



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

**Fakulta elektrotechnická
katedra radioelektroniky**

Detekce vad u mechanického záznamu zvuku

Defect detection at mechanical sound records

Diplomová práce

Studijní program: Komunikace, multimédia a elektronika
Studijní obor: Multimediální technika

Vedoucí práce: Ing. František Rund, Ph.D.

Bc. Michal Bolek

Praha 2016

České vysoké učení technické v Praze
Fakulta elektrotechnická
katedra radioelektroniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Michal Bolek**

Studijní program: Komunikace, multimédia a elektronika
Obor: Multimediální technika

Název tématu: **Detekce vad u mechanického záznamu zvuku**

Pokyny pro vypracování:

Seznamte se s typickými vadami mechanického záznamu, které vznikají během výrobního procesu. Studujte algoritmy detekce uvedených vad. Vybranou metodu implementujte a vyzkoušejte na reálných vzorcích, zaměřte se na eliminaci falešných detekcí. Studujte možnosti další klasifikace vad, s ohledem na identifikaci fáze výroby, ve které daná vada vznikla.

Seznam odborné literatury:

- [1] Dušek, K. Záznam a reprodukce zvuku. 5 vydání. Praha: Vydavatelství ČVUT, 1982.
- [2] Godsill, S J, Rayner, P J. Digital audio restoration: a statistical model based approach. New York: Springer, 1998, xix, 328 p. ISBN 3540762221.

Vedoucí: Ing. František Rund, Ph.D.

Platnost zadání: do konce letního semestru 2016/2017

L.S.

doc. Mgr. Petr Páta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 19. 2. 2016

Abstrakt

Práce se zabývá mechanickým záznamem a jeho vadami, které vznikají při výrobním procesu. Konkrétně se jedná o impulsní vady. Práce dále seznamuje s algoritmy pro detekci těchto vad, jako jsou autoregresní modelování, vlnková transformace a derivace signálu. Dále jsou představeny metody pro omezení falešné detekce, která u těchto algoritmů vzniká. Součástí práce jsou vytvořené algoritmy v programovém prostředí Matlab, které detekují vady pomocí uvedených metod a dále ověřují správnost detekce. Algoritmy jsou nakonec otestovány na reálných ukázkách obsahující vady a z výsledků plyne, že pro účely výstupní kontroly dosahují poměrně dobrých výsledků.

Klíčová slova

mechanický záznam, detekce vad, impulsní šum, autoregresní model, vlnková transformace, derivace, Matlab

Abstract

The thesis is concerned with mechanical recording and its defects, which result from manufacturing process. Namely the defects are impulse noise. The thesis introduces the algorithms for detection of these defects, in particular the methods are an autoregressive model, a wavelet transform and derivatives of a signal. Furthermore the methods for the limitation of a false detection are introduced. A component part of the thesis are the algorithms developed in Matlab, which are able to detect these defects and also verify the correctness of the detection. The algorithms are in the end tested on naturally degraded signals and in according of the results the algorithms work fairly well.

Keywords

mechanical recording, defect detection, impulse noise , autoregressive model, wavelet transform, derivatives of a signal, Matlab

Prohlášení

„Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokškolských závěrečných prací.“

V Praze dne

.....

Poděkování

Chtěl bych poděkovat svému vedoucímu práce Inq. Františkovi Rundovi, Ph.D., za cenné připomínky a rady při tvorbě této práce.

Obsah

1 Úvod.....	7
2 Mechanický záznam.....	8
2.1 Princip záznamu	8
2.2 Výroba záznamu.....	9
2.3 Typické vady na záznamu	9
3 Detekce vad.....	12
3.1 AR model	12
3.2 Vlnková transformace.....	17
3.3 Derivace	20
4 Ověření detekce vad	22
4.1 Krátkodobá Fourierova transformace.....	22
4.2 Průběh součtové a rozdílové složky.....	24
5 Implementace	26
5.1 AR model	26
5.2 Vlnková transformace.....	27
5.3 Derivace	28
5.4 Další zpracování a ověření detekce	28
6 Testování a výsledky	31
6.1 Parametry AR modelu.....	31
6.2 Parametry vlnkové transformace.....	34
6.3 Parametry derivace	36
6.4 Úroveň detekce a ověření detekce.....	38
6.5 Vyhodnocení.....	41
6.6 Možnosti klasifikace.....	43
7 Závěr	45
Literatura	46
Příloha A	47
Příloha B.....	49

1 Úvod

Mechanický záznam se v posledních letech začal opět vyrábět ve větším množství. Problém je, že ve výrobním procesu mohou vznikat různé vady, které se ve výsledném výrobku typicky projevují jako krátké zvukové impulzy při přehrávání. Tyto vady je třeba při výstupní kontrole výrobků detekovat a podle nalezené vady ideálně identifikovat místo výrobního procesu, kde tato vada vznikla. Tato kontrola je většinou prováděna poslechem vyrobené desky pracovníkem, který si musí poslechnout celou desku. Kontrolu by tedy bylo vhodné zefektivnit postupem, který by automaticky našel poškozená místa a pracovník by jen vyhodnotil závažnost vady. Jednou možností může být optická kontrola, která bude hledat poškozené drážky. Další možností, kterou se dále zabývá tato práce je celý záznam digitalizovat a využít k nalezení vad metod číslicového zpracování signálu.

Cílem práce je tedy seznámení se s problematikou mechanického záznamu a jeho typických vad. A dále pak vytvoření algoritmu, který tyto vady dokáže v signálu detekovat.

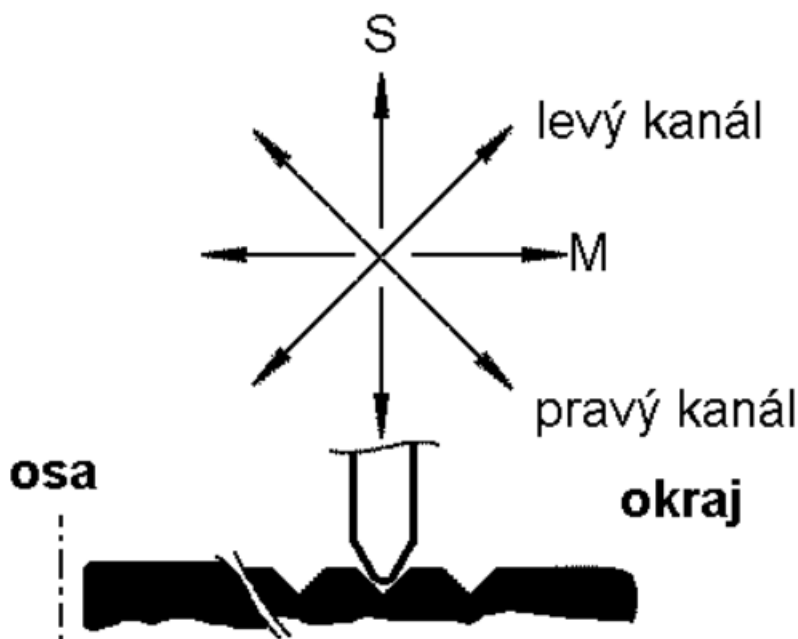
V první části se práce zaměřuje na princip mechanického záznamu a jeho typické vady, které vznikají při jeho výrobě. V kapitole 3 pak práce popisuje metody, jak tyto vady detekovat v digitalizovaném signálu. V kapitole 4 jsou dále uvedeny metody jak eliminovat falešnou detekci, která může při hledání vad nastat. V kapitole 5 je dále popsána implementace těchto metod v prostředí Matlab. Kapitola 6 se nakonec zabývá testováním těchto algoritmů na reálných vzorcích a jsou diskutovány výsledky detekce. Dále jsou pak uvedeny možnosti klasifikace, některých vad.

2 Mechanický záznam

2.1 Princip záznamu

Mechanický záznam zvuku je založen na mechanické deformaci záznamového materiálu při procesu záznamu vhodným nástrojem, tj. rycí jehlou, úměrně k okamžité hodnotě zvukového signálu, který přivádíme do elektricko-mechanického měniče zvaného řezací hlava. Záznam se provádí na kruhovou desku záznamového materiálu, která se otáčí rovnoměrnou úhlovou rychlostí. Základními typy záznamu jsou hloubkový a stranový. Hloubkový záznam je takový, při kterém se rycí jehla pohybuje při záznamu ve směru kolmém k rovině záznamového materiálu. Stranový záznam je pak takový, při kterém se rycí jehla pohybuje ve směru rovnoběžném [1].

Stereofonní záznam se zaznamenává kombinací dvou záznamů, které jsou v navzájem kolmých rovinách, které svírají s povrchem desky úhel 45°. Oba signály musí být v tomto případě pólovány tak, aby výsledný pohyb hrotu byl převážně stranový s nevelkou hloubkovou složkou. Princip tohoto záznamu je zobrazen na obr. 2.1. Umístění jednotlivých signálů v drážce je následující: pravý kanál je nahrán na vnějším boku drážky, čili směrem k okraji desky a signál pro levý kanál na vnitřní straně drážky [1].



Obr. 2.1: Princip stereofonního mechanického záznamu. Upraveno z [2]

Reprodukce záznamu se pak provádí pomocí mechanicko-elektrického měniče – přenosky, na jejichž svorkách dostáváme zvukový signál ve formě elektrického napětí. Samotný měnič, ve kterém dochází ke změně mechanického pohybu v elektrickou energii, se nazývá systém přenosky a bývá různý, podle fyzikálního principu přenosky. Nejpoužívanějšími fyzikálními principy jsou piezoelektrický, elektromagnetický a elektrodynamický. Snímání stereofonního záznamu se provádí jedním hrotem, který pohání dva mechanicko-elektrické měniče, z nichž každý je citlivý pouze na jednu složku záznam, čili každý systém snímá záznam jednoho kanálu [1].

2.2 Výroba záznamu

Výroba záznamu probíhá pomocí lisovacích matic, které do vinylové směsi za tepla vylisují odpovídající zvukový záznam. V současnosti existují dvě technologie, pomocí kterých lze k tomuto výsledku dojít. První a starší z nich je výroba pomocí lakových fólií. Zvukový signál se zde pomocí řezací hlavy přenesne na desku masteru, která je pokrytá tenkou folií. Hlava do této fólie vyryje drážku odpovídající zvukovému záznamu. Lisovací matrice je poté z masteru vyrobena pomocí tří kroků galvanoplastických procesů. Nejdříve je nutno záznam na fólii udělat elektricky vodivý. Toho je dosaženo pomocí vodivé vrstvy stříbra, které se nanese na povrch. Pomocí prvního kroku galvanoplastiky se pak vytvoří originál v podobě niklové desky. U originálu drážky vystupují nad povrch, což odpovídá negativu záznamu. Z originálu se pak vytváří několik niklových kopií dalším galvanoplastickým procesem. Tyto kopie odpovídají pozitivnímu záznamu. V posledním kroku galvanoplastiky se pak z těchto pozitivních kopií vytvoří samotné lisovací matrice, které mají opět negativní orientaci. Pomocí jedné matrice pak lze vylisovat kolem 500-1000 vinylových desek. Mezi výhody této technologie patří zejména možnost zaznamenat širší a hlubší drážky, pomocí kterých lze vyrobit velmi hlasité nahrávky [1,3].

Druhou technologií je pak DMM (Direct Metal Mastering). Zde se zvukový signál vyryje pomocí hlavy do měděné vrstvy masteru. Z takto vytvořeného originálu se pak rovnou zhotoví lisovací matrice pomocí jednoho kroku galvanoplastiky. Výhodou této technologie je zejména lepší reprodukce vyšších frekvencí, lepší odstup signálu od šumu, lepší využití místa díky užším drážkám a v neposlední řadě výroba matic jen jedním galvanoplastickým procesem [3].

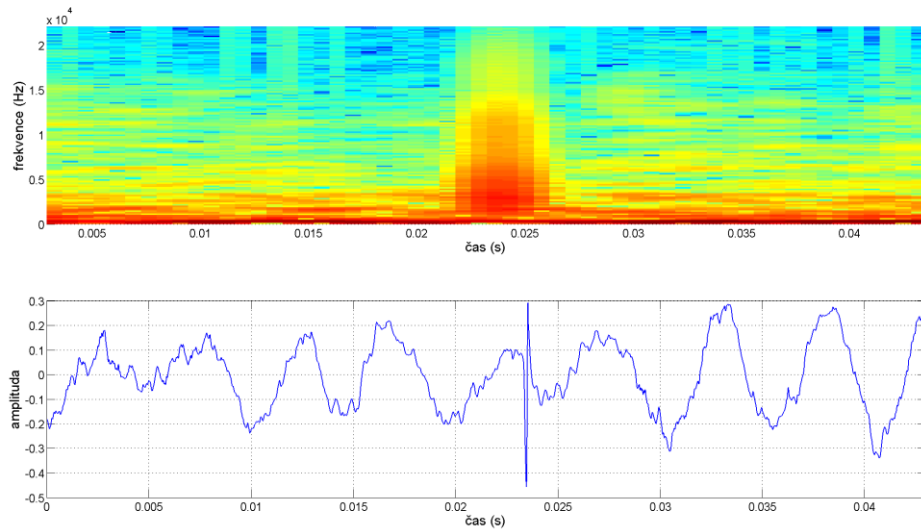
2.3 Typické vady na záznamu

V předchozí části byl popsán postup výroby vinylových desek. Tato výroba se skládá z několika částí a v každé z nich se může objevit nějaký problém, který vyústí v nějakou vadu ve finálním výrobku. Mezi typické a nejvíce rušivé patří impulsní rušení, které se projevuje lupancem či praskotem nebo podobných zvukovým projevem. V tabulce 2.1 poskytnutou firmou GZ Media, a.s je uveden seznam vad, které se projevují právě tímto charakterem. Z tabulky je také patrné, v jakých místech výroby tyto vady vznikají.

Zmíněné vady se typicky vyznačují plochým spektrem, kde jsou všechny složky spektra zastoupeny v podobné míře. Impulsní vady se zároveň vyskytují jen v některých místech signálu a nedegradují signál v celé délce jako např. šum. Na obr. 2.2 je zobrazen časový průběh a spektrogram lupance, což je typický projev všech zmíněných vad. Lze pozorovat zmíněnou vlastnost plochého spektra. V dalších částech se práce zaměřuje právě na tyto impulsní vady a jejich detekci.

