{\theta}^T(n) \mathbf{w}(n) + \mathbf{w}^T(n) \mathbf{\Psi}(n) \mathbf{w}(n). \quad (2.32)$$

Zafixováním gradientu chybové funkce $\xi(n)$ na nulu a podobným odvozením jako pro Wienerův filtr dostaneme vztah pro koeficienty filtru ve smyslu nejmenších čtverců:

$$\hat{\mathbf{w}}(n) = \mathbf{\Psi}^{-1}(n) \boldsymbol{\theta}(n). \quad (2.33)$$

Po dosazení do vztahu pro výpočet chyby (2.32) dostaneme:

$$\xi(n) = \mathbf{d}^T(n) \mathbf{d}(n) - \boldsymbol{\theta}^T(n) \mathbf{\Psi}^{-1}(n) \boldsymbol{\theta}(n) = \mathbf{d}^T(n) \mathbf{d}(n) - \boldsymbol{\theta}(n) \hat{\mathbf{w}}(n). \quad (2.34)$$

Tato rovnice v takovéto podobě nemá praktický význam, protože pro výpočet koeficientů filtru vyžaduje všechny vzorky od začátku adaptace a tím pádem také narůstá výpočetní náročnost. Proto se v případě metody RLS zavádí tzv. zapomínací faktor definovaný jako:

$$\rho_n(k) = \lambda^{n-k}, k = 1, 2, \dots, n, \quad (2.35)$$

tedy pro $\lambda = \langle 0, 1 \rangle$ bude zapomínací faktor vážit předchozí vzorky tím méně, čím jsou starší (staré vzorky se blíží nule). Hodnota λ se obvykle volí blízko jedné, pro $\lambda = 1$ se bude počítat se všemi vzorky (nekonečná paměť).

Zapomínací faktor implementujeme pomocí diagonální matice:

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda^{n-1} & 0 & 0 & \dots & 0 \\ 0 & \lambda^{n-2} & 0 & \dots & 0 \\ 0 & 0 & \lambda^{n-3} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}, \quad (2.36)$$

kterou vložíme do chybové funkce:

$$\xi(n) = \mathbf{e}^T(n)\mathbf{\Lambda}(n)\mathbf{e}(n), \quad (2.37)$$

a stejným postupem jako při odvození vztahu (2.33) dostaneme vztah pro nové koeficienty filtru:

$$\hat{\mathbf{w}}(n) = \mathbf{\Psi}_\lambda^{-1}(n)\boldsymbol{\theta}_\lambda(n), \quad (2.38)$$

kde

$$\mathbf{\Psi}(n) = \mathbf{X}(n)\mathbf{\Lambda}(n)\mathbf{X}^T(n), \quad (2.39)$$

a

$$\boldsymbol{\theta}(n) = \mathbf{X}(n)\mathbf{\Lambda}(n)\mathbf{d}(n), \quad (2.40)$$

Pro spočtení koeficientů $\hat{\mathbf{w}}$ se musí spočítat inverzní matice $\mathbf{\Psi}^{-1}$.

Rovnici (2.39) můžeme rozepsat takto:

$$\mathbf{\Psi}_\lambda(n) = \mathbf{x}(n)\mathbf{x}^T(n) + \lambda\mathbf{x}(n-1)\mathbf{x}^T(n-1) + \lambda\mathbf{x}(n-2)\mathbf{x}^T(n-2) + \dots \quad (2.41)$$

a

$$\boldsymbol{\theta}_\lambda(n) = \mathbf{x}(n)d(n) + \lambda\mathbf{x}(n-1)d(n-1) + \lambda^2\mathbf{x}(n-2)d(n-2) + \dots \quad (2.42)$$

Z předchozích dvou vztahů je vidět, že je možné je zapsat také rekurzivně jako:

$$\Psi_\lambda(n) = \mathbf{x}(n)\mathbf{x}^T(n) + \lambda\Psi_\lambda(n-1) \quad (2.43)$$

a

$$\boldsymbol{\theta}_\lambda(n) = \mathbf{x}(n)d(n) + \lambda\boldsymbol{\theta}_\lambda(n-1), \quad (2.44)$$

což znamená to, že pro získání nové matice (nebo vektoru) vezmeme starou a připočteme její "update."

V dalším odvození se používá tzv. Maticové inverzní lemma, podle nějž pro pozitivně definitní matice \mathbf{A} , \mathbf{B} rozměrů $N \times N$ a \mathbf{C} rozměru $P \times P$ platí:

$$\mathbf{A} = \mathbf{B}^{-1} + \mathbf{C}\mathbf{D}^{-1}\mathbf{C}^T \quad (2.45)$$

$$\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B}\mathbf{C}(\mathbf{D} + \mathbf{C}^T\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^T\mathbf{B}, \quad (2.46)$$

což lze dokázat vzájemným vynásobením $\mathbf{A}\mathbf{A}^{-1} = \mathbf{E}$, kde \mathbf{E} je jednotková matice.

Pokud provedeme záměnu:

$$\Psi_\lambda(n) = \mathbf{A}, \quad (2.47)$$

$$\lambda\Psi_\lambda(n-1) = \mathbf{B}^{-1}, \quad (2.48)$$

$$\mathbf{x}(n) = \mathbf{C}, \quad (2.49)$$

a \mathbf{D} je jednotková matice, pak:

$$\Psi_\lambda^{-1}(n) = \lambda^{-1}\Psi_\lambda^{-1}(n-1) - \frac{\lambda^{-2}\Psi^{-1}(n-1)\mathbf{x}(n)\mathbf{x}^T(n)\Psi_\lambda^{-1}(n-1)}{1 + \lambda^{-1}\mathbf{x}^T(n)\Psi^{-1}(n-1)\mathbf{x}(n)}. \quad (2.50)$$

Pro zjednodušení se část zlomku označí jako sloupcový vektor:

$$\mathbf{k}(n) = \frac{\lambda^{-1}\Psi^{-1}(n-1)\mathbf{x}(n)}{1 + \lambda^{-1}\mathbf{x}^T(n)\Psi^{-1}(n-1)\mathbf{x}(n)}, \quad (2.51)$$

takže dostaneme:

$$\Psi_\lambda^{-1}(n) = \lambda^{-1} \left(\Psi_\lambda^{-1}(n-1) - \mathbf{k}(n)\mathbf{x}^T(n)\Psi_\lambda^{-1}(n-1) \right) \quad (2.52)$$

Rovnice (2.50) se zjednoduší:

$$\mathbf{k}(n) \left(1 + \lambda^{-1}\mathbf{x}^T(n)\Psi^{-1}(n-1)\mathbf{x}(n) \right) = \lambda^{-1}\Psi^{-1}(n-1)\mathbf{x}(n), \quad (2.53)$$

$$\mathbf{k}(n) = \Psi^{-1}(n)\mathbf{x}(n), \quad (2.54)$$

kde na vektor $\mathbf{k}(n)$ můžeme nahlížet jako na vektor zisku, počítá se tedy ze zpožděných vzorků $(n - 1)$.

Nyní můžeme dosadit do rovnice (2.38) pro odhad koeficientů filtru:

$$\hat{\mathbf{w}}(n) = \lambda \Psi_{\lambda}^{-1}(n)\boldsymbol{\theta}_{\lambda}(n-1) + \Psi_{\lambda}^{-1}(n)\mathbf{x}(n)d(n) = \lambda \Psi_{\lambda}^{-1}(n)\boldsymbol{\theta}_{\lambda}(n-1) + \mathbf{k}(n)d(n) \quad (2.55)$$

Funkci Ψ_{λ}^{-1} ve vztahu (2.55) nahradíme funkcí (3.1) tak, abychom pracovali rekurzivně, se vzorkem posunutým dozadu o $(n - 1)$.

$$\begin{aligned} \hat{\mathbf{w}}(n) &= \lambda \Psi_{\lambda}^{-1}(n-1)\boldsymbol{\theta}_{\lambda}(n-1) - \mathbf{k}(n)\mathbf{x}^T(n)\Psi_{\lambda}^{-1}(n-1)\boldsymbol{\theta}_{\lambda}(n-1) + \mathbf{k}(n)d(n) \\ &= \hat{\mathbf{w}}(n-1) - \mathbf{k}(n)\mathbf{x}^T(n)\hat{\mathbf{w}}(n-1) + \mathbf{k}(n)d(n) \\ &= \hat{\mathbf{w}}(n-1) + \mathbf{k}(n) \left(d(n) - \hat{\mathbf{w}}^T(n-1)\mathbf{x}(n) \right) \end{aligned} \quad (2.56)$$

Výraz v závorce připomíná chybu danou rozdílem mezi požadovaným signálem a signálem z filtru, rovnice se tedy dá zapsat také jako:

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)\hat{e}_{n-1}(n), \quad (2.57)$$

což je rekurzivní rovnice používaná RLS algoritmem pro výpočet nových koeficientů $(w)(n)$ Míra změny je určena vektorem $\mathbf{k}(n)$ a odhadem chyby \hat{e}_{n-1} .

3 IMPLEMENTACE ALGORITMU DO POTLAČOVAČE ECHA

3.1 Postup při implementaci

Nejdříve je potřeba upřesnit pojmy které vystupují v potlačovači echa a při výpočtu algoritmu. Vstupním signálem $\mathbf{x}(n)$ je signál ze vzdáleného konce, tj. signál ke kterému budeme vztahovat výpočet koeficientů filtru. V potlačovači se značí *Rin*, je to hovor vzdáleného mluvčího. Požadovaným signálem (desired signal $d(n)$) je signál z blízkého konce, který už obsahuje charakteristiku prostředí, kterým prošel, je to tedy signál $\mathbf{x}(n)$ zkonvolvovaný s impulsní odezvou daného prostředí, v potlačovači se značí *Sin*) a v neadaptivní fázi filtrace může obsahovat signál mluvčího z blízkého konce. Chybový signál $\hat{e}(n)$ vznikne odečtem požadovaného signálu od signálu odhadnutého filtrem s koeficienty $\hat{\mathbf{w}}(n)$ a v potlačovači echa je to výstup který posíláme ke vzdálenému konci, značí se *Sout* a v ideálně konvergovaném fitru je nulový (pokud není na blízkém konci řečový signál). V následujícím textu uvedu dvě implementace, které jsem provedl zavedením matlabovských skriptů (viz příloha B a C), implementace LMS algoritmu již byla provedena v práci [3]

Implementace NLMS

V práci jsem zvolil tento postup:

1. Vybereme část vstupního vektoru, délka vybraného bloku je určena délkou filtru, kterou nastavíme podle požadované adaptační úlohy (v potlačovači se jako dostatečná ukázala délka 160 vzorků), tento blok zkonvolvujeme s doposud neadaptovanými (zde nulovými) koeficienty filtru.

$$y(n) = \sum_{i=0}^{N-1} w_i x(n-i) = \mathbf{x}^T(n) \cdot \mathbf{w}$$

2. Spočteme odchylku požadovaného a zkonvolvovaného signálu:

$$e(n) = d(n) - y(n) = d(n) - \mathbf{x}^T(n) \cdot \mathbf{w}.$$

3. Provedeme normalizaci:

$$\mu_{NLMS} = \frac{\alpha}{\mathbf{x}^T(n)\mathbf{x}(n) + \sigma}.$$

4. Vypočteme nové koeficienty filtru:

$$w_k(n+1) = w_k(n) + 2\mu e(n)x(n-k).$$

5. Postup opakujeme pro další vstupní blok, ten je tvořen předchozím blokem s jedním novým vzorkem a jedním vzorkem který na konci vypadne.

Implementace RLS

Při vlastním výpočtu RLS byl zvolen tento způsob:

1. Vypočteme vektoru zisku:

$$\mathbf{k}(n) = \frac{\lambda^{-1} \mathbf{\Psi}^{-1}(n-1) \mathbf{x}(n)}{1 + \lambda^{-1} \mathbf{x}^T(n) \mathbf{\Psi}^{-1}(n-1) \mathbf{x}(n)},$$

Jako iniciální hodnotu $\mathbf{\Psi}^{-1}$ zvolíme diagonální matici s hodnotou δ^{-1} na diagonále. Hodnotu λ nastavíme např. na 0,5.

2. Určíme chybový vektor:

$$\hat{e}_{n-1} = d(n) + \hat{\mathbf{w}}^T(n-1) \mathbf{x}(n).$$

Jako iniciální hodnotu filtru použijeme $\hat{\mathbf{w}}^T = [0, 0, \dots, 0]^T$.

3. Spočteme nové koeficienty filtru:

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{k}(n) \hat{e}_{n-1}(n),$$

4. Obnovíme hodnotu matice $\mathbf{\Psi}^{-1}$:

$$\mathbf{\Psi}_{\lambda}^{-1}(n) = \lambda^{-1} \left(\mathbf{\Psi}_{\lambda}^{-1}(n-1) - \mathbf{k}(n) \mathbf{x}^T(n) \mathbf{\Psi}_{\lambda}^{-1}(n-1) \right)$$

5. Cyklus 1-4 opakujeme až do dosažení konce datového vektoru $\mathbf{x}(n)$

3.2 Testování algoritmů

Charakteristiky, které by měly potlačovače echa nasazené do provozu splňovat jsou popsány v doporučení ITU-T G.168. Doporučení taktéž popisuje sérii laboratorních testů, které by měly být na potlačovači provedeny, aby byla zajištěna jeho funkce v podmínkách, které se v síti budou pravděpodobně vyskytovat. Pro otestování použitelnosti algoritmů pro potlačovač jsem vybral test č. 2A což je test konvergence, rekonvergence a úrovně vráceného echa. Test je zaměřen na rychlost konvergence pro náhlé změny (případně změny kombinací) charakteristiky prostředí, ve kterém echo vzniká (nezahrnuje tedy kontinuálně se měnící prostředí). Vrácená reziduální úroveň echa by měla být nízká tak, aby bylo možné použít následně nelineární zpracování (omezení nízkých hodnot).

Test 2A vychází z počátečního předpokladu resetovaného registru H , tedy vynulovaných koeficientů \mathbf{w} , povoleného nelineárního zpracování (dále v textu bude značeno zkratkou NLP), rozsahu vstupních hodnot $L_{Rin,act} = \langle -30, 0 \rangle$ dBm a z velikosti útlumu ozvěnové cesty minimálně $ERL = 6$ dB.

V testu 2A se postupně volají následující funkce:

1. Funkce pro vygenerování signálu Single Talk.

Jedná se o simulaci jednosměrného hovoru - funkce vytvoří složený testovací signál CSS (Composite Source Signal) podle doporučení [1]. Doporučení uvádí přesné délky znělé části, pauzy a počet opakování tak, aby výkonová hustota spektra byla podobná hustotě řeči. Délka znělé části je 48,62 ms, následuje pseudošum délky 200 ms a nakonec pauza 101,38 ms. Pseudošum se získá pomocí `fft` s délkou bloku 4096 vzorků, ve frekvenční oblasti se náhodně nagenarují frekvenční složky a zpětnou transformací se převedou do časové oblasti. Signál se pak ještě omezí pásmovým filtrem od 50 do 3680 Hz. Celá sekvence se pak opakuje až do požadované délky zadávané při spuštění.

2. Načtení vybrané impulsní odezvy prostředí echa.

V doporučení [1] přílohy D je odezva prostředí simulována pro různá zpoždění, útlumy ozvěnových cest s rozdílnými disperzními charakteristikami. Na výběr je v příloze celkem 8 ozvěnových cest, pro účely testu 2A jsem zvolil následující cesty:

- Konvergence:

Cesta č.1 je simulace hybridu s krátkou disperzní charakteristikou a vysokým útlumem $ERL=12$ dB

- Rekonvergence:

Cesta č.4, opět simulace hybridu s dlouhou disperzní charakteristikou a vysokým útlumem $ERL=12$ dB, následuje přechod na cestu č.1 s nízkým útlumem $ERL=2$ dB a poté na cestu č.6 což je jednoduchý odraz

s disperzí 6 ms který odpovídá průměrné impulsní odezvě ze všech odezev uvedených v doporučení [1] G.168.

3. Kompozitní CSS signál SingleTalk se zkonvoluje s vybranou impulsní odezvou.

Pro konvoluci se zde volí matlabovská funkce `filtr`, nastavená na FIR variantu.