Tab. 2.1: Seznam impulsních vad poskytnutý firmou GZ Media, a.s

Lupanec/narážka	Obecná porucha, která se projevuje lupancem. Jeho intenzita je různá. Důvodem této poruchy může být: elektrostatický lupanec, nečistota v drážce, prořez, přeskok, škrábanec a dokonce i chyba v podkladu od zákazníka.
Praskot	Posloupnost lupanců s různou intenzitou. Praskot může být způsoben jak poruchami z galvanoplastiky, tak i kvůli poruchám z lisování.
Škrábanec	Na desce můžeme pozorovat čáru (úsečku), která je umístěna přes záznam. Pokud škrábanec zasahuje do dna drážky, tak se může projevit jako lupanec na otáčku. V případě mělkého škrábance (nepoškození drážky), který poškodí pouze zrcátko, se může projevovat jen jako optická vada. Pod mikroskopem pak můžeme pozorovat, jestli je škrábanec na negativu nebo pozitivu. Je velký rozdíl mezi škrábancem na matrici nebo jen na desce. V případě škrábance na matrici, je touto vadou zatížena veškerá výroba, která byla vyrobena z dané matrice.
Škráblá matrice	Při škrábnutí desky můžeme mít štěstí, že se poškodí pouze zrcátko a deska se pak nebude projevovat lupancem na otáčku. U škrábnuté matrice takové štěstí již nemáme. Matrice je totiž negativ a jakýkoliv škrábanec poškodí výstupky, z kterých jsou po vylisování dna drážek. Důsledkem toho pak vznikají různé vady, jako jsou například lupance na otáčku, samostatný lupanec a praskot. Tato vada je způsobena především špatnou manipulací s matricí.
Nedolisy	Tato vada se především projevuje rušivým praskáním v jednom kanálu (v pravém kanálu). Nedolisy jsou fyzicky umístěny v závětrí hran matrice. Ve většině případů je vada umístěna přes několik drážek a praskání tak můžeme slyšet na otáčku. Příčinou je lisování, kde hmota při lisování nedostatečně obteče matrici. Tento problém se obvykle objevuje ke konci stran. Opticky můžeme nedolisy pozorovat jako shluky bílých teček.
Škrták – pravý	Pravý škrták vzniká při rozebírání nástrojů (špatné nebo neplynulé rozebírání), kde dochází k opětovnému přiblížení a zarytí výstupků do jiných míst. Postižen je tak jak negativ, tak i pozitiv. V případě, že se porucha zvukově neprojevuje, došlo k poškození zrcátka. Pokud ano, došlo k poškození drážky (silnější souvislé praskání). Další možností jak může vzniknout pravý škrták, je takový, že při lisování o sebe obě matrice škrtnou. Poškozeny jsou pak obě matrice. Pravý a nepravý škrták lze od sebe rozeznat z následujících desek. Pokud je problém i na dalších, jedná se o škrták pravý. Opticky můžeme škrtáky pozorovat jako drobné tečky nebo čárky.
Škrták – nepravý	Nepravý škrták vzniká při lisování, kdy při manipulaci s deskou dojde o škrtnutí desky o matrici. Deska například zůstane přilepená na horní matrici, zapruží a plasticky se při snaze desku sundat vrátí zpět k matrici. Tím se poškodí deska. Pravé a nepravé škrtáky lze rozeznat z následující desky. Pokud i na ní není ve stejné oblasti škrták, jde o škrták nepravý. Opticky můžeme škrtáky pozorovat jako drobné tečky nebo čárky.
Návalky	Návalky vznikají při řezání do lakových fólií a jsou způsobeny: vlastnostmi lakové vrstvy, tvarem řezacího nože, topením (topení by mělo být nastaveno tak, aby byla drážka hladká) a řezací rychlostí. Důsledkem tohoto jevu při řezání dojde při následném procesu k tomu, že výsledné desky mají na okrajích drážky ostré hrany. Když dojde ke kontaktu s rukou nebo jiným tělesem, tak se návalky rozmažou po zrcátku desky a může tak dojít i k znečištění drážky. Tato vada způsobí nesmazatelnou šmouhu na desce. Dále také může způsobit problémy při technologických procesech v galvanoplastice a drobný praskot.
Smeť	Smeť je vada, která ve svém důsledku vzniká již při přípravě na lisování. Lisařka by měla otřít matrici ze zadní strany a zbavit jí veškerých nečistot. Pokud toto neudělá, tak se v průběhu lisování nečistota tlačí na matrici a ta se poškodí (výstupek na matrici), poté je tato vada již na všech deskách. Samozřejmě i může nastat, že se smet vytvoří i v průběhu lisování, kdy lis nasaje nečistotu a ta se dostane opět pod matrici. Princip je pak stejný. Na desce jí pak můžeme pozorovat jako malý důlek. Smeť se projevuje zvukem, co obsahuje nízké frekvence (bouchání).



Obr. 2.2: Ukázka spektrogramu a časového průběhu typické impulsní vady

3 Detekce vad

3.1 AR model

Metoda autoregresního modelu je založena na predikci následujícího vzorku signálu z předešlých vzorků jejich váhovým součtem. Při modelování nejčastěji uvažujeme aditivní model rušení, kde se chyby přičítají k původnímu zvukovému signálu. Takovýto model lze popsat rovnicí [4]

$$x_t = s_t + n_t, \quad (3.1)$$

kde x_t je výsledný signál tvořený součtem původního signálu s_t a rušení n_t .

Protože uvažujeme aditivní model a rušení je impulsního charakteru je třeba ještě složku n_t rozdělit na dvě komponenty podle rovnice [4]

$$n_t = i_t v_t, \quad (3.2)$$

kde i_t je složka udávající, kdy je vzorek poškozen, která nabývá hodnot 1 v případě poškození a 0 v opačném případě. Složka v_t pak je samotné impulsní rušení.

Uvažujme dále, že původní nepoškozený signál lze aproximovat lokálně stacionárním AR modelem podle rovnice [4]

$$s_t = -\sum_{k=1}^p a_k s_{t-k} + e_t, \quad (3.3)$$

kde p je řád AR modelu, a_k jsou koeficienty modelu a e_t excitační signál v podobě bílého šumu s variancí σ_e^2 .

Pomocí rovnic 3.1 a 3.3 lze dále x_t přepsat do tvaru [4]

$$x_t = -\sum_{k=1}^p a_k s_{t-k} + e_t + n_t, \quad (3.4)$$

$$x_t = -\sum_{k=1}^p a_k (x_{t-k} - n_{t-k}) + e_t + n_t, \quad (3.5)$$

Z rovnice 3.5 pak lze nakonec vyjádřit detekční signál d_t , který získáme součtem známého signálu a predikovaného signálu [4]

$$d_t = x_t + \sum_{k=1}^p a_k x_{t-k} = \sum_{k=1}^p a_k n_{t-k} + e_t + n_t, \quad (3.6)$$

Detekční signál je tedy složen z excitačního signálu e_t , impulsního rušení n_t a efektu jeho předchozích vzorků. Lokalizovat rušení pak lze pomocí vhodně zvolené prahu v detekčním signálu $|d_t|$. Excitační signál e_t je totiž bílý šum, který má oproti složce rušení n_t malou amplitudu a velké amplitudy detekčního signálu d_t proto odpovídají poškozeným vzorkům. Problém je přesná lokalizace v čase, protože složky impulsního rušení se promítnou i do následujících p -ti vzorků detekčního signálu dle řádu modelu [4].

Dalším problémem pak je odhad koeficientů AR modelu, který musíme provést z poškozeného signálu. Předpokládejme, že odhadnuté koeficienty \hat{a}_k jsou odlišné od správných parametrů a_k . Odhadnutý detekční signál pak lze napsat jako [4]

$$\hat{d}_t = x_t + \sum_{k=1}^p \hat{a}_k x_{t-k} = \sum_{k=1}^p a_k n_{t-k} + e_t + n_t + \sum_{k=1}^p (\hat{a}_k - a_k) x_{t-k}. \quad (3.7)$$

Do detekčního signálu přibyla v tomto případě další složka, která odpovídá chybě odhadu koeficientů AR modelu. V případě, že bude tato chyba malá, lze opět uvažovat, že složka rušení n_t bude mít největší amplitudu a můžeme provést detekci stejným způsobem.

Pro odhad koeficientů \hat{a}_k lze použít několika algoritmů. Nejběžnějším způsobem je řešení tzv. Yule-Walkerových rovnic, které vychází z odhadu autokorelační funkce pomocí rovnice [4]

$$\hat{R}(\tau) = \frac{1}{N} \sum_{k=\tau+1}^N x_k x_{k-\tau}, \quad (3.8)$$

kde N je počet vzorků signálu.

Odhad koeficientů pak vychází z následující soustavy rovnic [4]

$$\begin{pmatrix} \hat{R}(0) & \hat{R}(1) & \cdots & \hat{R}(p-1) \\ \hat{R}(1) & \hat{R}(0) & \cdots & \hat{R}(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{R}(p-1) & \hat{R}(p-2) & \cdots & \hat{R}(0) \end{pmatrix} \begin{pmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \vdots \\ \hat{a}_p \end{pmatrix} = \begin{pmatrix} \hat{R}(1) \\ \hat{R}(2) \\ \vdots \\ \hat{R}(p) \end{pmatrix}. \quad (3.9)$$

Matice soustavy je tzv. Toeplitzova matice a pro její řešení existuje rychlý Levinson-Durbinův algoritmus, který má časovou složitost $O(p^2)$ [4].

Výpočet detekčního signálu d_t lze pak v reálné aplikaci provést pomocí FIR filtru s impulsní odezvou [5]

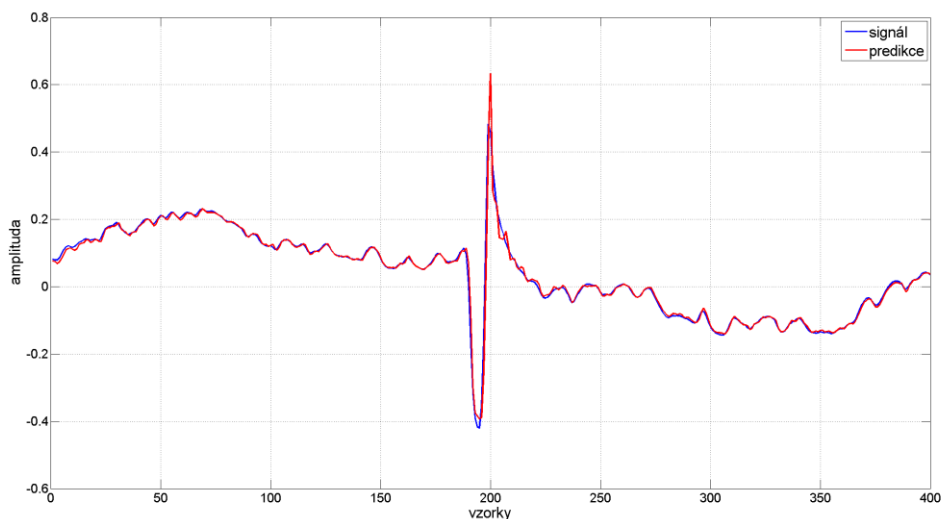
$$h = \{1, -\hat{a}_1, -\hat{a}_2, \dots, -\hat{a}_p\}. \quad (3.10)$$

Jednoduchý algoritmus založený na této metodě pak typicky vypadá následovně [4]:

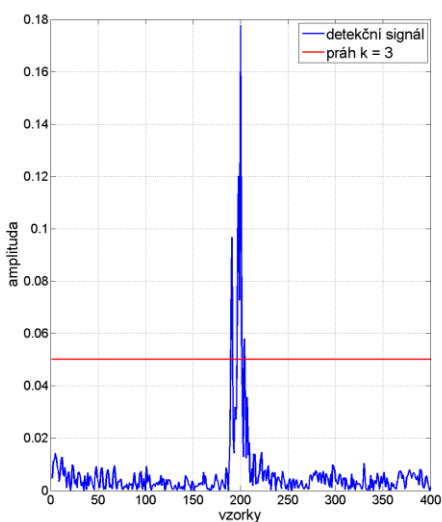
- Vybrat ze signálu segment, který je lokálně stacionární
- Odhadnout koeficienty modelu \hat{a}_k
- Vypočítat detekční signál d_t
- Odhadnout varianci excitačního signálu $\hat{\sigma}_e$
- Pokud $|d_t| > k\hat{\sigma}_e$, potom $i_t = 1$, jinak $i_t = 0$

Pro správnou detekci je důležité zvolit vhodnou hodnotu konstanty k , tak abychom detekovali větší nu chyb. Při předpokladu, že excitační signál e_t má Gaussovo rozložení, můžeme zvolit její hodnotu na 3, tedy standardní kritérium pro detekci neobvyklé hodnoty [6].

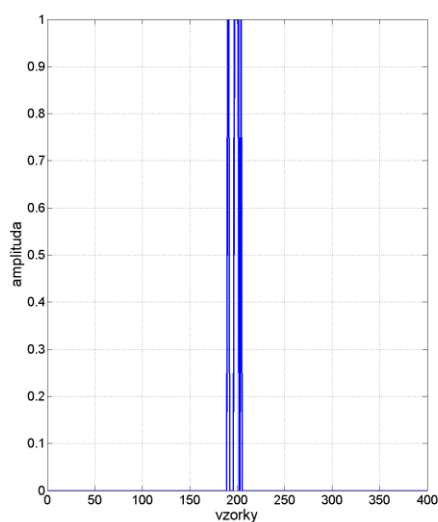
Ukázka takového postupu je zobrazena na následujících obrázcích. Na obr. 3.1 je zobrazen poškozený signál x_t a jeho predikce získána pomocí AR modelu a odhadnutých koeficientů \hat{a}_k . Na obr. 3.2 je dále zobrazen detekční signál d_t , který odpovídá rozdílu signálu a jeho predikovaným hodnotám. Dále je zde zobrazen práh, pomocí kterého lokalizujeme poškozenou část. Na obr. 3.3 je nakonec zobrazen výsledek tohoto procesu, kterým je detekční složka i_t , která pomocí dvou hodnot udává, které vzorky signálu obsahují rušení.



Obr. 3.1: Signál a jeho predikce pomocí AR modelu



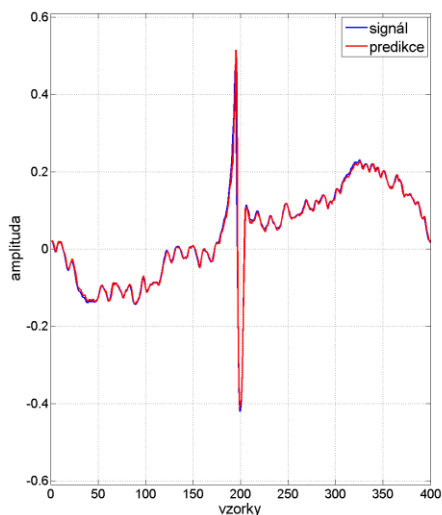
Obr. 3.2: Detekční signál a práh pro lokalizaci rušení



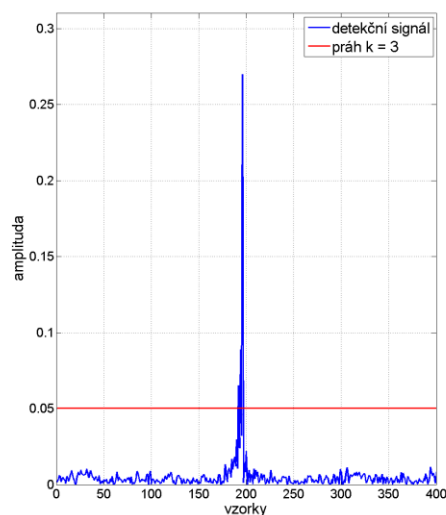
Obr. 3.3: Výsledný detekční vektor

Vylepšení metody

Zmíněnou metodu lze dále pomocí několika jednoduchých kroků vylepšit. Prvním vylepšením je provést stejný postup navíc se signálem v opačném smyslu, tedy obráceným [6]. Tento postup je zobrazen na obr. 3.4, kde je vykreslen stejný signál jako v předchozím případě a jeho predikce, ale ve zmíněném obráceném smyslu. Na obr. 3.5 je pak zobrazen detekční signál pro tento případ. Detekce z obou způsobů pak spojíme dohromady do jednoho detekčního vektoru. Tento krok může v některých případech pomoci detekovat rušení, které by jinak nebylo nalezeno, a také zlepšit časovou lokalizaci.