4. Zavolá se funkce pro potlačení echa.

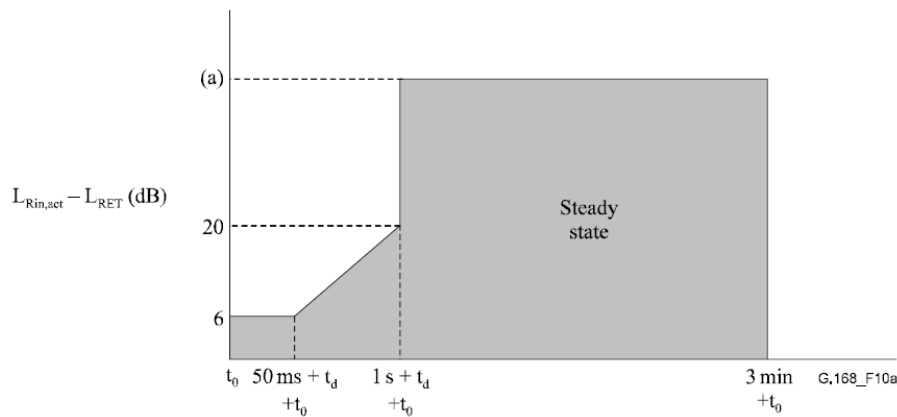
Potlačení pro LMS, NLMS nebo RLS se volá jako samostatné funkce, obecně se nejdřív testuje, jestli je na blízkém konci řečový signál, pokud ano, pak se provádí pouze filtrace bez adaptace. Pro test 2A podmínka neplatí takže se provádí zrovna filtrace s adaptací. Funkce potlačení se volá celkem čtyřikrát, poprvé pro konvergenci s koeficienty filtru nastavenými na nulu, a poté třikrát pro rekonvergenční testy, které na začátku pracují s filtrem konvergovaným na předchozí ozvěnovou cestu.

5. Poslední funkcí je funkce měřící úroveň signálu na výstupu z potlačovače.

Jedná se o `rms` funkci používanou pro měření efektivní hodnoty signálů, tedy průměru druhých mocnin vzorků. Výstup se normalizuje na signálový generátor 1mW. Rozlišuje se mezi částí signálu včetně pauz a bez pauz.

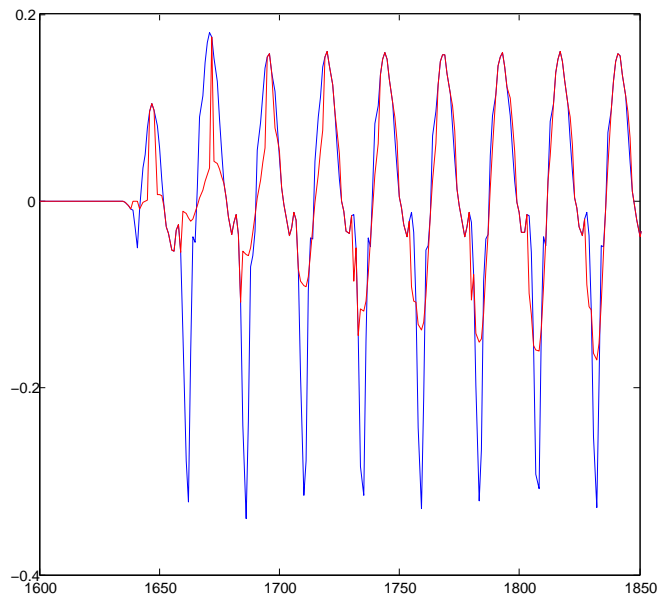
3.3 Diskuze výsledků

V obrázku 3.1 jsou zobrazeny časové charakteristiky, které by měl potlačovač nasažený do provozu splňovat. Jedná se o velikost útlumu aktivní části signálu ze vstupu na výstup jako funkce času, potlačovače by měly pracovat s hodnotou útlumu větší, než je zobrazená hranice. Ustálená hodnota (steady state) je odvozená z úrovně signálu ze vzdáleného konce, v testech je tato hodnota na hranici 55 dB. Úroveň potlačení i rychlost konvergence je závislá na různých veličinách, podle typu algoritmu, společným jmenovatelem je však délka filtru, která určuje míru toho, co se filtr "naučí." V příloze jsou výsledky testů pro tři různé délky filtru s použitím všech tří algoritmů, přičemž parametry konkrétních algoritmů byly nastaveny tak, aby dávaly nejlepší výsledky. Pro NLMS filtr se jedná o normalizační koeficient $\alpha = 0,55$ a ochranu proti dělení nulou $\sigma = 10^{-10}$, pro RLS filtr to je hodnota zapomínacího faktoru (forgetting factor) $\lambda = 0,97$ a startovací hodnota matice Ψ^{-1} na diagonále $= 1/0,005$.

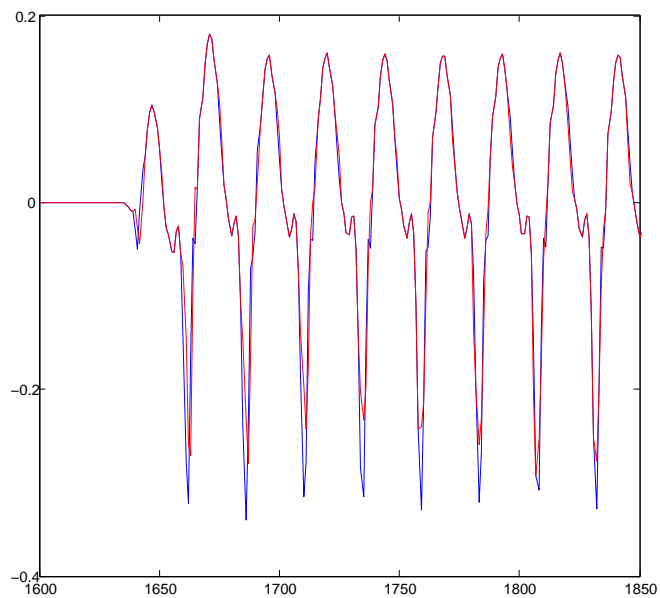


Obr. 3.1: Požadované konvergenční charakteristiky s povoleným NLP.
Převzato z doporučení ITU-T G.168, strana 19.

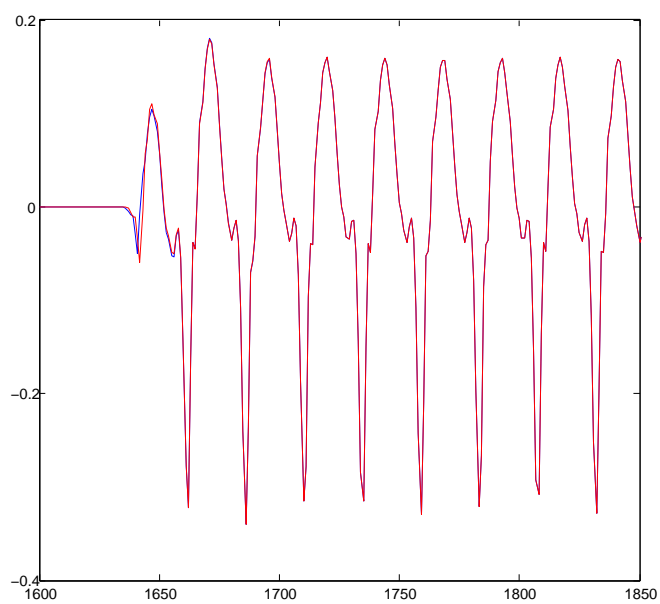
Na následujících třech obrázcích je srovnání časového průběhu adaptace všech tří algoritmů pro první periody znělé části řeči. Tato část byla konvolována s cestou č.1, na počátku adaptace byly koeficienty vynulované, u všech algoritmů byla délka filtru w zvolena $n=160$. Ze srovnání podle předpokladů nejlépe vychází algoritmus RLS, nejdelší dobu přizpůsobení má LMS algoritmus. Na obrázcích je modře zobrazen časový průběh počátku znělé části zkonvolované s impulsní odezvou cesty 1, červeně pak výstupní signál filtru - v ideálním případě by se průběhy měly začít překrývat v co nejkratším čase.



Obr. 3.2: Časový průběh počátku znělé části řeči zkonvolované s imp. odezvou cesty 1 - modře, průběh z postupně adaptovaného LMS filtru - červeně.



Obr. 3.3: Časový průběh počátku znělé části řeči zkonvolované s imp. odezvou cesty 1 - modře, průběh z postupně adaptovaného NLMS filtru - červeně.



Obr. 3.4: Časový průběh počátku znělé části řeči zkonvolované s imp. odezvou cesty 1 - modře, průběh z postupně adaptovaného RLS filtru - červeně.

4 ZÁVĚR

Při otestování algoritmů LMS, NLMS a RLS jsem navázal na diplomovou práci [3], ve které byly v Matlabu vytvořeny funkce pro testování potlačovačů podle doporučení ITU-T G.168 s použitím LMS algoritmu. Závěrem práce bylo zjištění podle něhož se v základní podobě LMS algoritmus pro potlačovače nehodí. Na tuto práci jsem navázal a provedl optimalizaci skriptů tak, aby se zrychlilo testování, díky čemuž je běh přibližně o řád rychlejší a sada skriptů je přehlednější. Dalším krokem byla normalizace LMS algoritmu a jeho testování, závěrem je zjištění podle něž by s vhodně nastavenými parametry NLMS algoritmus mohl být nasazen do potlačovačů, pokud bychom nepovažovali požadavek na úroveň vráceného echa na začátku ustáleného stavu za příliš zavazující (viz. příloha). Zavedení RLS algoritmu řeší požadavky na potlačovač velmi dobře, rychlost konvergence je naprosto dostačující a filtr je plně adaptovaný již po několika málo iteracích, problémem je však náročnost výpočtu, která je přibližně o řád větší než u NLMS algoritmu. Proto by se do potlačovačů echa hodily spíše některé z rychlejších metod, jako např. FTF (Fast Transversal Filter) nebo FLMS (Fast Least Mean Square).

LITERATURA

- [1] Recommendation ITU-T.G168. *Digital network echo cancellers*. 08/2004.
- [2] Recommendation ITU-T.G167. *Acoustic Echo Controllers*. 03/93.
- [3] Lajtkep, T. *Potlačovač Echa podle doporučení G.168 -Semestrální práce*. Brno: Vysoké Učení Technické v Brně, 2010. 31 s.
- [4] B.Fahrang-Boroujeny *Adaptive Filters*. England: John Wiley & sons Ltd, 2005. 529 s.
- [5] Saeed V. Vaseghi *Advanced Digital Signal Processing and Noise Reduction*. England: John Wiley & sons Ltd, 2000. 317 s.
- [6] MALENOVSKÝ, Vladimír. Elektrotechnika [online]. 2002 [cit. 2011-12-14]. *Adaptivní filtrace zašuměných řečových signálů*. Dostupné z WWW: <<http://www.elektrotechnika.cz/clanky/02063/index.html>>
- [7] R.Barry, E.A.Lee, D.G.Messerschmitt [online] [cit. 2012-03-10] *Echo Cancellation*. Dostupné z WWW:<<http://users.ece.gatech.edu/barry/digital/-supp/20echo.pdf>>

A SOUBORY NA CD

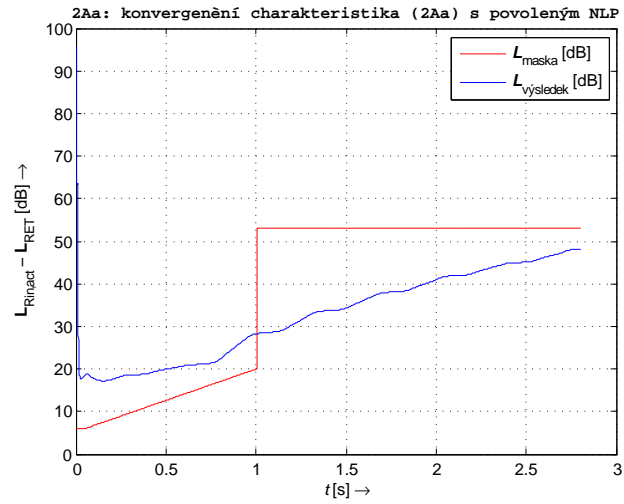
- Adresář /PDF obsahuje tuto práci v souboru bc_prace.pdf a dále doporučení ITU-T G.168 a G.167.
- Adresář /OBR obsahuje obrázky použité v této práci ve formátu pdf.
- Adresář /MATLAB obsahuje testovací soubory jednotlivých algoritmů a další funkce nutné pro běh. Jedná se o tyto soubory:

¹

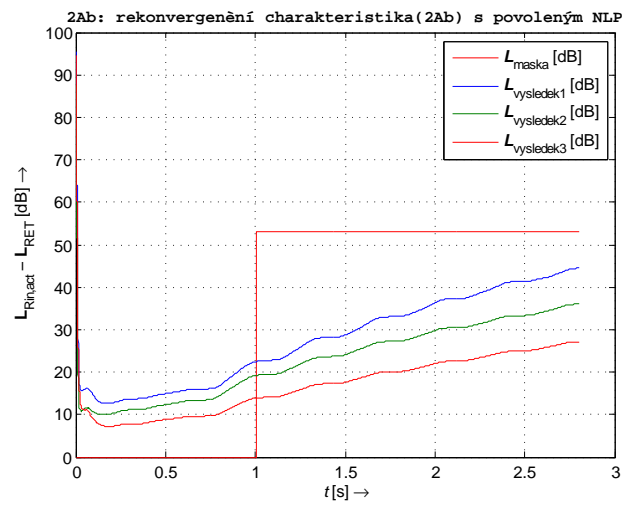
- `test2A.m`, hlavní skript, který volá všechny ostatní funkce z adresáře. Na řádce č.45 je definice globální struktury jejíž částí je i délka adaptivního filtru, zde je možné testovat vliv délky filtru na konvergenci. Ostatní inicializační parametry jsou ve skriptu okomentovány, tak jako i funkce, které jsou skriptem dále volány. Na řádce č.111 a č.133 je pak možné odkomentováním řádku vybrat algoritmus RLS nebo NLMS pro konvergenci a rekonvergenci.
 - `newEchoCancellor.m` vytvoří strukturu obsahující globální proměnné a konstanty, zde je možné měnit některá nastavení. Tato funkce dále volá :
`newLMSFilter.m`,
`NearSpeechDetector.m`,
`newNearSpeechDetector.m`,
`newResErrSupp.m`.
 - `EchoCancellation_LMSN.m` provádí filtraci a adaptaci NLMS, pokud chceme pouze LMS algoritmus, pak na řádce č.69 odkomentujeme konstantu `new_mu`, čímž se přepíše hodnota normalizovaného kroku nějakou zadanou konstantou.
 - `EchoCancellation_RLS.m` provádí filtraci a adaptaci RLS, zde se nastavuje hodnota tzv. zapomínacího faktoru λ .
 - `rms.m` provádí měření úrovně vráceného echa.
 - `echoPath.m` obsahuje všechny impulsní odezvy definované v [1], konkrétní cesta se vybírá ve skriptu `text2A.m`
- Soubor popisující obsah CD