Obr. 3.4: Signál a jeho predikce
v obráceném smyslu



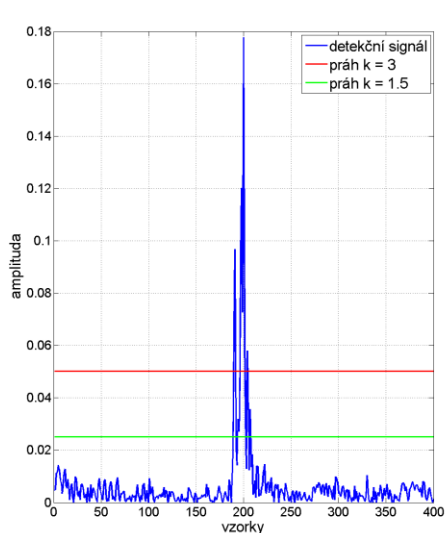
Obr. 3.5: Detekční signál pro případ
obráceného smyslu

Dalším vylepšením je úprava výsledného detekčního vektoru. Jak je vidět z obr. 3.2, detekční signál je v některých místech rušení pod prahem a v detekčním vektoru jsou následně v tomto místě střídavě dva stavy. To lze jednoduše v případě jednoho impulsu vyřešit tak, že najdeme první a poslední vzorek, pro který je amplituda nad prahem. Mezi těmito vzorky pak uvažujeme všechny vzorky jako poškozené. Problémem může být, když se vyskytne v signálu více rušivých impulsů těsně u sebe. To lze vyřešit přidáním parametru b , který udává maximální počet po sobě jdoucích vzorků, které jsou pod zvoleným prahem v detekčním signálu. V případě, že mezi dvěma vzorky, které jsou určeny jako poškozené, bude větší počet vzorků, než udává parametr, pod prahem, pak se budou tyto dvě oblasti detekovat jako dva impulsy rušení. V opačném případě se bude celá oblast detekovat jako jeden impuls. Detekce se tak rozšíří o další krok a je následovná [4]:

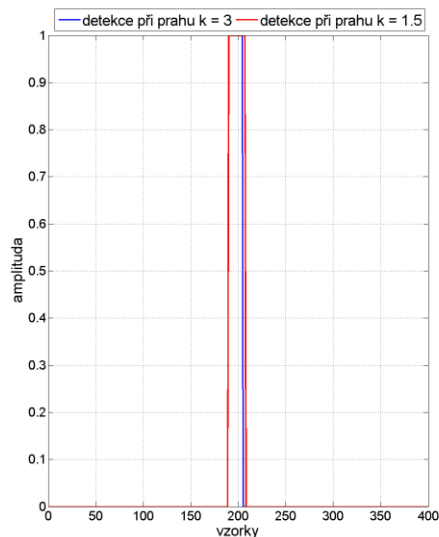
- Detekovat množinu vzorků $T \subset \{1, \dots, N\}$ tak, že $\forall t \in T, |d_t| > k\hat{\sigma}_e$
- Když bude platit $(t, \hat{t}) \in T^2$ a $|t - \hat{t}| \leq b$ tak přidáme vzorky mezi t a \hat{t} do množiny T

Dalším vylepšením je přidání dalšího prahu, který bude mít menší hodnotu a pomůže nám lépe určit časovou lokalizaci rušení. Princip je patrný z obr. 3.6, kde je opět zobrazen stejný detekční signál. Pomocí většího prahu nejprve detekujeme oblast rušení a poté pomocí menšího prahu určíme přesnější lokalizaci v čase tak, že kolem nalezené oblasti hledáme vzorky detekčního signálu, které mají amplitudu větší než je menší práh [7].

Výsledná detekce při použití popsaných vylepšení je zobrazena na obr. 3.7, kde je vykreslen detekční vektor pro metodu z jedním prahem i dvěma prahy. Přidáním druhého prahu se podařilo lépe určit oblast rušení v poškozeném signálu.



Obr. 3.6: Detekce pomocí dvou prahů



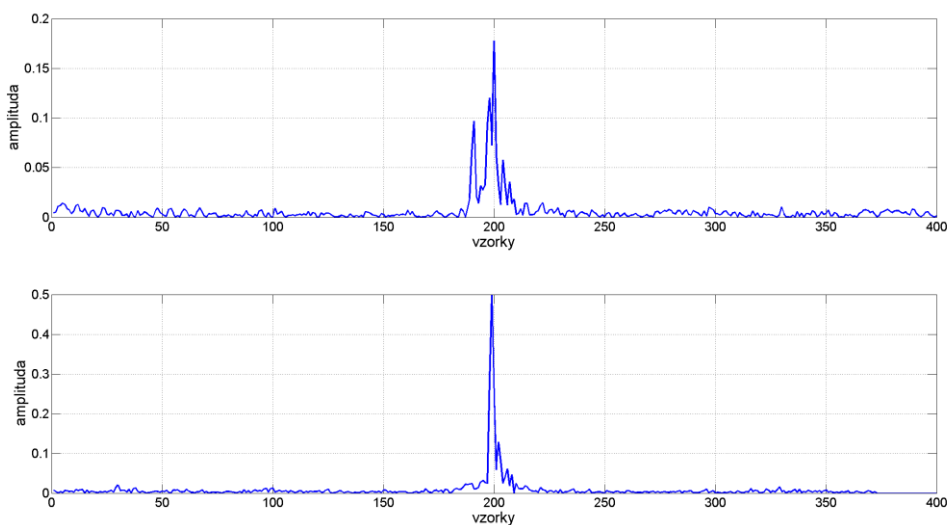
Obr. 3.7: Výsledná detekce po přidání vylepšujících kroků

Matched filtr

Další úpravou AR modelu je metoda přizpůsobeného filtru. Tou lze dosáhnout zvýraznění impulsního rušení v detekčním signálu d_t vzhledem k okolnímu signálu. Detekční signál je zde filtrován filtrem s obrácenou impulsní odezvou h , která byla použita u filtru prvního, pomocí kterého byl získán detekční signál d_t . Impulsní odezva přizpůsobeného filtru tedy nabývá hodnot [5]

$$h_m = \{-\hat{a}_p, -\hat{a}_{p-1}, -\hat{a}_{p-2}, \dots, 1\}. \quad (3.11)$$

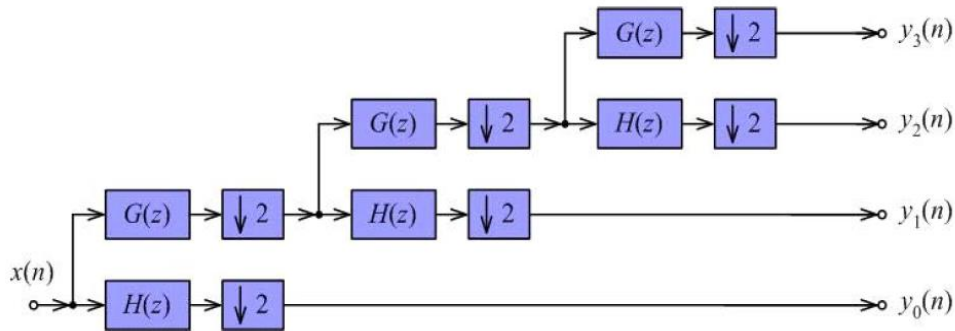
Porovnání základního AR modelu a přizpůsobeného filtru je zobrazeno na obr. 3.8. Jak je patrné došlo k zvýraznění rušení, ale zároveň byla zhoršena časová lokalizace.



Obr. 3.8: Porovnání základního AR modelu (nahore) a přizpůsobeného filtru (dole)

3.2 Vlnková transformace

Další metoda pro detekci vad využívá dyadickou vlnkovou transformaci s diskrétním časem [8]. Ta rozkládá signál pomocí zrcadlově kvadraturních filtrů na aproximační a detailní části. Struktura takového algoritmu je zobrazena na obr. 3.9. Vstupní signál je v první úrovni filtrován hornopropustným filtrem s přenosovou charakteristikou $H(z)$ a decimován. Výstupem pak je signál s detaily v této úrovni D_0 . Obdobně je signál filtrován i dolnopropustným filtrem s přenosovou charakteristikou $G(z)$. Zde je výstupem aproximační signál A_0 . Z tohoto signálu lze pak dále postupovat do větší hloubky a opakovat stejný proces filtrace [9].



Obr. 3.9: Dyadická dekompozice signálu. Převzato z [10]

Algoritmus pro detekci dále využívá koeficienty detailů z jednotlivých úrovní. Tyto koeficienty jsou porovnány s dynamickým prahem, který je dán rovnicí [8]

$$\tau(n, s) = k \cdot Env[|D_s(n)|], \quad (3.12)$$

kde D_s jsou koeficienty transformace v úrovni s , k je empiricky určená konstanta a $Env[\cdot]$ je obálka signálu, který není narušen vadami. Vyhodnocení je dále stejné jako v případě AR modelu tedy

- Pokud $|D_s| > \tau(n, s)$, potom $i_t = 1$, jinak $i_t = 0$

Jako operátor $Env[\cdot]$ lze pak použít mediánový filtr s vhodně nastavenou délkou okna. Práh je v tomto případě dán rovnicí [8]

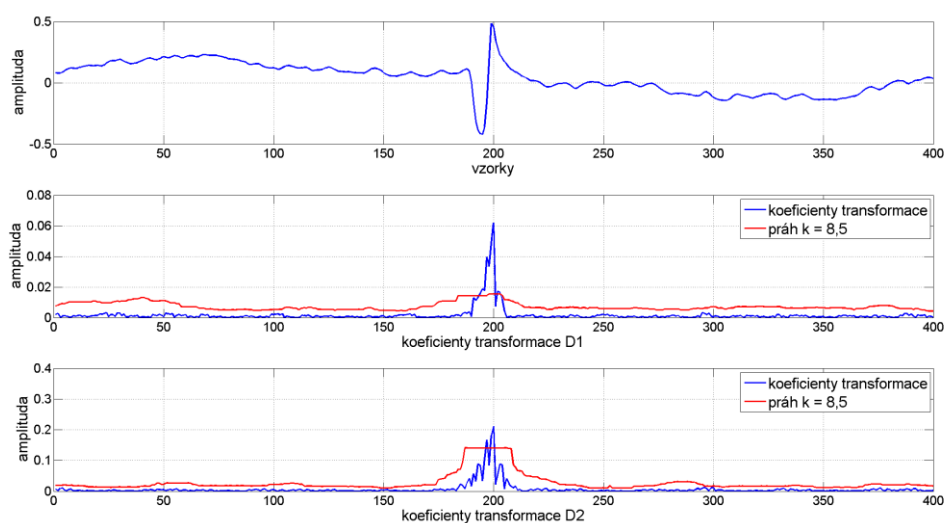
$$\tau(n, s) = k \cdot MED[|D_s(n - K)|, \dots, |D_s(n)|, \dots, |D_s(n + K)|], \quad (3.13)$$

kde K vychází z délky mediánového filtru, která by měla být pro správnou detekci dostatečně dlouhá v porovnání s délkou impulsního rušení. V případě, že maximální délka rušení v signálu je rovna hodnotě K_{MAX} , tak by měla délka filtru splňovat podmínku [8]

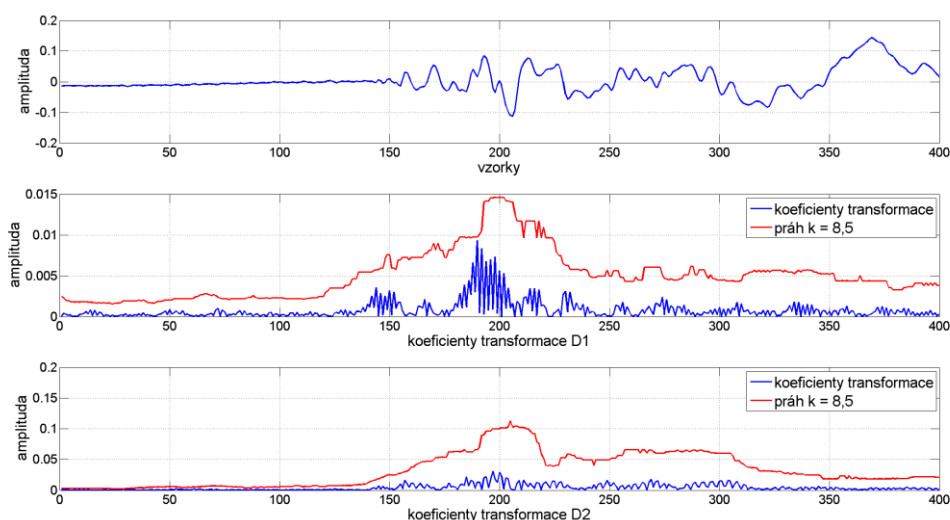
$$L_{MED} > 2K_{MAX} + 1 \quad (3.14)$$

Délka filtru by dále neměla být příliš dlouhá, protože předpokladem pro správnou detekci je, že délka impulsních vad je kratší než délka vysokofrekvenčních částí signálu.

Ukázka detekce pomocí této metody je zobrazena na obr. 3.10. Signál je rozložen do druhé úrovně a koeficienty detailů jsou prahovány dle popsaného postupu pomocí mediánového filtru. Na obr. 3.11 je pak zobrazen stejný postup pro signál obsahující vysokofrekvenční složky, které jsou delší než uvažované impulsní vady. Práh je v tomto případě nad koeficienty transformace a úsek je vyhodnocen správně jako bez vady.



Obr. 3.10: Detekce pomocí vlnkové transformace (nahore časový průběh signálu)



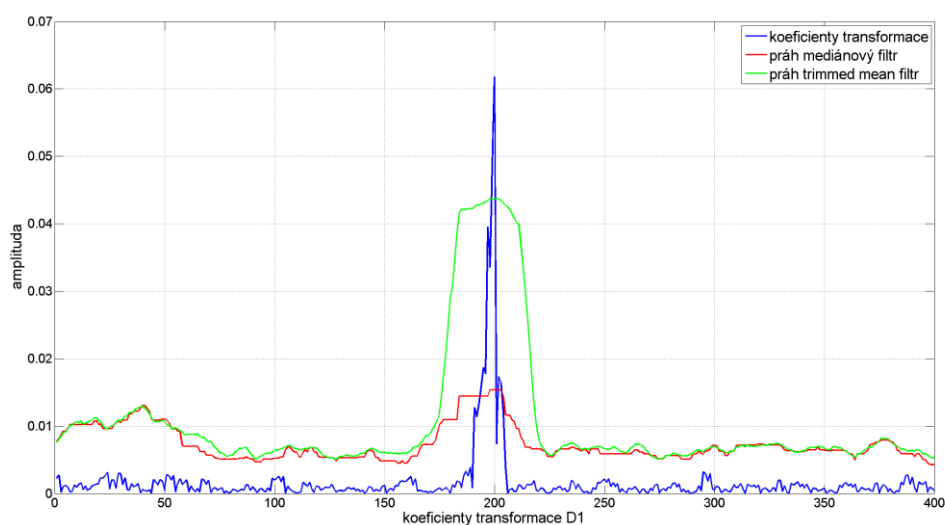
Obr. 3.11: Vyhodnocení v případě vysokofrekvenční části signálu (nahore časový průběh signálu)

Další možností implementace operátoru $Env[\cdot]$ je použití α -trimmed mean filtru [8], který nejdříve vyřadí nastavené procento největších a nejmenších hodnot a ze zbývajících vypočte průměrnou hodnotu. Počet hodnot, které jsou vyřazeny, je ovládán parametrem α , který nabývá hodnot v rozmezí $0 \leq \alpha \leq 0,5$. V případě hodnoty 0 přechází výpočet do klasického průměru a v případě hodnoty 0,5 je výsledek roven mediánu. Výpočet hodnoty α -trimmed mean je pak dán rovnicí [11]

$$X_{\alpha} = \frac{1}{L_{\alpha} - 2\lfloor \alpha L_{\alpha} \rfloor} \sum_{i=\lfloor \alpha L_{\alpha} \rfloor + 1}^{L_{\alpha} - \lfloor \alpha L_{\alpha} \rfloor} X_i, \quad (3.15)$$

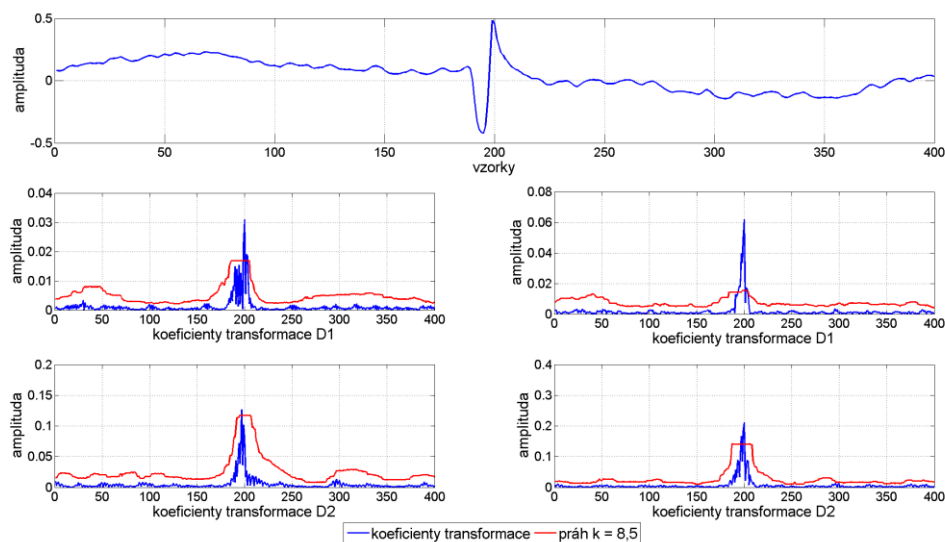
kde L_{α} je délka okna, $\lfloor \alpha L_{\alpha} \rfloor$ je funkce celá část a hodnoty X_i jsou seřazené, tak že platí $X_1 \leq X_2 \leq \dots \leq X_{L_{\alpha}}$. Pro správnou detekci je dále opět třeba zvolit vhodnou délku okna filtru a zde také parametr α .

Porovnání těchto dvou prahů je zobrazeno na obr. 3.12. Délka okna filtrů je stejná a parametr $\alpha = 0,25$.



Obr. 3.12: Porovnání výpočtu prahů pro detekci

Pro správnou detekci je dále nutné zvolit vhodný typ vlnky a její řád. Podle [8] jsou pro tuto úlohu nevhodnější vlnky Daubechiový. Řád vlnky pak závisí na parametrech impulsních vad. V případě, že je energie vady vzhledem k okolnímu signálu velká, je lepší zvolit menší řád a naopak. Dalším faktorem pak je i délka vad, kde se zvyšující se délkou je lepší volit zároveň vyšší řád [8]. Porovnání detekce při použití dvou různých řádů Daubechiový vlnky, které byly použity v [8, 10], je zobrazeno na obr. 3.13. Lze pozorovat, že v případě vlnky řádu 4 jsou koeficienty transformace v místě vady větší než v případě vlnky řádu 10.



Obr. 3.13: Porovnání dvou řádů Daubechiový vlnky – 10(vlevo) a 4(vpravo)

3.3 Derivace

Další metoda využívá charakteristiky impulsního rušení, kterou je rychlá změna. Detekční signál je v tomto případě vytvořen derivací signálu, kterou lze pro případ diskrétního signálu vypočítat dle rovnice [12]

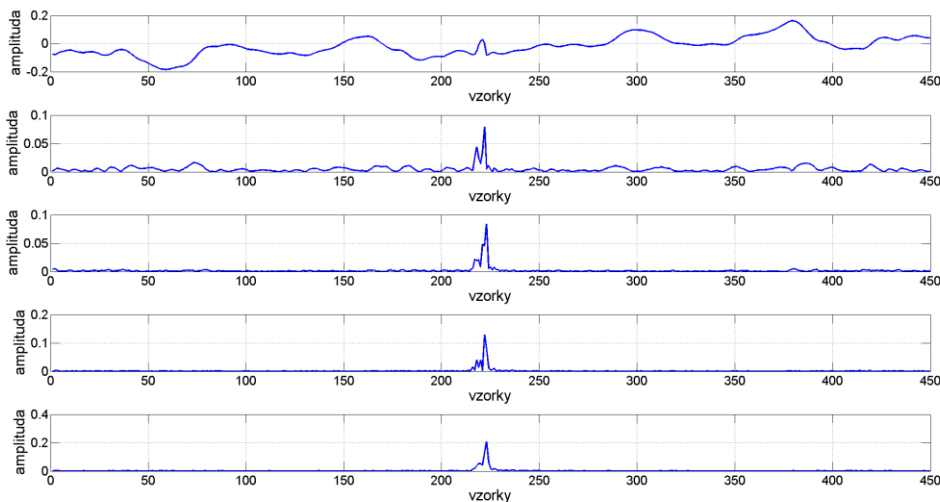
$$d_n = \frac{x_{n+1} - x_n}{\Delta t}, \quad (3.16)$$

kde Δt je vzorkovací krok. Pro zlepšení detekce pak lze aplikovat derivaci vícekrát. Podle [12] je pro detekci optimální 4. derivace signálu, kterou lze vypočítat dle rovnice [12]

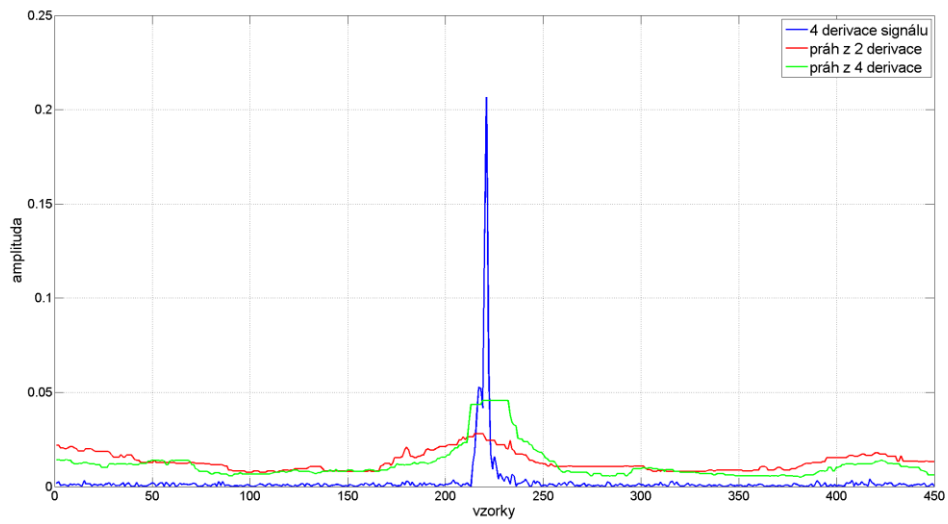
$$d_n = \frac{x_{n-2} - 4x_{n-1} + 6x_n - 4x_{n+1} + x_{n+2}}{(\Delta t)^4}. \quad (3.17)$$

Ilustrace absolutní hodnoty 1-4 derivace signálu poškozeného impulsním rušením je znázorněna na obr. 3.14. Lze pozorovat, že z větším stupněm derivace se zvýrazňuje místo poruchy vzhledem k okolí.

Detekce dále probíhá stejně jako v případě vlnkové transformace. Absolutní hodnoty derivovaného signálu se porovnají s prahem získaným filtrací tohoto signálu mediánovým nebo α -trimmed mean filtrem (viz rovnice 3.12). Podle [12] je pak možné detekci dále vylepšit výpočtem prahu ze signálu s nižším stupněm derivace. V případě, že za detekční signál budeme uvažovat 4. derivaci signálu, tak je práh počítán ze signálu 2. derivace. Porovnání takto získaných prahů pomocí mediánového filtru je zobrazeno na obr. 3.15.



Obr. 3.14: Signál s impulsním rušením a jeho derivace (vyšší stupeň derivace směrem dolů)

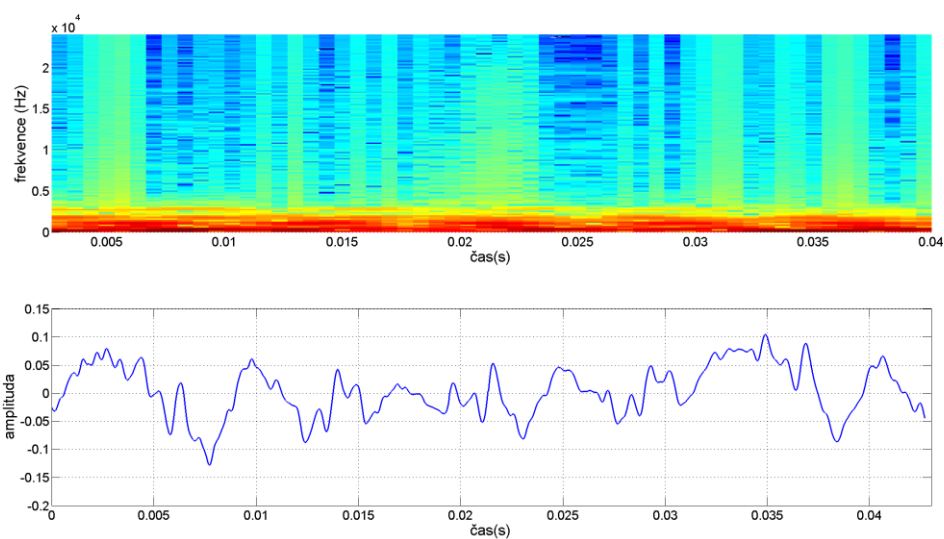


Obr. 3.15: Porovnání prahů získaných z 2 a 4 derivace

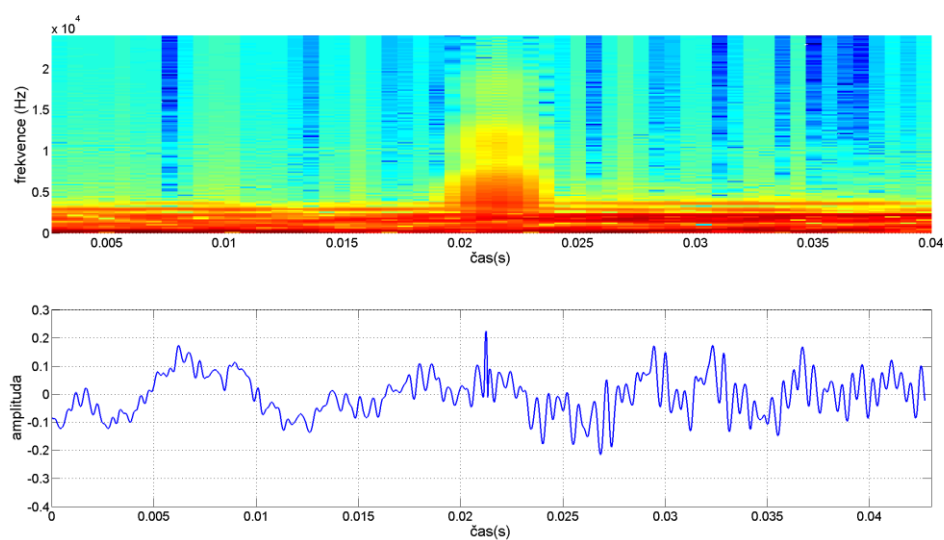
4 Ověření detekce vad

4.1 Krátkodobá Fourierova transformace

Při detekci často vznikají falešné detekce a to zejména při nastaveném nízkém prahu, který je potřeba pro hledání menších chyb. Na obr. 4.1 je příklad takové falešné detekce. Jak je patrné ze spektrogramu, tak se o vadu nejedná. Na obr. 4.2 je pak naopak uveden příklad, jak se typicky projevuje ve spektru impulsní vada. Toho je využito při ověření následujícím způsobem.



Obr. 4.1: Ukázka spektrogramu a časového průběhu falešné detekce



Obr. 4.2: Ukázka spektrogramu a časového průběhu správné detekce

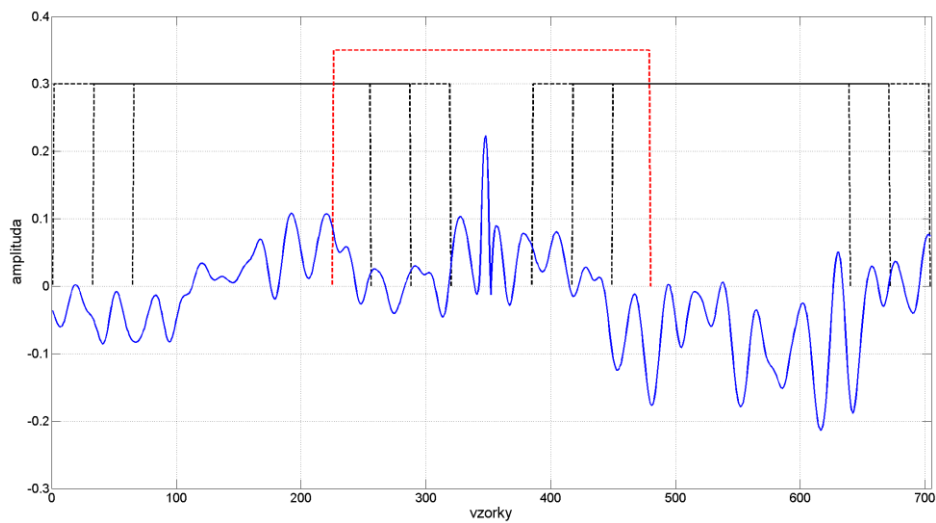
Nejdříve je vybráno po 3 segmentech okolo vady a jeden segment v místě vady jak ilustruje obr. 4.3. Z těchto segmentů jsou dále vypočtena 2 průměrná spektra ze segmentů kolem vady. Dále se tedy počítá se třemi spektry, jejichž grafické znázornění je zobrazeno na obr. 4.4. Je vidět, že v místě vady je spektrum zejména na vyšších frekvencích odlišné a naopak okolní segmenty jsou si na těchto frekvencích podobné. Ověření je pak provedeno porovnáním těchto 3 segmentů na vyšších frekvencích navzájem pomocí upraveného výpočtu spektrální vzdálenosti daného rovnicí [13]

$$C = 4.34 \sqrt{2 \sum_{k=h}^N (c_{k1} - c_{k2})^2}, \quad (4.1)$$

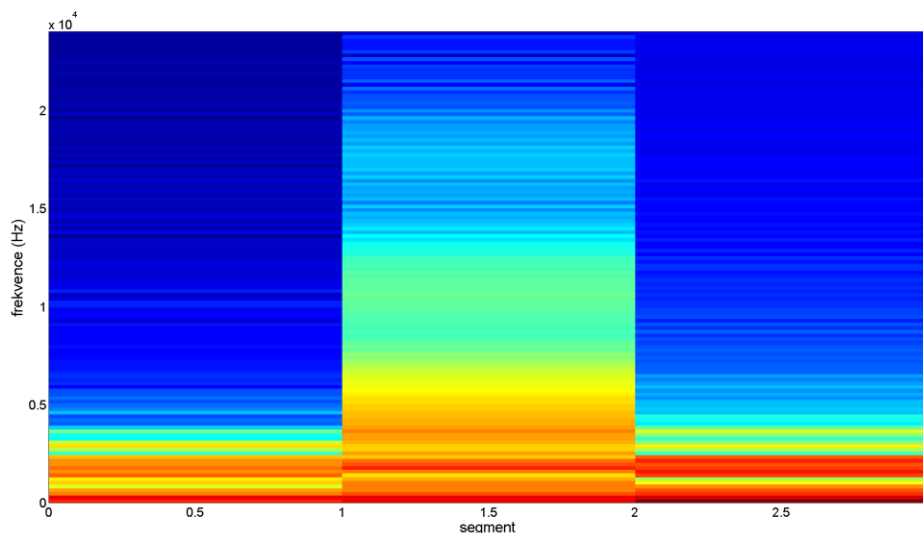
kde c_{k1} jsou koeficienty Fourierovy transformace jednoho segmentu, c_{k2} jsou koeficienty druhého segmentu a h je index odpovídající zvolené hodnotě frekvence, od které se výpočet realizuje. Výsledkem pak jsou tři hodnoty C_{12} , C_{13} a C_{23} , které odpovídají rozdílům segmentů mezi sebou. Rozhodnutí o detekci je pak provedeno dle rovnice

$$C_{12} + C_{23} - C_{13} < TH, \quad (4.2)$$

kde TH je vhodně nastavený práh. V případě platnosti této podmínky se detekce vyhodnotí jako falešná a je vyřazena ze seznamu. Podmínka odpovídá tomu, že impulsní vada naruší v krátkém úseku signál. Rozdíl mezi segmentem s vadou a segmenty okolními je pak velký a rozdíl mezi dvěma okolními segmenty je malý.