¹Poznámka: Skripty běžely ve verzi Matlab 7.12

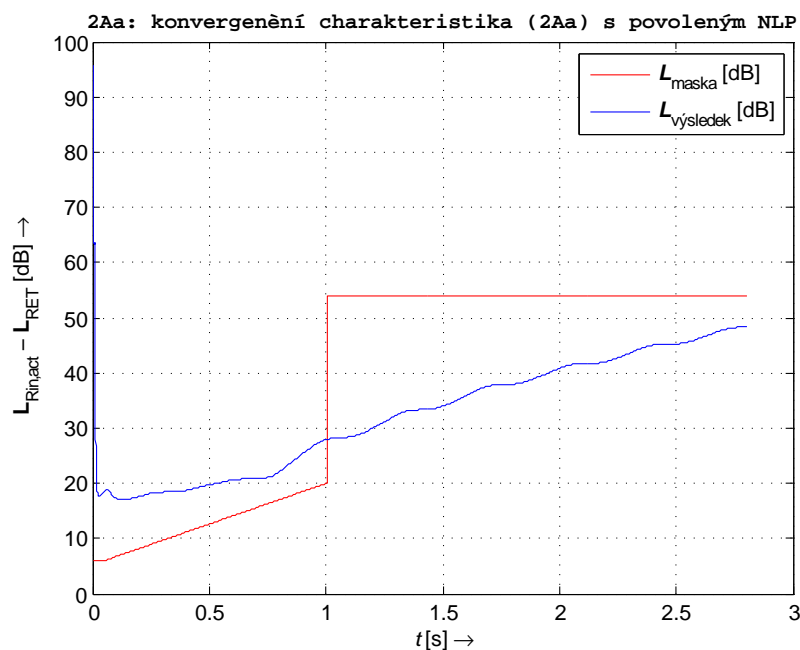
B LMS TESTY



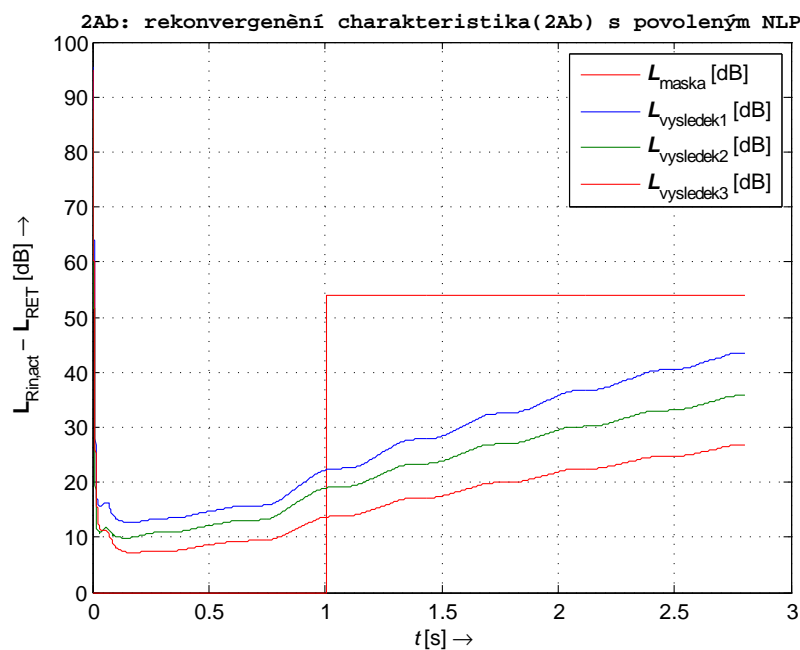
Obr. B.1: Konvergenční charakteristika LMS filtru s povoleným NLP.
Délka filtru 160 vzorků



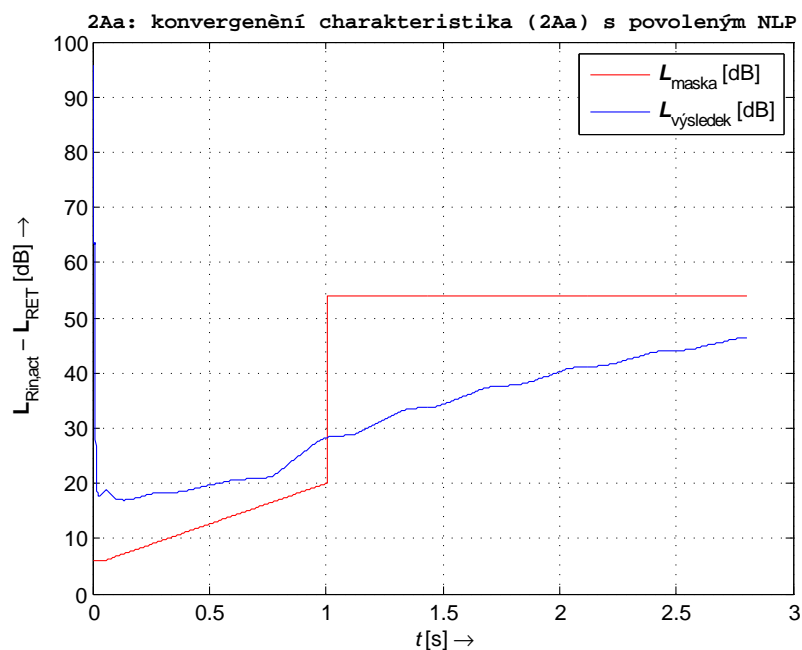
Obr. B.2: Rekonvergenční charakteristika LMS filtru s povoleným NLP.
Délka filtru 160 vzorků. Modře je vyznačena rekonvergence z cesty 1 na cestu 4,
zeleně z cesty 4 na cestu 1 a červeně z cesty 1 na cestu 6.



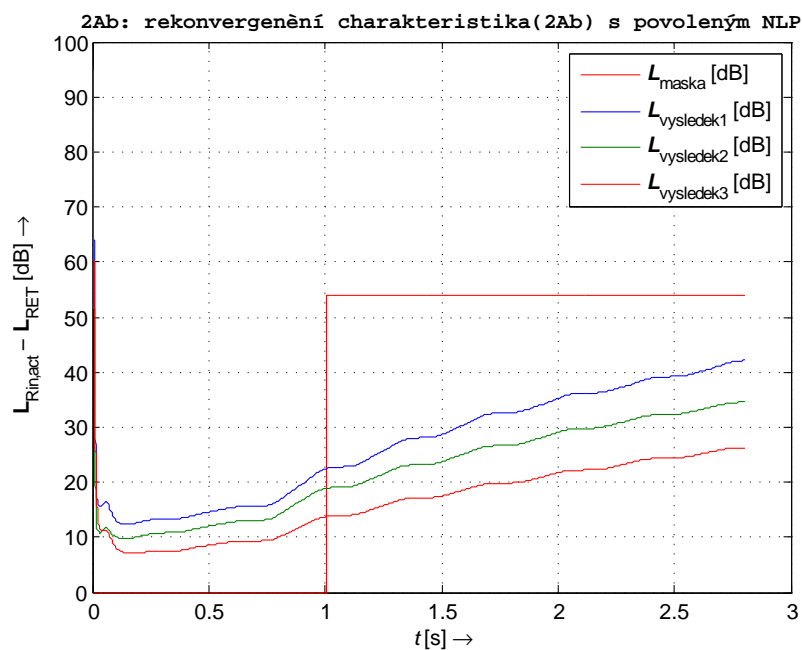
Obr. B.3: Konvergenční charakteristika LMS filtru s povoleným NLP.
Délka filtru 200 vzorků



Obr. B.4: Rekonvergenční charakteristika LMS filtru s povoleným NLP.
Délka filtru 200 vzorků. Modře je vyznačena rekonvergence z cesty 1 na cestu 4,
zeleně z cesty 4 na cestu 1 a červeně z cesty 1 na cestu 6.



Obr. B.5: Konvergenční charakteristika LMS filtru s povoleným NLP.
Délka filtru 300 vzorků



Obr. B.6: Rekonvergenční charakteristika LMS filtru s povoleným NLP.
Délka filtru 300 vzorků. Modře je vyznačena rekonvergence z cesty 1 na cestu 4,
zeleně z cesty 4 na cestu 1 a červeně z cesty 1 na cestu 6.