Obr. 4.3: Segmentace v okolí nalezené vady



Obr. 4.4: Grafické znázornění získaného spektrogramu

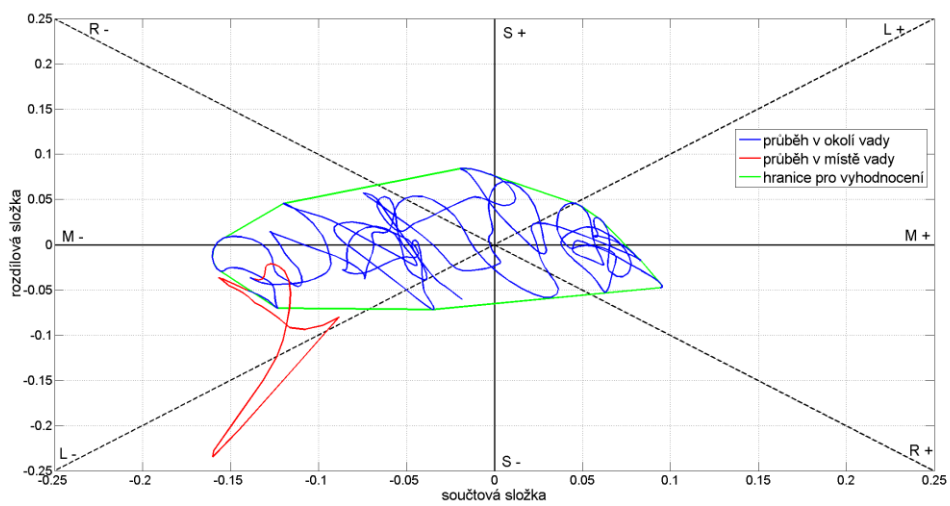
4.2 Průběh součtové a rozdílové složky

Po konzultaci ve firmě GZ Media, a.s pak bylo zjištěno, že další možností ověření je využití rozdílové a součtové složky signálu, které získáme z pravého a levého kanálu podle rovnic

$$M = (L + R) / 2 \quad (4.3)$$

$$S = (L - R) / 2 \quad (4.4)$$

Když pak vyneseme na horizontální osu grafu součtovou složku a na vertikální rozdílovou složku, bude se signál pohybovat v určitém prostoru. Ověření pak lze provést tak, že vezmeme určitý počet vzorků před a po nalezeném místě vady, a pomocí těchto vzorků si definujeme prostor, kde se signál pohybuje. Vzorky z místa vady pak porovnáme s tímto prostorem. V případě, že je v daném místě vada, její vzorky se obvykle pohybují mimo definovanou oblast a lze tak určit, že se jedná opravdu o vadu. Znázornění tohoto postupu je zobrazeno na obr. 4.5. Pohyb v tomto prostoru souvisí s fyzikálním principem záznamu a pohybem snímacího hrotu. Vady jsou pak způsobeny důlkem nebo překážkou ve drážce. Podle toho, jestli je tato překážka přítomna v drážce pro levý nebo pravý kanál, tak se pohyb vychýlí v tomto prostoru do odpovídajícího směru. V případě, že je vada v obou kanálech, je pohyb ve většině případů vertikální. Vady tedy způsobí zejména velkou výchylku v rozdílové složce, kterou je obvyklé při masteringu eliminovat kvůli technickým parametrům výroby. Toho lze dále využít u ověření a sledovat jestli se pohyb v místě detekované vady nepohybuje v oblasti vymezené úhly -15° až 15° a 165° až -165° , když by se hodnoty převedly do polárních souřadnic. Pohyb v těchto místech obvykle značí falešnou detekci, protože vady se vyznačující už zmíněnou velkou rozdílovou složkou.



Obr. 4.5: Ověření pomocí součtové a rozdílové složky

5 Implementace

Z metod popsaných v předchozích kapitolách byly vytvořeny detekční algoritmy implementované v prostředí MATLAB verze R2013b. U všech algoritmů je použita stejná struktura lišící se jenom funkcí odpovídající použitému typu detekce. Celkem byly vytvořeny 4 skripty s metodami AR modelu, AR modelu + matched filtru, vlnkové transformace a derivace. Dále bude popsána struktura algoritmů.

Signál je zpracováván po segmentech. Parametry segmentace jsou dány proměnnými:

- *delkaseg* odpovídající délce segmentu
- *krok* odpovídající délce posunu okna

Pro každý segment je dále provedena detekce zvlášť pro pravý a levý kanál. K tomu slouží funkce *AR_segment*, *matched_segment*, *vlnkova_segment* a *derivace_segment*. Struktura těchto funkcí je dále popsána v jednotlivých kapitolách

5.1 AR model

Detekci pomocí AR modelu s dvěma prahy provádí funkce *AR_segment* s parametry:

$[\text{detek_seg_pravy}, \text{detek_seg_levy}] = \text{AR_segment}(\text{segment_sig}, \text{rad_AR}, \text{prah_chyb}, \text{prah_chyb2})$

- *segment_sig* je segment signálu obsahující pravý a levý kanál
- *rad_AR* je hodnota řádu AR modelu
- *prah_chyb* je hodnota konstanty k pro větší práh
- *prah_chyb2* je hodnota konstanty k pro menší práh
- *detek_seg_pravy* je detekční vektor pro pravý kanál signálu
- *detek_seg_levy* je detekční vektor pro levý kanál signálu

Algoritmus detekce byl popsán v kapitole 3.1. Pro odhad koeficientů \hat{a}_k a variance $\hat{\sigma}_e$ je využito Matlabovské funkce *lpc*, která počítá koeficienty podle zmíněných Yule-Walkerových rovnic (viz rovnice 3.9). Pro predikci signálu x_t je pak využito Matlabovské funkce *filter*, která realizuje predikci pomocí FIR filtru. Detekční signál d_t se následně získá odečtením signálu a predikce (viz rovnice 3.6). Vzorky na začátku tohoto signálu jsou v délce řádu AR modelu pak kvůli přechodným jevům vynulovány. Další krok je detekce rušení podle prahu definovaném hodnotou *prah_chyb* (viz obr. 3.2). Tím se získá umístění poškozených vzorků. Dále je provedeno vyhodnocení podle druhého prahu definovaném hodnotou *prah_chyb2*. Tím se získá přesnější lokalizace v čase. Obě tyto detekce jsou provedeny porovnáním amplitudy vzorků a prahu, který vznikl vynásobením hodnoty variance $\hat{\sigma}_e$ a příslušné hodnoty *prah_chyb* respektive *prah_chyb2*. Hledání vzorků v případě druhého prahu je pak omezeno oblastí, která byla detekována větším prahem. Konkrétně je hledán sled vzorků na obě strany od vzorků vymezených prvním prahem, jejichž amplituda je větší než hodnota druhého prahu. Popsaný postup je aplikován odděleně na oba kanály signálu a výstupem funkce jsou nakonec dva detekční vektory obsahující složku i_t (viz obr. 3.7).

Detekci pomocí AR modelu + matched filtru provádí funkce *matched_segment* s parametry:

$[\text{detek_seg_pravy}, \text{detek_seg_levy}] = \text{matched_segment}(\text{segment_sig}, \text{rad_AR}, \text{prah_chyb}, \text{prah_chyb2})$

- *segment_sig* je segment signálu obsahující pravý a levý kanál

- *rad_AR* je hodnota řádu AR modelu
- *prah_chyb* je hodnota konstanty *k* pro větší práh
- *prah_chyb2* je hodnota konstanty *k* pro menší práh
- *detek_seg_pravy* je detekční vektor pro pravý kanál signálu
- *detek_seg_levy* je detekční vektor pro levý kanál signálu

Postup je zde stejný jako v případě funkce *AR_segment*, ale přibyl zde jeden krok v podobě filtrace detekčního signálu filtrem s obrácenou impulsní odezvou (viz rovnice 3.11). Filtrace je opět realizována funkcí *filter*. U detekčního signálu je dále ještě kompenzován časový posun způsobený filtrací. Výstupem jsou pak opět detekční vektory i_t .

V případě AR modelu je stejnou funkcí následně provedena detekce i pro signál opačného smyslu. Výsledné vektory pro oba směry jsou pak sečteny a vznikne celková detekce.

5.2 Vlnková transformace

Detekci pomocí vlnkové transformace provádí funkce *vlnkova_segment* s parametry:

$$[\text{detek_seg_pravy}, \text{detek_seg_levy}] = \text{vlnkova_segment}(\text{segment_sig}, \text{vlnka}, \text{prah_fil}, \text{delka_fil}, \text{typ_fil}, \text{alfa_p})$$

- *segment_sig* je segment signálu obsahující pravý a levý kanál
- *vlnka* je typ vlnky k analýze
- *prah_fil* je hodnota konstanty *k*, kterou se násobí práh
- *delka_fil* je hodnota délky okna filtru pro výpočet prahu
- *typ_fil* je hodnota určující typ filtru – 1 je mediánový a 2 trimmean
- *alfa_p* je hodnota parametru α pro trimmean filtr
- *detek_seg_pravy* je detekční vektor pro pravý kanál signálu
- *detek_seg_levy* je detekční vektor pro levý kanál signálu

Algoritmus detekce je popsán v kapitole 3.2. Nejdříve je provedena dyadická dekompozice do druhého stupně pomocí Matlabovské funkce *wavedec*. Ta realizuje dekompozici pomocí dolnoproústného a hornoproústného filtru, jejichž přenosové charakteristiky jsou dané typem zvolené vlnky (viz obr. 3.9). Dále jsou použity jen absolutní hodnoty koeficientů detailů. Z těch se vypočte práh τ buď pomocí mediánového filtru nebo trimmean filtru (viz rovnice 3.12). K výpočtu prahu pomocí mediánového filtru je použita Matlabovská funkce *medfilt1*, která rovnou vypočítá hodnoty prahu dle zvolené délky okna. Pro výpočet trimmean filtru pak byla vytvořena funkce *trimmeanfil*, která postupně počítá hodnoty prahu pomocí for cyklu. K výpočtu hodnoty trimmean je pak využito Matlabovské funkce *trimmean*. Hodnoty prahu jsou nakonec vynásobeny hodnotou *prah_fil* a jsou porovnány s koeficienty transformace. Hodnota konstanty je stejná pro obě úrovně. Vzhledem k délce okna filtrů při počítání prahů jsou následně ještě vynulovány hodnoty v této délce na začátku a konci detekčních signálů. Detekce je takto provedena v obou úrovních dekompozice a výsledná detekce vznikne vynásobením detekčních signálů z první a druhé úrovně. Vada je tedy detekována jen v případě, že je nalezena v obou úrovních. Postup je opět aplikován na oba kanály a výstupem jsou pak detekční vektory i_t .

5.3 Derivace

Detekci pomocí derivace provádí funkce *derivace_segment* s parametry:

$$[\text{detek_seg_pravy}, \text{detek_seg_levy}] = \text{derivace_segment}(\text{segment_sig}, \text{prah_fil}, \text{delka_fil}, \text{typ_fil}, \text{alfa_p})$$

- *segment_sig* je segment signálu obsahující pravý a levý kanál
- *prah_fil* je hodnota konstanty k , kterou se násobí práh
- *delka_fil* je hodnota délky filtru pro výpočet prahu
- *typ_fil* je hodnota určující typ filtru – 1 je mediánový a 2 trimmean
- *alfa_p* je hodnota parametru α pro trimmean filtr
- *detek_seg_pravy* je detekční vektor pro pravý kanál signálu
- *detek_seg_levy* je detekční vektor pro levý kanál signálu

Algoritmus detekce je popsán v kapitole 3.3. Nejdříve je vypočtena druhá a čtvrtá derivace signálu pomocí Matlabovské funkce *diff* a dále jsou použity jejich absolutní hodnoty. Prahy pro detekci jsou vypočteny z druhé derivace stejně jako v případě vlnkové transformace pomocí funkcí *medfilt1* nebo *trimmeanfil* podle zvoleného typu filtru. Hodnoty prahu jsou nakonec vynásobeny hodnotou *prah_fil* a jsou porovnány s hodnotami čtvrté derivace. Nakonec jsou ještě vynulovány hodnoty na začátku a konci detekčního signálu v délce okna filtru. Výstupem jsou pak opět detekční vektory i_t .

5.4 Další zpracování a ověření detekce

Detekce pomocí popsaných funkcí se pak postupně aplikuje na všechny segmenty signálu a tím získáme informace o všech nalezených vadách. Výstupní detekční vektory z funkcí jsou pak spojeny do jednoho a jsou uloženy v jedné matici, kde je v prvním řádku detekce pro levý kanál a v druhém řádku pro pravý kanál. S touto maticí se pak dále pracuje.

V dalším kroku po detekci se provede spojení chyb podle kritéria popsaného v kapitole 3.1. Nesouvislé detekce se objevují zejména u AR modelu, ale někdy i u ostatních metod, proto je spojení provedeno u všech typů detekce. K tomu slouží funkce *spojeni_chyb*, která má následující parametry:

$$[\text{det_sig}] = \text{spojeni_chyb}(\text{det_sig}, \text{vzdalenost_b})$$

- *det_sig* je spojený detekční vektor pro levý a pravý kanál získaný z funkcí pro detekci
- *vzdalenost_b* je hodnota parametru b
- *det_sig* na výstupu funkce jsou pak upravené detekční vektory

Funkce nejdříve pomocí Matlabovské funkce *find* najde v detekčním vektoru vzorky rušení a poté je postupně prochází a počítá vzdálenost mezi sousedními vzorky. Pokud je vzdálenost mezi vzorky menší než zadaný parametr b definovaný hodnotou *vzdalenost_b*, tak přidá vzorky mezi těmito dvěma do detekčního vektoru, tak jak je popsáno v kapitole 3.1.

Dalším krokem je nalezení indexů, které odpovídají nalezeným vzorkům obsahující rušení. To je provedeno funkcí *souradnice_chyb*, která má parametry:

$$[\text{chyby_s}] = \text{souradnice_chyb}(\text{det_sig})$$

- *det_sig* je zde detekční vektor pro jeden kanál

- *chyby_s* je struktura o dvou položkách *.zacatek* a *.konec*, kde je uložen údaj o začátku respektive konci chyby či chyb.

K nalezení údajů je nejdříve využito Matlabovské funkce *findpeaks*, která najde indexy začátků chyb, které odpovídají přechodu z 0 do 1. K nalezeným začátkům je pak nalezen konec chyb a to hledáním konce spojitého úseku 1, který začíná na nalezeném začátku. V případě více chyb v jednom segmentu jsou údaje přidávány postupně za sebe.