C NLMS TESTY

```
function [Error]=EchoCancellation(Sin)
% provede potlaceni echa
%=====
##### N A S T A V E N I   P R O M E N N Y C H #####
global Rin;
global Can2;

% koeficienty pro normalizaci
alpha = 0.55;
sigma = 1E-10; %kvuli deleni nulou

%prekopiruju si cast struktury tak abych nemusel vypisovat cely nazev
rho = Can2.ResErrSupp.rho;

% delka filtru – pocet vzorku z definice struktury Can2
M = length(Can2.AdaptFilter.Coeff);

Rin_select(1:M) = 0;
Filtr_Output(1:M) = 0;

if( any( size( Sin) ~= size( Rin)))
    error( 'Rin a Sin musi mit stejnou delku');
end

for( k = M:length( Rin))

    % ##### DETEKCE SIGNALU NA BLIZKEM KONCI #####

    %pokud se nadetkuje signal na blizkem konci, tak se jenom filtruje

    [NearDetect, EchoCancellor.NearSpeechDetector] = ...
NearSpeechDetect( EchoCancellor.NearSpeechDetector, Sin(k), Rin(k));

    if( NearDetect)
        [EchoEstimate, EchoCancellor.AdaptFilter] = ...
LMSFiltering( EchoCancellor.AdaptFilter, Rin(k));

        ##### F I L T R A C E #####
    end
end
end
```

```

%vyberu cast Rin a otocim ho kvuli konvoluci
Rin_select = Rin(k:-1:k-M+1);

%provedu filtraci s aktualnimi koeficienty filtru
y_k = Can2.AdaptFilter.Coeff*Rin_select;
Can2.AdaptFilter.State=[Rin(k), ...
    Can2.AdaptFilter.State(1:end-1)];

Error(k) = Sin(k) - y_k;

##### A D A P T A C E      F I L T R U #####

% Tady je normalizace -vstupni vektor se prenasobi transvezalne
% tim samym, dostaneme jedno cislo
new_mu = alpha / ((Rin_select'*Rin_select) +sigma);

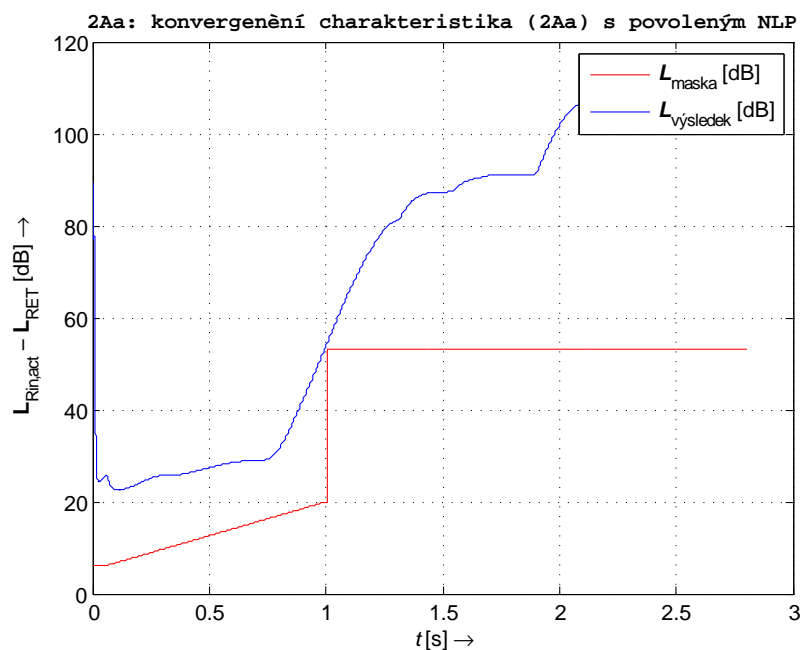
Can2.AdaptFilter.Coeff = Can2.AdaptFilter.Coeff ...
+ 2*new_mu*Error(k)*Rin_select';

% ##### P O T L A C E N I Z B Y T K O V E H O E C H A #####

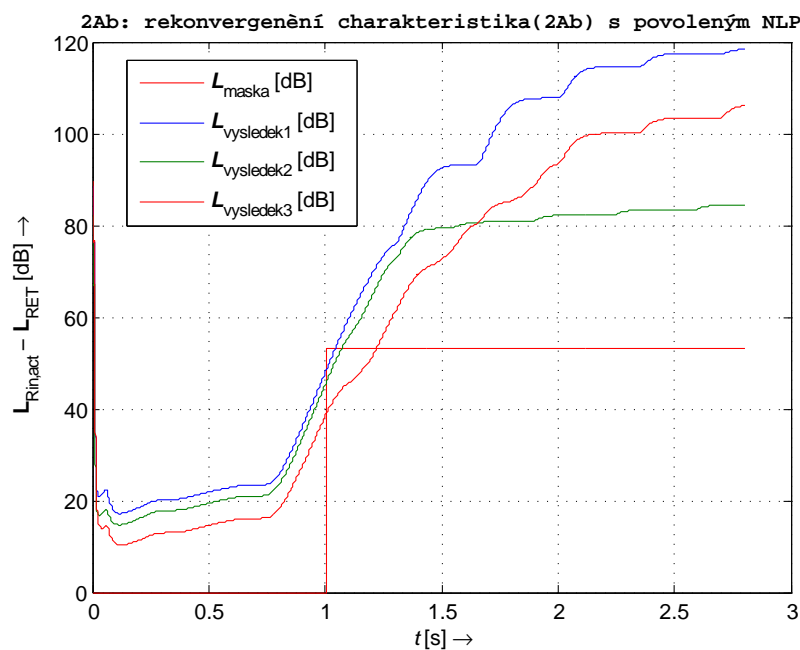
if strcmp(Can2.NLP, 'on')
    Can2.ResErrSupp.NearLevel=(1-rho)*Can2.ResErrSupp.NearLevel...
        + rho*abs( Error(k) );
    Can2.ResErrSupp.FarLevel = (1-rho)*Can2.ResErrSupp.FarLevel...
        + rho*abs( Rin(k) );
    if(Can2.ResErrSupp.NearLevel < ...
        Can2.ResErrSupp.Threshold *Can2.ResErrSupp.FarLevel)
        %
        Error(k) = 0;
    end
end
end

end

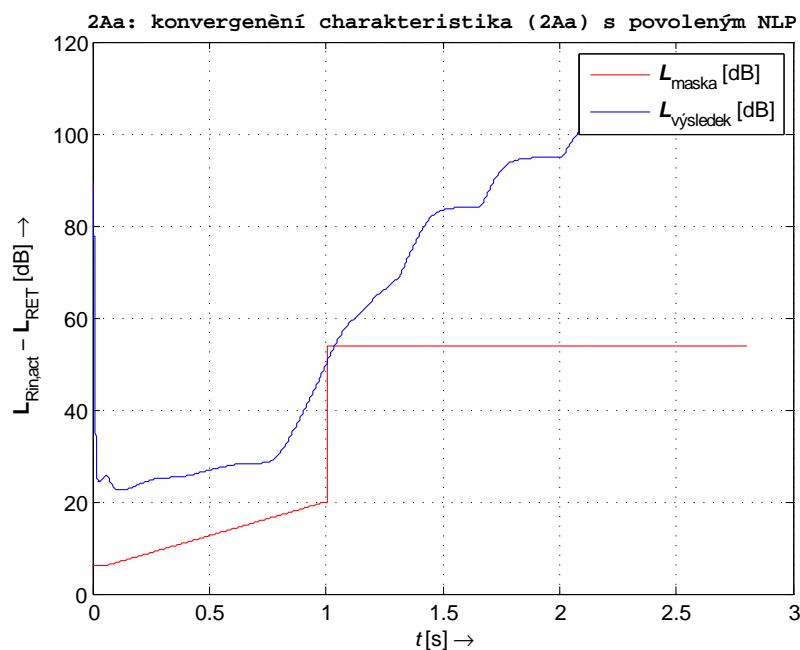
```



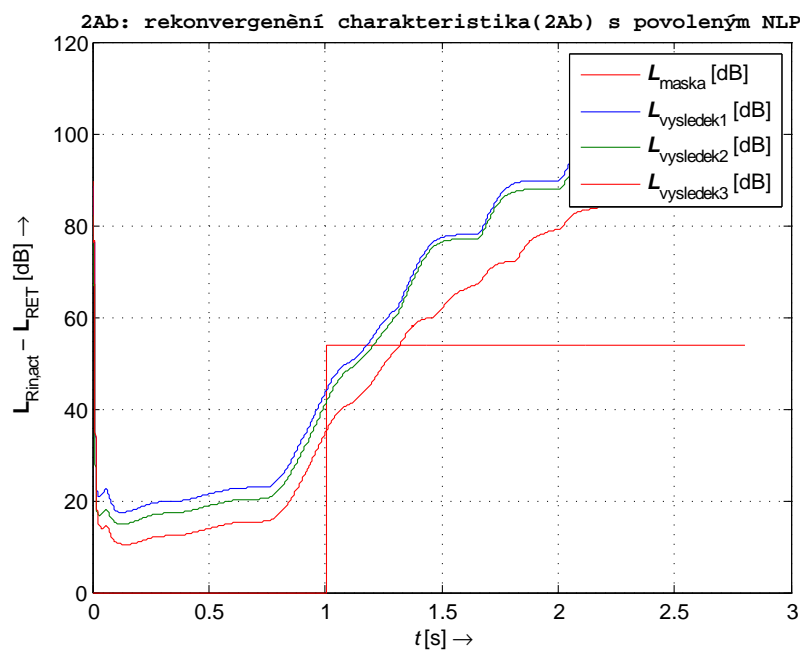
Obr. C.1: Konvergenční charakteristika NLMS filtru s povoleným NLP.
Délka filtru 160 vzorků