Souřadnice se takto najdou pro oba kanály a výsledky jsou uloženy ve dvou strukturách *chyby_vys_levy* a *chyby_vys_pravy*, které obsahují informace o začátcích a koncích nalezených chyb.

Po detekční fázi je posledním krokem algoritmu nakonec ověření chyb. Využívá se zde analýzy ve spektrální oblasti a dále vlastností v časové oblasti, které byly popsány v kapitole 4. Ověření probíhá zvláště pro vady nalezené v pravém a levém kanálu. K tomu slouží funkce *overeni_chyb* s parametry:

[*detekce_chyb*, *vymazat*] = *overeni_chyb*(*s_sig*, *chyby_s*, *detekce_chyb*, *kanal*, *prah_s*, *delka_oko*, *metoda_over*)

- *s_sig* je zde signál obsahující levý i pravý kanál
- *chyby_s* je struktura s indexy chyb kanálu, který ověřujeme
- *detekce_chyb* je detekční vektor kanálu, který ověřujeme
- *kanal* je parametr pro určení ověřovaného kanálů, který má hodnotu 1 pro levý a jinou pro pravý kanál
- *prah_s* je hodnota prahu *TH* pro ověření pomocí spektra
- *delka_oko* je hodnota délky okolí použitá u ověření v časové oblasti
- *metoda_over* určuje metodu pro ověření – 1 je metoda spektra, 2 je časový průběh a 3 jsou obě metody
- *detekce_chyb* na výstupu funkce je upravený detekční vektor, v kterém jsou odstraněny případné falešné detekce
- *vymazat* je pak vektor s indexy, které se mají vymazat ze struktury *chyby_s*

Ověření v časové oblasti je provedeno pomocí součtové a rozdílové složky, které se vypočtou podle rovnice 4.3 respektive 4.4. Délka oblasti, se kterou je počítáno, je dána parametrem *delka_oko*, který udává počet vzorků před a po chybě, podle kterých se určí hranice pro ověření. Zároveň je oblast ještě rozšířena o 5 vzorků na obě strany, o které je rozšířena oblast detekované vady. Po výpočtu součtové a rozdílové složky se vezmou jen vzorky z oblasti mimo vadu a z těch se vypočte hranice, která obklopuje tyto vzorky (viz obr. 4.5). Pro výpočet je využito Matlabovské funkce *convhull*, která vrací body, které toto ohraničení tvoří. Poté jsou použity vzorky, které odpovídají oblasti nalezené vady a vyhodnotí se, jestli leží mimo oblast vypočtenou v předchozím kroku. K tomu je využito Matlabovské funkce *inpolygon*, která vrací vzorky, které jsou uvnitř zvolené oblasti. Vyhodnocení jestli je nalezená vada jen falešná detekce, je pak dáno tak, že v případě kdy jsou všechny vzorky uvnitř oblasti, je tento případ vyhodnocen jako falešná detekce. Současně je také provedeno vyhodnocení, jestli se vada nepohybuje v oblasti mezi úhly -15° až 15° a 165° až -165° . To je provedeno převedením hodnot do polárních souřadnic pomocí Matlabovské funkce *cart2pol*. Jako falešná detekce je pak oblast vyhodnocena, když se všechny vzorky, které odpovídají nalezené vadě, pohybují v oblasti vymezené těmito úhly.

Ověření pomocí spektra pak vychází z porovnání místa vady a okolí pomocí upravené spektrální vzdálenosti. Nejdříve je vypočtena krátkodobá Fourierova transformace z úseku signálu dlouhém 576 vzorků, kde středem tohoto úseku je střed detekované vady. K výpočtu je využito Matlabovské funkce *spectrogram*.

Parametry výpočtu pak jsou následující:

- délka okna 256 vzorků
- použité okno Hammingovo
- krok 32 vzorků
- počet vzorků spektra stejný jako délka okna

Výsledkem je 11 segmentů a hodnoty jejich spekter, z nichž je pro další výpočet použita jen absolutní hodnota. V dalším kroku se ze tří krajních segmentů na obou stranách vypočtou dvě průměrná spektra, které představují oblast okolo místa detekované vady (viz obr. 4.4). Spektrum místa vady je pak dáno segmentem číslo 6. Výpočtem tedy vzniknou tři spektra, které nesou informaci o místě vady a jejím okolí. V dalším kroku pak jsou podle rovnice 4.1 vypočteny tři hodnoty spektrální vzdálenosti navzájem mezi segmenty. Hodnota indexu h , od které se výpočet realizuje, je 35 a odpovídá přibližně 6 kHz při uvažované vzorkovací frekvenci 44100 Hz. Vyhodnocení je pak dáno porovnáním hodnoty $prah_s$ a součtu vypočtených vzdáleností, jak je dáno rovnicí 4.2.

Celkové vyhodnocení je pak určeno podle toho, jestli je zvoleno ověření jen jednou určenou metodou nebo se berou v úvahu obě metody. V tom případě musí být splněny obě kritéria. To je dáno parametrem *metoda_over*.

Posledním krokem pak je vymazání vad, které byly vyhodnoceny jako falešné detekce, ze struktur *chyby_vys_levy* a *chyby_vys_pravy*, které po tomto kroku obsahují výslednou detekci pro oba kanály.

6 Testování a výsledky

Implementované metody byly testovány na 3 signálech poskytnutých firmou GZ Media, a.s. Signály jsou záznamem celých stran vinylových desek v délce 20 minut a v jednom případě 15 minut. Spolu se signály byly poskytnuty výsledky jejich detekčního softwaru založeném na porovnání záznamu z vyrobené vinylové desky a masteru, z kterého byla vyrobena. K výsledkům tohoto softwaru byly porovnány implementované metody pomocí dvou kritérií a to procenta nedetekovaných vad, které se vypočte dle rovnice

$$\% \text{ nedetekovaných vad} = \left(\frac{\text{nedetekované_vady}}{\text{vady_signálu}} \right) \cdot 100, \quad (6.1)$$

kde *vady_signálu* je počet vad, které byly nalezeny pomocí softwaru firmy GZ Media, a.s a *nedetekované_vady* je počet vad, které nebyly nalezeny testovanou metodou. Druhým kritériem pak je procento falešné detekce vypočtené dle rovnice

$$\% \text{ falešné detekce} = \left(\frac{\text{špatně_detekované_vady}}{\text{nalezené_vady}} \right) \cdot 100, \quad (6.2)$$

kde *nalezené_vady* je celkový počet nalezených vad pomocí testované metody a *špatně_detekované_vady* je počet vad, které nejsou v seznamu ze softwaru firmy GZ Media, a.s. Výsledky pro jednotlivé metody jsou dále popsány v následujících kapitolách. Nastavení parametrů je pak spojeno se vzorkovací frekvencí signálů, která činí 44100 Hz. Zároveň byly pomocí energetického kritéria vyřazeny šumové segmenty mezi jednotlivými nahrávkami, které jsou na deskách, protože v těchto místech vznikal velký počet falešných detekcí.

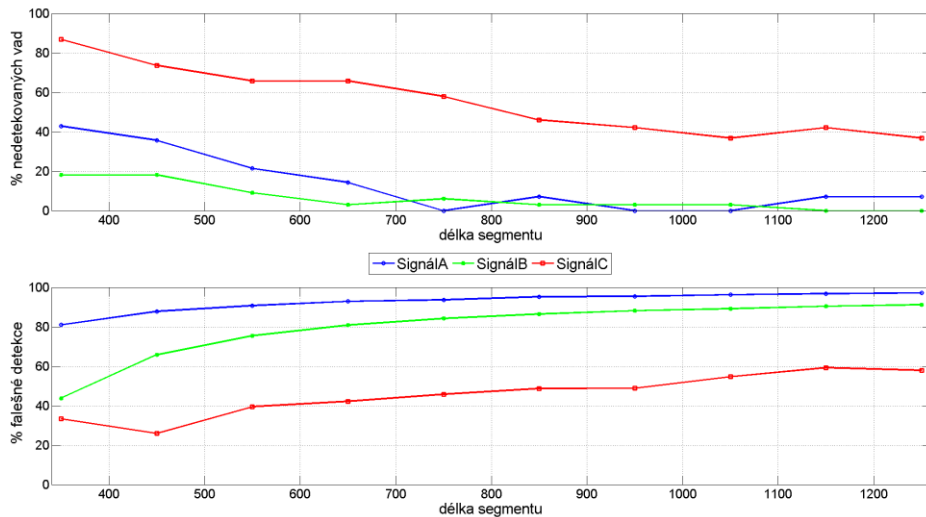
6.1 Parametry AR modelu

Pro nalezení optimálních parametrů bylo provedeno několik průběhů algoritmu při různém nastavení parametrů. Počáteční hodnoty parametrů vychází z hodnot, které byly používány v [4, 6] a počáteční práh byl nastaven na hodnotu, kdy nejsou nalezeny všechny chyby. Hodnota druhého prahu, která slouží jen k lepšímu určení v čase, pak byla nastavena pevně. Při testování pak byla měněna vždy jen hodnota jednoho parametru a ostatní zůstaly v základních hodnotách, které byly následující:

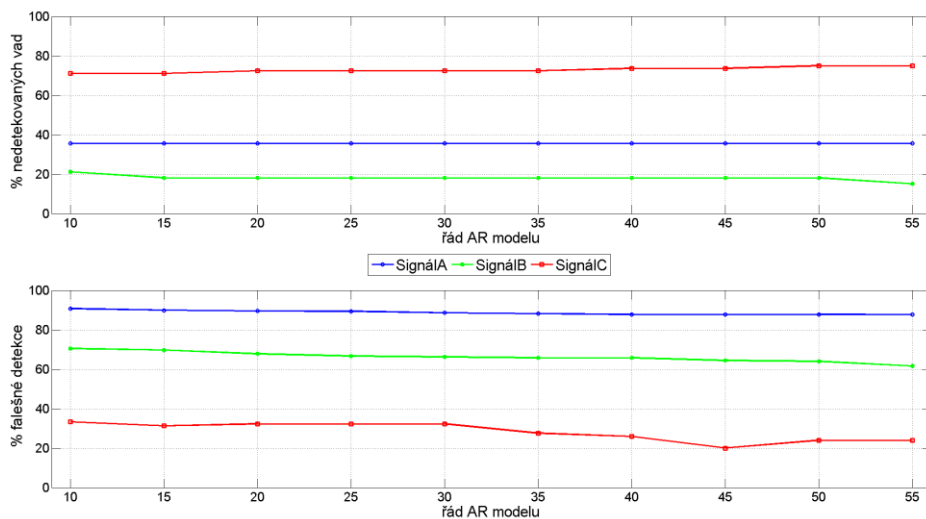
- *delkaseg* = 450
- *krok* = 0,8*delkaseg*
- *rad_AR* = 40
- *prah_chyb* = 12
- *prah_chyb2* = 3

Prvním zkoumaným parametrem je *delkaseg*, jehož výsledky jsou zobrazeny na obr. 6.1. Jak je vidět s rostoucí délkou segmentů dochází ke zmenšování počtu nedetekovaných vad, ale zároveň i k větší falešné detekci. To je zjevně způsobeno zvětšujícím se nedodržením podmínky stacionarity segmentů.

Dalším parametrem je *rad_AR*, jehož výsledky jsou zobrazeny na obr. 6.2. Je patrné, že hodnota tohoto parametru výsledky detekce příliš neovlivňuje. Jediným znatelným znakem je, že při příliš malém řádu dochází k větší falešné detekci.



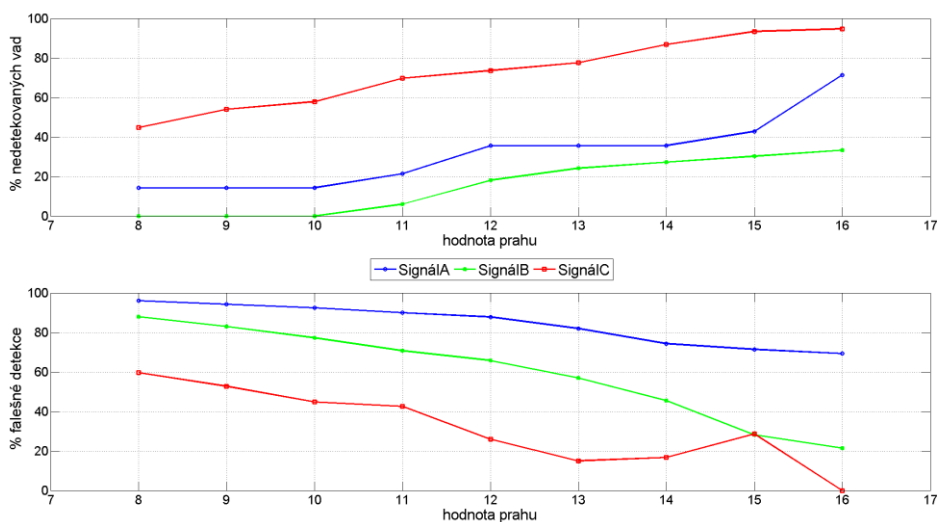
Obr. 6.1: Výsledky detekce pro různé hodnoty parametru *delkaseg*



Obr. 6.2: Výsledky detekce pro různé hodnoty parametru *rad_AR*

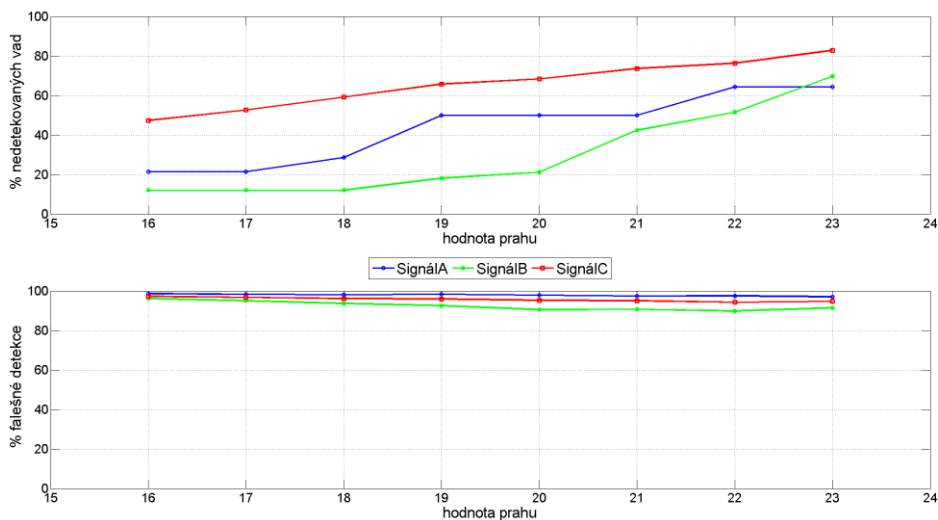
Posledním parametrem je pak *prah_chyb*, jehož výsledky jsou zobrazeny na obr. 6.3. Zde je průběh očekávaný a je patrné, že se zvětšujícím se prahem dochází k růstu procenta nedetekovaných vad a k menšímu počtu falešné detekce a naopak.

Z výsledků vychází, že chování algoritmu ovlivňují hlavně hodnota prahu, ale i délka segmentů. Z hodnot falešné detekce pak vychází, že když chceme snížit procento nedetekovaných vad, tak je lepší snížit hodnotu prahu, než zvětšit délku segmentů. U větší délky segmentů totiž při srovnatelném procentu nedetekovaných vad dochází k většímu počtu falešné detekce. Citlivost algoritmu je tak lepší řídit jen hodnotou prahu a ostatní parametry mít pevně nastavené. Co se týče optimálních hodnot parametrů, tak nastavená délka segmentů se jeví dobře, maximálně by mohla být trochu prodloužena. Hodnota řádu detekci příliš neovlivňovala, ale souvisí s výpočetní náročností algoritmu a menší řád by byl z tohoto hlediska lepší. Z hodnot falešné detekce se jeví proto jako nejlepší možnost řád hodnoty 30. Hodnota prahu pak záleží na kompromisu mezi falešnou detekcí a počtem nedetekovaných vad a dále bude ještě řešena v kapitole 6.4.