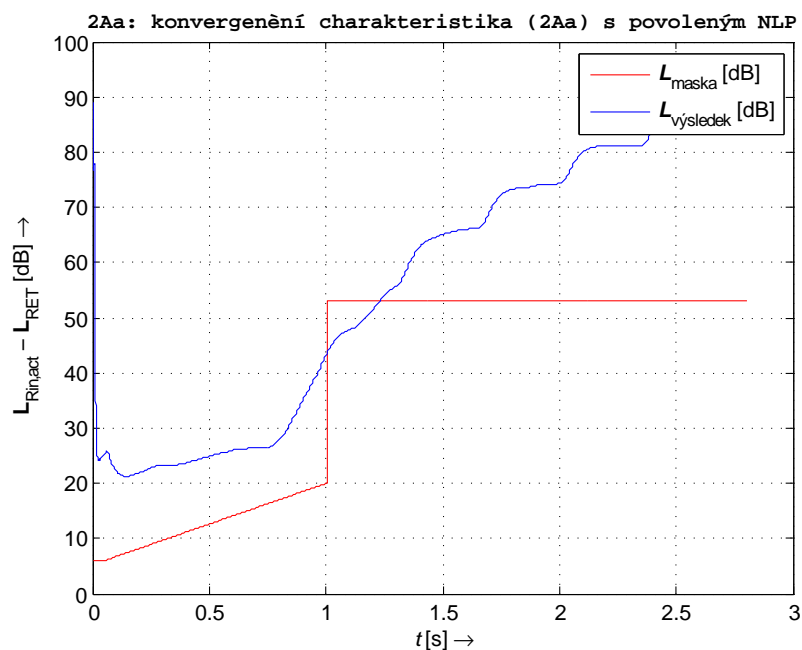
Obr. C.2: Rekonvergenční charakteristika NLMS filtru s povoleným NLP.
Délka filtru 160 vzorků. Modře je vyznačena rekonvergence z cesty 1 na cestu 4,
zeleně z cesty 4 na cestu 1 a červeně z cesty 1 na cestu 6.



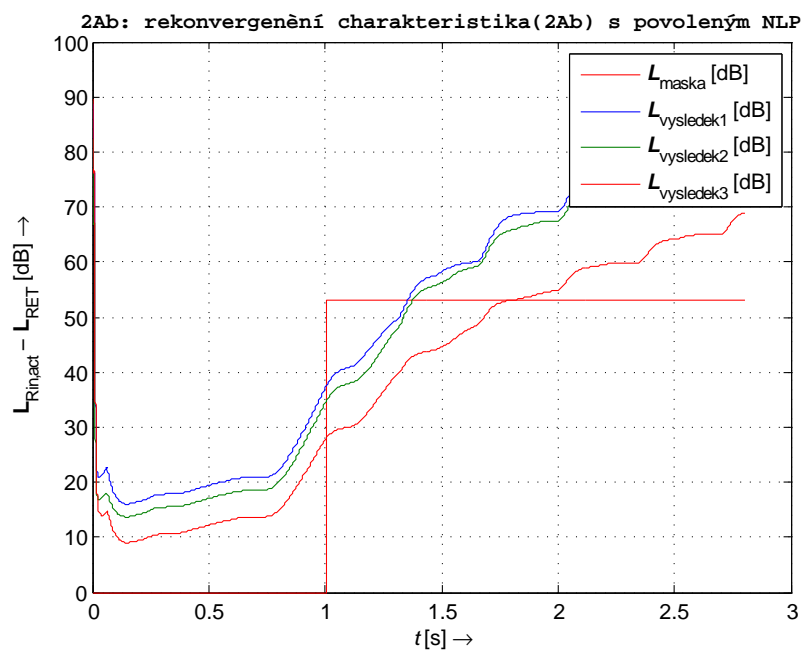
Obr. C.3: Konvergenční charakteristika NLMS filtru s povoleným NLP.
Délka filtru 200 vzorků



Obr. C.4: Rekonvergenční charakteristika NLMS filtru s povoleným NLP.
Délka filtru 200 vzorků. Modře je vyznačena rekonvergence z cesty 1 na cestu 4,
zeleně z cesty 4 na cestu 1 a červeně z cesty 1 na cestu 6.



Obr. C.5: Konvergenční charakteristika NLMS filtru s povoleným NLP.
Délka filtru 300 vzorků



Obr. C.6: Rekonvergenční charakteristika NLMS filtru s povoleným NLP.
Délka filtru 300 vzorků. Modře je vyznačena rekonvergence z cesty 1 na cestu 4,
zeleně z cesty 4 na cestu 1 a červeně z cesty 1 na cestu 6.

D RLS TESTY

```
function [Error]=EchoCancellation(Sin)
% provede potlaceni echa
%=====
##### D E K L A R A C E      P R O M E N N Y C H #####

global Rin;
global Can2;

lambda = 0.97;      %forgetting faktor
delta = 0.005;     %matice Po -prevrac. hodnota na diagonale

%M = 256;          %delka filtru
M = length(Can2.AdaptFilter.Coeff);

N = length(Rin);   %delka vstupniho vektoru

%               i n i c i a l i z a c e

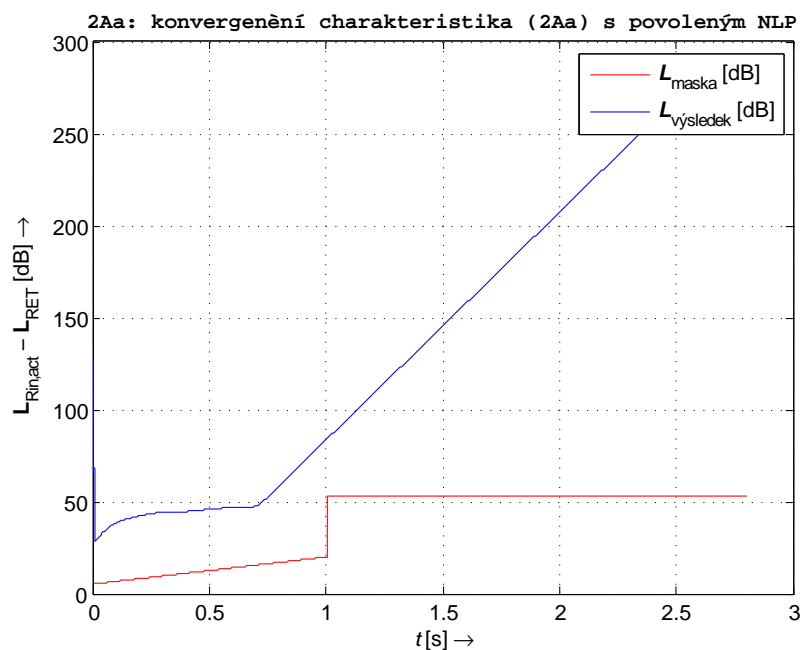
Error = zeros(N,1); %inicializace vystupu
w = zeros(M, 1);   %koeficienty filtru
w = Can2.AdaptFilter.Coeff';
P = eye(M)/delta;  %diag. matice delky filtru M

Error = Sin;       %chybovy vektor

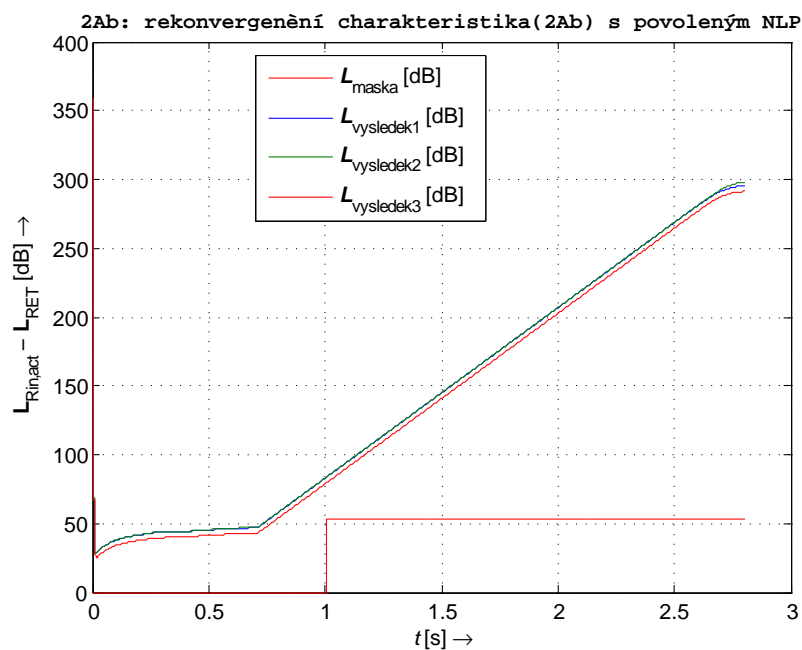
##### R L S      A L G O R I T M U S #####

for n=M:N
    Rin_rev=Rin(n:-1:n-M+1);
    k=lambda^(-1)*P*Rin_rev/(1+lambda^(-1)*Rin_rev'*P*Rin_rev);
    Error(n)=Sin(n)-w'*Rin_rev;
    w=w+k*Error(n);
    P=lambda^(-1)*P-lambda^(-1)*k*Rin_rev'*P;
end

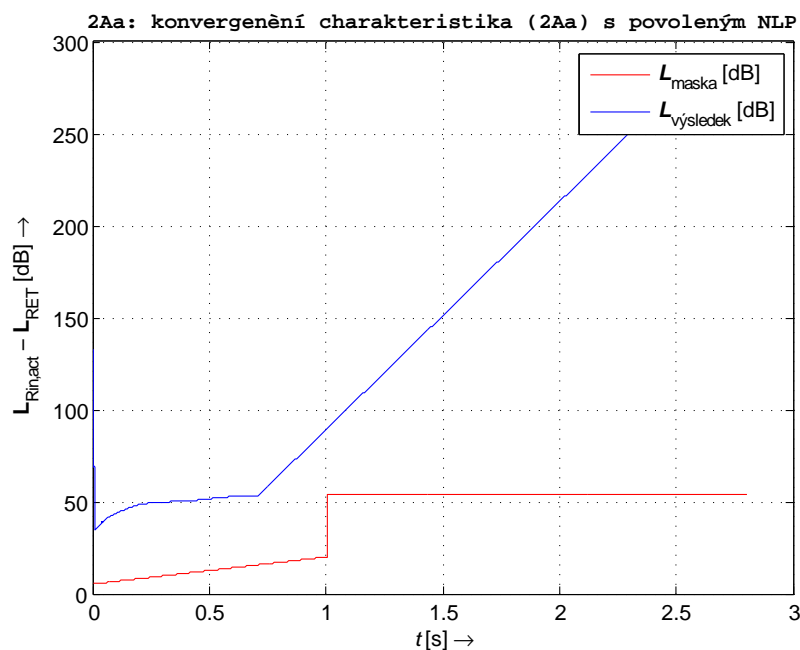
Can2.AdaptFilter.Coeff = w';
```



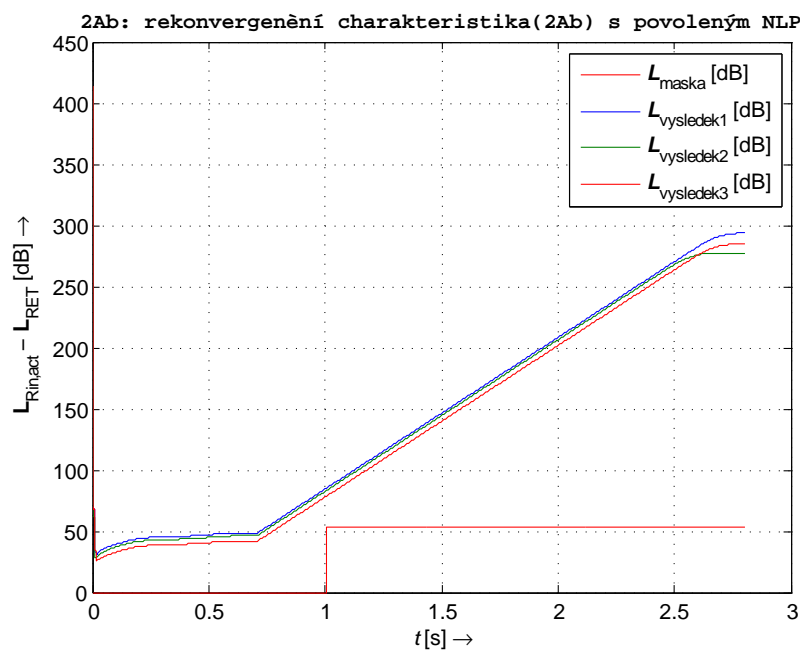
Obr. D.1: Konvergenční charakteristika RLS filtru s povoleným NLP.
Délka filtru 160 vzorků



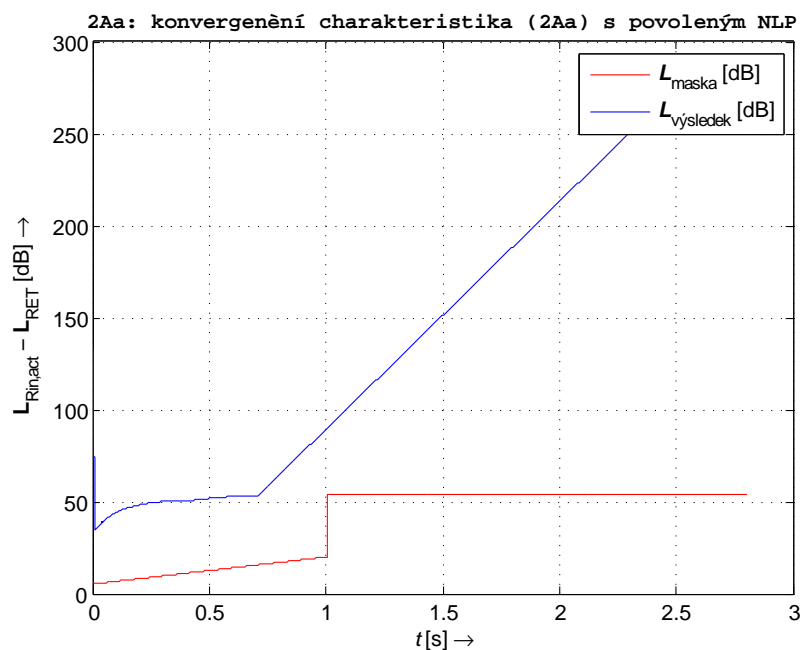
Obr. D.2: Rekonvergenční charakteristika RLS filtru s povoleným NLP.
Délka filtru 160 vzorků. Modře je vyznačena rekonvergence z cesty 1 na cestu 4,
zeleně z cesty 4 na cestu 1 a červeně z cesty 1 na cestu 6.



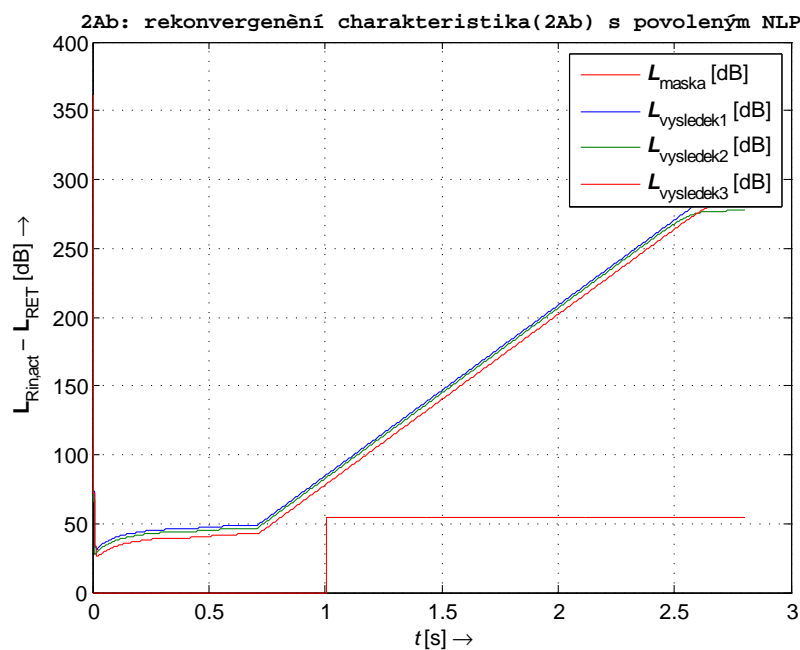
Obr. D.3: Konvergenční charakteristika RLS filtru s povoleným NLP.
Délka filtru 200 vzorků



Obr. D.4: Rekonvergenční charakteristika RLS filtru s povoleným NLP.
Délka filtru 200 vzorků. Modře je vyznačena rekonvergence z cesty 1 na cestu 4,
zeleně z cesty 4 na cestu 1 a červeně z cesty 1 na cestu 6.



Obr. D.5: Konvergenční charakteristika RLS filtru s povoleným NLP.
Délka filtru 300 vzorků



Obr. D.6: Rekonvergenční charakteristika RLS filtru s povoleným NLP.
Délka filtru 300 vzorků. Modře je vyznačena rekonvergence z cesty 1 na cestu 4,
zeleně z cesty 4 na cestu 1 a červeně z cesty 1 na cestu 6.