Obr. 6.3: Výsledky detekce pro různé hodnoty parametru *prah_chyb*

Pro nalezené hodnoty parametrů byla dále testována metoda AR modelu + matched filtru. Filtr zde zvýrazňuje amplitudu v místě vady oproti okolí, proto byly zvoleny větší hodnoty prahu a zkoumáno, jestli metoda zlepšuje výsledky. Hodnoty detekce jsou zobrazeny na obr. 6.4. Jak je vidět z hodnot falešné detekce, tak přidání filtru zhoršilo výsledky oproti základní metodě z hlediska procenta falešné detekce. Vzhledem k těmto výsledkům není dále tato metoda používána a je uvažován pouze AR model bez matched filtru.



Obr. 6.4: Výsledky detekce pro různé hodnoty parametru *prah_chyb*

6.2 Parametry vlnkové transformace

Parametry u metody vlnkové transformace byly testovány stejným způsobem jako v případě AR modelu. Opět byla měněna jedna hodnota zkoumaného parametru a zbývající zůstávaly v základních hodnotách, které byly:

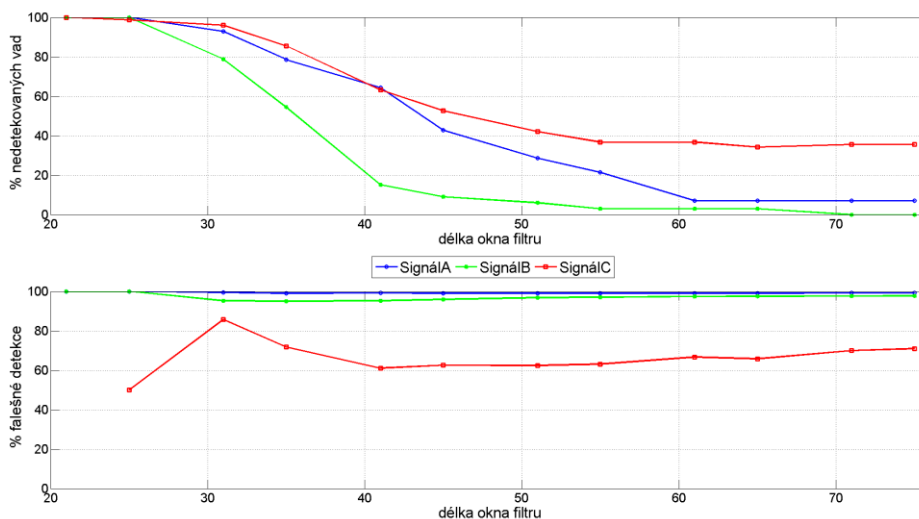
- $delkaseg = 450$
- $krok = 250$
- $delka_fil = 41$
- $prah_fil = 11$
- vlnka Daubechiova řádu 10
- výpočet prahu pomocí mediánového filtru

Na velikosti segmentů u této metody nezáleží, ale hodnoty byly zvoleny stejně jako u AR modelu kvůli stejnému vyřazení šumových segmentů. Délka okna byla zvolena s ohledem k typické délce vad a práh pak tak, aby opět nebyly detekovány všechny vady. Typ vlnky a její řád pak byl zvolen s ohledem na její použití v [10].

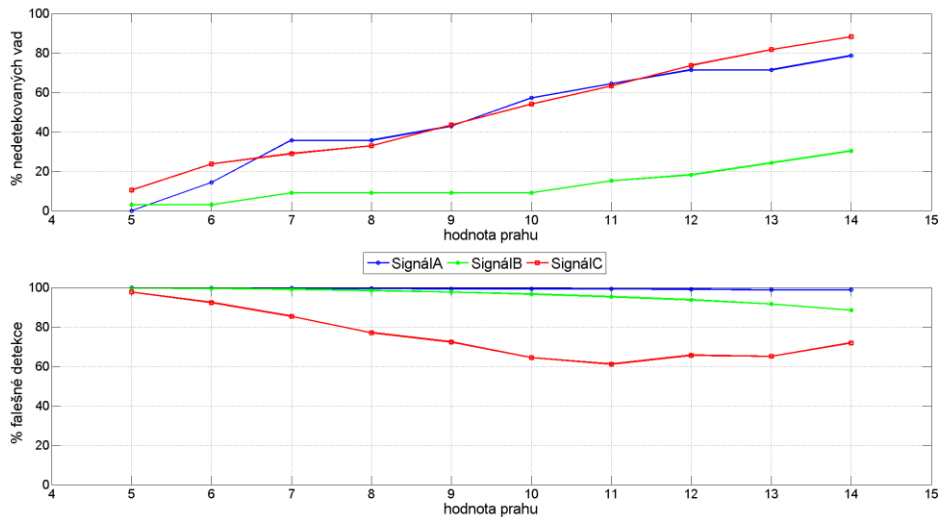
Prvním testovaným parametrem je $delka_fil$, jehož výsledky jsou zobrazeny na obr. 6.5. Je vidět, že délka okna musí mít alespoň určitou délku vzhledem k délce vady (viz rovnice 3.14), jinak nejsou žádné vady detekované. Z hodnot je dále patrná určitá oblast ideální délky okna filtru, od které při dalším prodlužování okna dochází už jen ke zvětšování počtu falešné detekce.

Dalším parametrem je $prah_fil$, jehož výsledky jsou zobrazeny na obr. 6.6. Zde je vidět, že opět platí, že s vyšší hodnotou prahu nastává menší počet falešné detekce, ale zároveň stoupá počet nedetekovaných vad a naopak.

Posledním zkoumaným parametrem je pak řád použité vlnky, jehož výsledky jsou zobrazeny na obr. 6.7. Z výsledků není patrný žádný trend, spíše jsou některé řády z hlediska výsledků detekce lepší než ostatní.

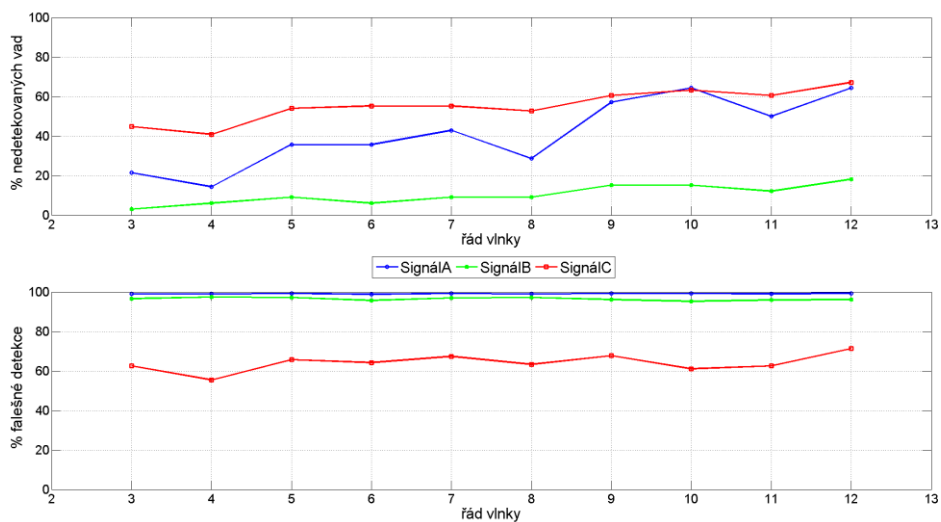


Obr. 6.5: Výsledky detekce pro různé hodnoty parametru $delka_fil$



Obr. 6.6: Výsledky detekce pro různé hodnoty parametru *prah_fil*

Z výsledků plyne, že všechny tři parametry jsou pro výsledky detekce důležité. Co se týče řádu vlnky, tak řád 10 vycházel nejlépe z hlediska nejmenšího počtu falešné detekce. Když vezmeme dále v potaz výsledky u parametru délky okna filtru, tak u tohoto řádů vlnky byl u delšího okna filtru menší počet falešné detekce při menším procentu nedetekovaných vad, než když byl snížen práh pro detekci, který také snížil procento nedetekovaných vad, ale zároveň výrazně zvýšil procento falešné detekce. Z hlediska optimálních parametrů se tak Daubechiová vlnka řádu 10 společně s délkou okna filtru 55 vzorků jeví nejlépe. Citlivost algoritmu lze pak při těchto pevně nastavených hodnotách řídit opět jen hodnotou prahu.



Obr. 6.7: Výsledky detekce pro různé hodnoty řádu Daubechiový vlnky

6.3 Parametry derivace

Parametry u metody derivace byly testovány stejným způsobem jako v předchozích případech. Opět byla měněna jedna hodnota zkoumaného parametru a zbývající zůstávaly v základních hodnotách, které byly:

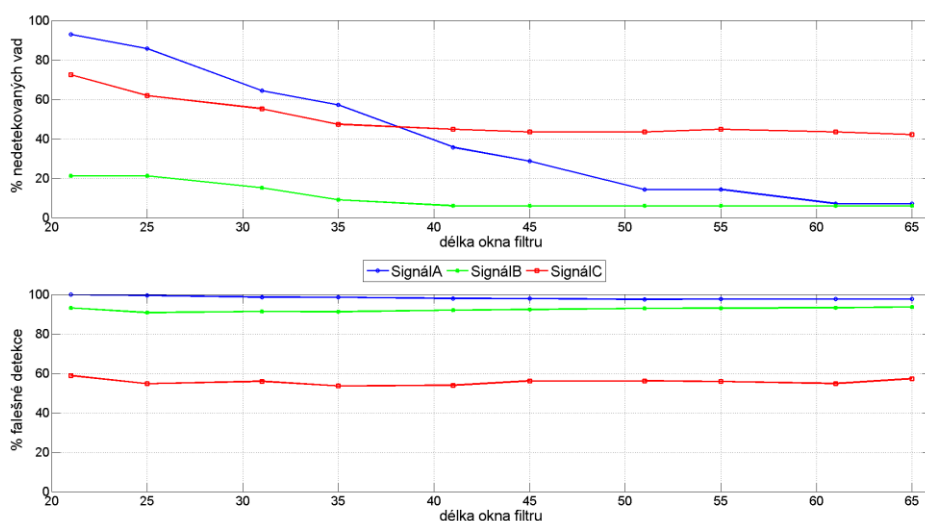
- $delkaseg = 450$
- $krok = 250$
- $delka_fil = 35$
- $prah_fil = 85$
- výpočet prahu pomocí mediánového filtru

Na velikosti segmentů u této metody také nezáleží, ale hodnoty byly zvoleny stejné jako u AR modelu kvůli stejnému vyřazení šumových segmentů. Délka okna byla zvolena s ohledem k typické délce vad a práh pak tak, aby opět nebyly detekovány všechny vady.

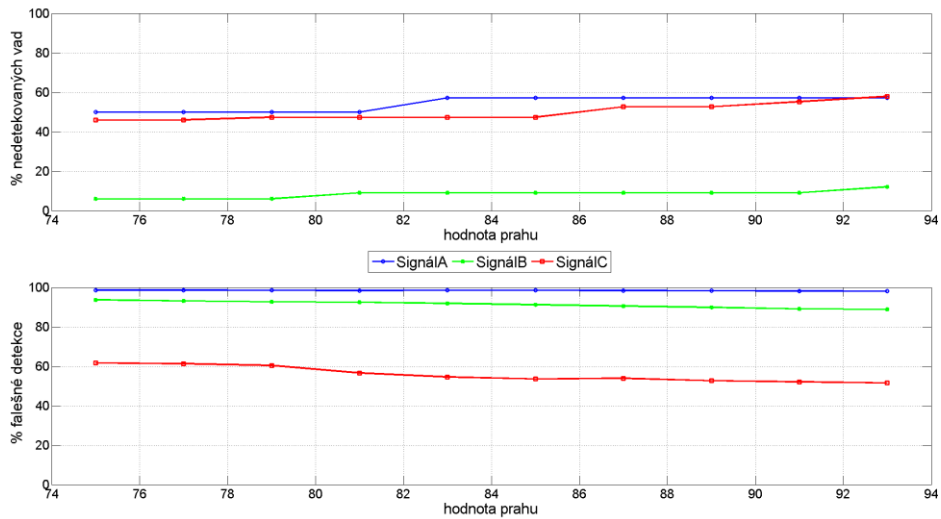
Prvním testovaným parametrem zde je $delka_fil$, jehož výsledky jsou zobrazeny na obr. 6.8. Stejně jako u vlnkové transformace je vidět, že od určité hodnoty už nedochází k výraznému snižování procenta nedetekovaných vad a narůstá už jen procento falešné detekce.

Druhým parametrem pak je $prah_fil$, jehož výsledky jsou zobrazeny na obr. 6.9. Na rozdíl od předchozích metod zde i při větších změnách hodnot prahu nedochází k výrazným změnám výsledků. Se zvyšujícím se prahem zde opět dochází k zmenšování procenta falešné detekce a růstu procenta nedetekovaných vad. Velké hodnoty prahu pak jsou způsobeny velkým rozdílem mezi amplitudou v místě vady a v místech okolních.

Z výsledků plyne, že délka okna filtru pro výpočet prahu je opět důležitá. Jako její optimální hodnota se jeví dobře délka 55 vzorků. Míru falešné detekce vzhledem k procentu nedetekovaných chyb pak lze opět řešit úpravou hodnoty prahu, ale ne tak citlivě jako v případě předchozích metod.

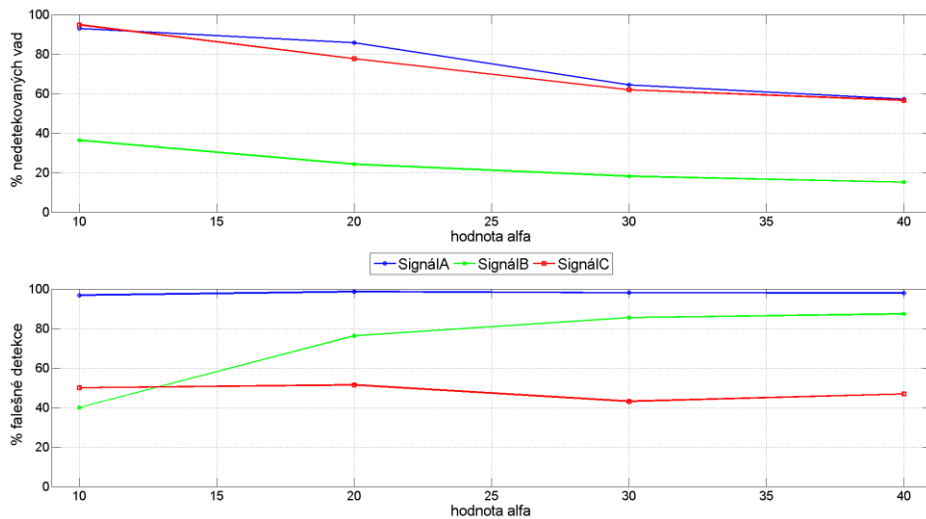


Obr. 6.8: Výsledky detekce pro různé hodnoty parametru $delka_fil$



Obr. 6.9: Výsledky detekce pro různé hodnoty parametru *prah_fil*

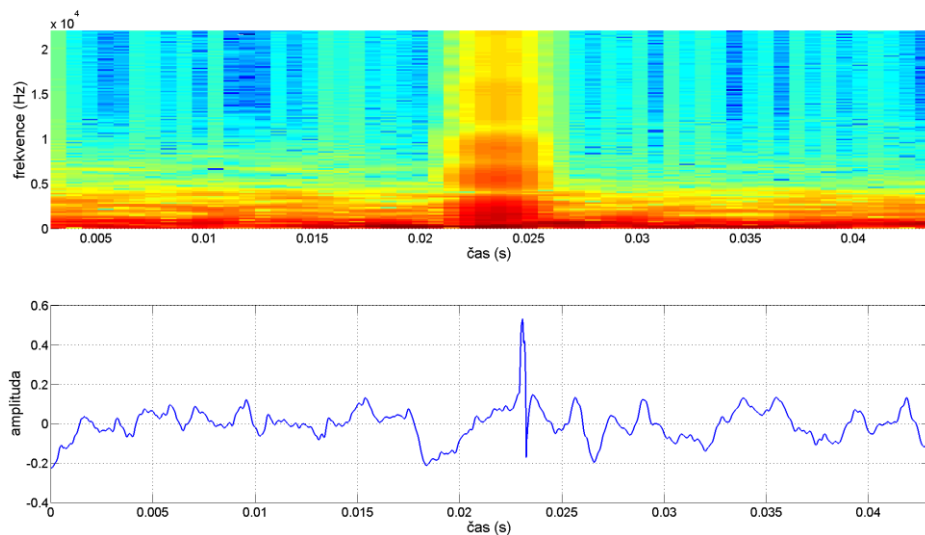
Pro nalezené parametry byl dále vyzkoušen výpočet prahu i pomocí trimmena filtru pro několik hodnot parametru α . Výsledky jsou zobrazeny na obr. 6.10. Pro tento způsob vychází hodnota falešné detekce trochu lépe než u mediánového filtru, ale procento nedetekovaných vad je zase trochu vyšší. Tento způsob výpočtu prahu tedy nenabízí výrazně lepší výsledky a zároveň byl oproti výpočtu mediánového filtru výrazně výpočetně náročnější. Z těchto důvodů byl tedy zvolen výpočet prahu pomocí mediánového filtru jako lepší a je dále používán jen tento způsob.



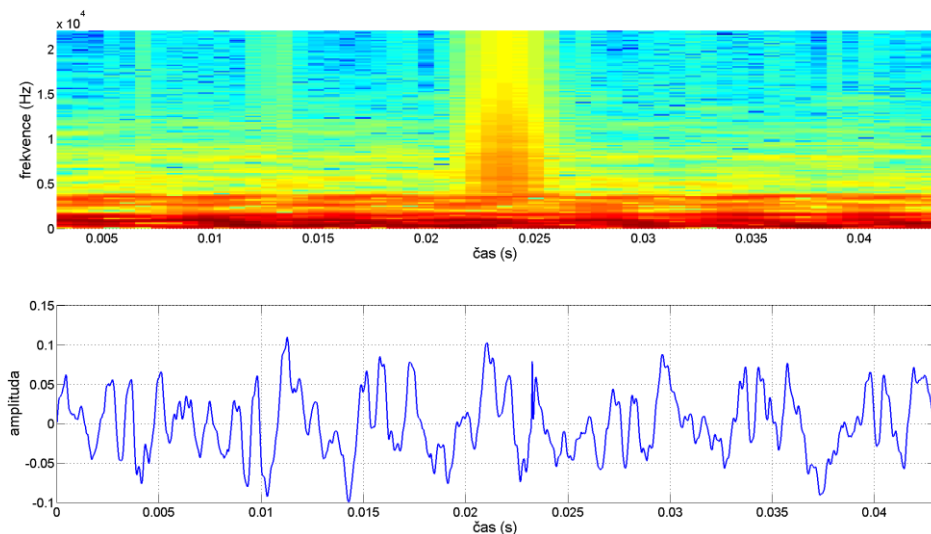
Obr. 6.10: Výsledky detekce pro různé hodnoty parametru α

6.4 Úroveň detekce a ověření detekce

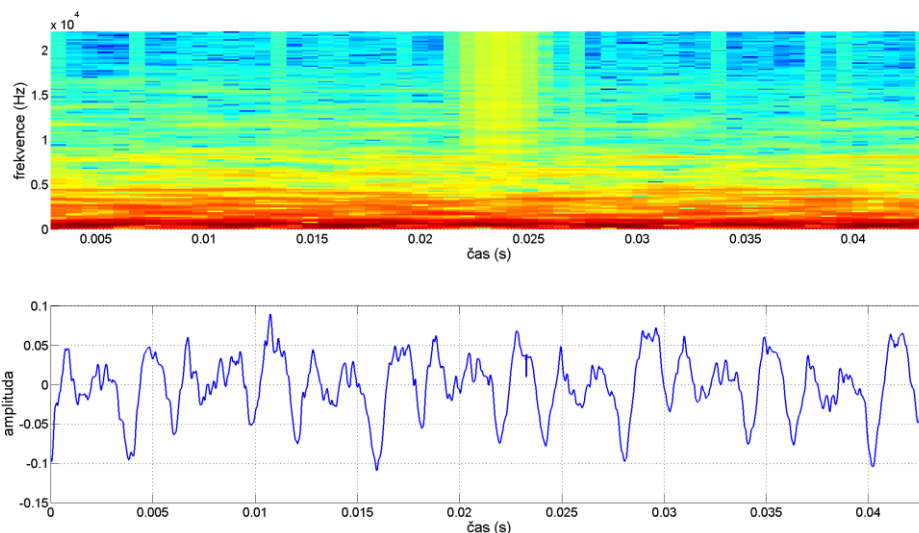
V předchozích částech byly nalezeny parametry pro detekci s tím, že pomocí změny hodnot prahů lze upravovat jistým způsobem míru mezi falešnou detekcí a počtem nedetekovaných vad. To souvisí s tím, jestli chceme hledat jen závažné vady nebo i menší vady. V softwaru firmy GZ Media, a.s je u nalezených vad také uveden stupeň jejich vážnosti. Stupňů je celkem šest s tím, že první dva stupně jsou nejzávažnější a takto ohodnocené vady by měly být nalezeny a práh pro detekci by tedy měl být přizpůsoben vzhledem k těmto vadám. Takto ohodnocené vady mají typicky velkou amplitudu, která přesahuje hodnoty okolního signálu. Z hlediska výstupní kontroly to jsou pak vady, které nejsou ve výrobku přípustné. Ukázka takové vady je zobrazena na obr. 6.11. Třetí a čtvrtý stupeň pak je ohodnocen jako rušivý respektive slyšitelný a typicky to jsou malé lupance, které mají amplitudu v úrovni signálu. Ukázka takové vady je na obr. 6.12. Poslední dva stupně pak představují typicky malé vady, které mají malou amplitudu, případně se jedná o nějaké zkreslení signálu. Ukázka takto hodnocené vady je na obr. 6.13.



Obr. 6.11: Časový průběh a spektrogram vady hodnocené nejzávažnějším stupněm



Obr. 6.12: Časový průběh a spektrogram vady hodnocené stupněm slyšitelné



Obr. 6.13: Časový průběh a spektrogram vady hodnocené nejnižším stupněm

Prahy u jednotlivých detekčních metod byly nastaveny s ohledem na tyto skutečnosti tak, aby byly detekovány zmiňované nejvážnější vady, které odpovídají prvním dvěma stupňům hodnocení. Prahy tak nejsou nastaveny na příliš nízkou hodnotu, která by zajistila detekci i malých vad. Jak je vidět na obr. 6.13 malé vady se totiž ve spektru neprojeví výraznou složkou na vyšších frekvencích, na které je založená metoda pro ověřování detekce, a tak by byly stejně vyřazovány. Zároveň tyto vady nejsou důvodem k vyřazení výrobku. Pro zvolené parametry jednotlivých detekčních metod pak je detekce schopna zachytit ve většině případů vady, které se projevují tak, jak je zobrazeno na obr. 6.12 a v závislosti na konkrétních případech i vady hodnoceny nejmenšími stupni. Výsledné parametry pro jednotlivé metody pak jsou následující:

AR model

- *delkaseg* = 450
- *krok* = 360
- *rad_AR* = 30
- *prah_chyb* = 11
- *prah_chyb2* = 3

Vlnková transformace

- *delka_fil* = 55
- *prah_fil* = 8
- vlnka Daubechiova řádu 10
- výpočet prahu pomocí mediánového filtru

Derivace

- *delka_fil* = 55
- *prah_fil* = 70
- výpočet prahu pomocí mediánového filtru

Společný parametr *b* pak má hodnotu 25.

Konečné výsledky s tímto nastavením pro tři testované signály jsou uvedeny v tabulce 6.1. V prvním sloupci pod názvem počet nalezených je uveden počet chyb, které byly nalezeny a jsou v příloženém seznamu vad. Ve druhém sloupci pak je pod názvem celkový počet uveden celkový

počet případů, které algoritmus vyhodnotil jako vadu, tedy jedná se o součet falešných a správných detekcí. Zároveň jsou uvedeny i kritéria procento nedetekovaných vad a procento falešné detekce.

Tab. 6.1: Výsledky detekce pro testované signály

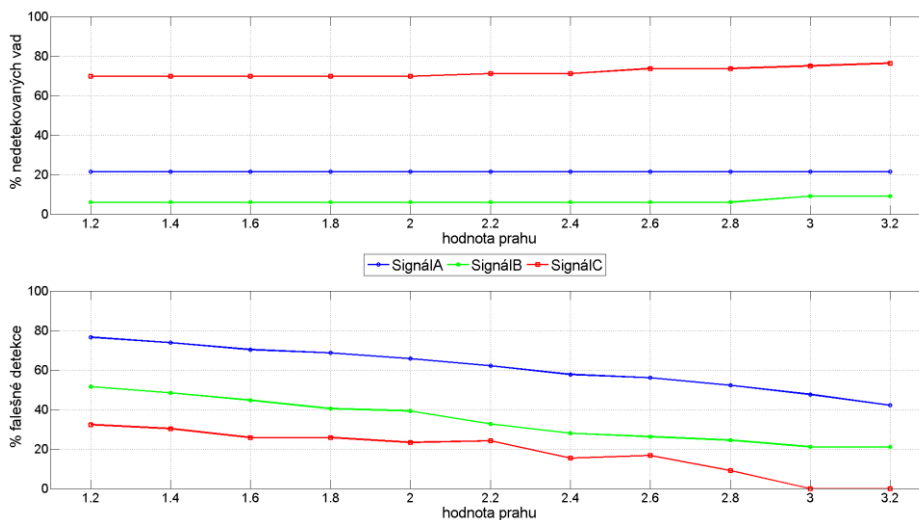
metoda	AR model		Vlnková transformace		Derivace	
	počet nalezených / % nedetekovaných vad	celkový počet / % falešné detekce	počet nalezených / % nedetekovaných vad	celkový počet / % falešné detekce	počet nalezených / % nedetekovaných vad	celkový počet / % falešné detekce
signálA	11 / 21,4	113 / 90,3	13 / 7,1	2741 / 99,5	13 / 7,1	788 / 98,4
signálB	31 / 6,1	107 / 71	32 / 3	2750 / 98,8	32 / 3	747 / 95,7
signálC	23 / 69,7	43 / 46,5	59 / 22,4	323 / 81,7	48 / 36,8	144 / 66,7

Počty chyb v seznamu pro jednotlivé signály pak jsou následující:

- signálA = 14
- signálB = 33
- signálC = 76

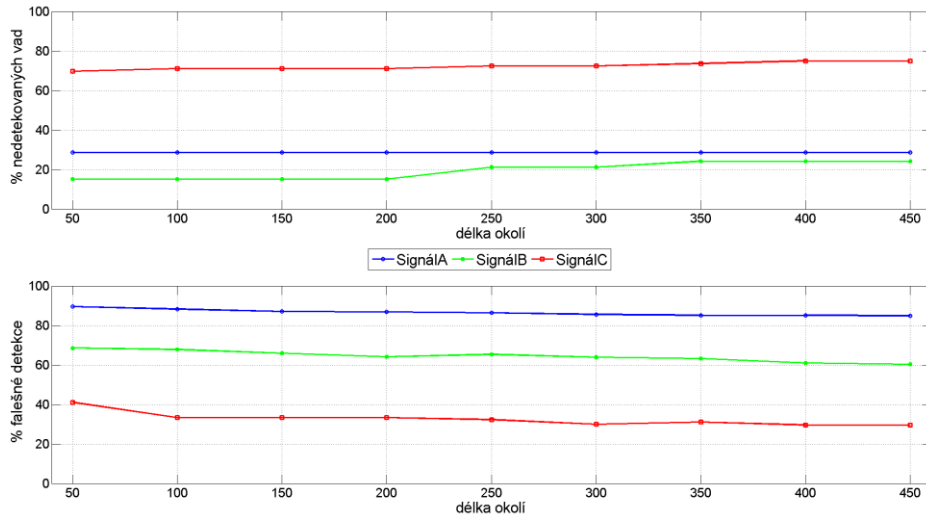
Jak bylo zmíněno hodnoty parametrů byly nastaveny tak, aby byly hledány jen větší chyby. Proto nejsou nalezeny všechny chyby, ale ty, které jsou ohodnoceny v seznamu jako závažné, jsou nalezeny všechny. Výrazně je tato skutečnost znát v případě signáluC, kde je většina vad hodnocena nejmenším stupněm a větších vad se v signálu vyskytuje jen kolem dvaceti a ty byly nalezeny.

Jak je vidět z tabulky 6.1, tak je ve výsledcích velké množství falešných detekcí, které lze částečně odstranit pomocí implementovaného ověření. Pro nalezení optimálních parametrů bylo stejně jako v případě detekčních algoritmů provedeno několik průběhů s různými hodnotami parametrů. K tomuto testování byly použity uvedené výsledky detekce pomocí AR modelu a porovnání proběhlo opět pomocí kritérií procenta nedetekovaných vad a procenta falešné detekce. Výsledky pro ověření pomocí spektra jsou zobrazeny na obr. 6.14. Jak je vidět, tak s rostoucí hodnotou prahu klesá počet falešné detekce a procento nedetekovaných chyb se začne zvětšovat až u největších hodnot a to u signáluC, kde se vyřadí některé menší chyby, ale ty velké jsou stále správně vyhodnoceny. Jako optimální se tak jeví hodnota 3, u které jsou některé menší vady stále vyhodnoceny jako vady.



Obr. 6.14: Výsledky po ověření pro různé hodnoty parametru *prah_s*

Druhým typem ověření je průběh součtové a rozdílové složky, kde se nastavuje velikost okolí, z které se určuje hranice. Výsledky pro toto ověření jsou zobrazeny na obr. 6.15. Jak je vidět, tak toto ověření vyřadilo více vad, které jsou obsaženy v seznamu a tak stouplu procento nedetekovaných vad. Zároveň ale nedošlo k vyřazení tolika falešných detekcí jako u ověření pomocí spektra. S větší velikostí okolí pak klesá hodnota falešné detekce, ale ne moc výrazně, a zároveň také dochází k většímu vyřazení skutečných vad.



Obr. 6.15: Výsledky po ověření pro různé hodnoty parametru *delka_oko*

Z výsledků tedy plyne, že je lepší použít ověření pomocí spektra, které vyřadí více falešných detekcí a zároveň někdy vyřadí jen některé menší vady. Pro určený parametr prahu u tohoto ověření, který má zmíněnou hodnotu 3, jsou konečné výsledky detekčního algoritmu pro testované signály uvedeny v tabulce 6.2 opět ve stejném tvaru jako v předchozím případě.

Tab. 6.2: Konečné výsledky detekce po přidání ověřovacího kroku

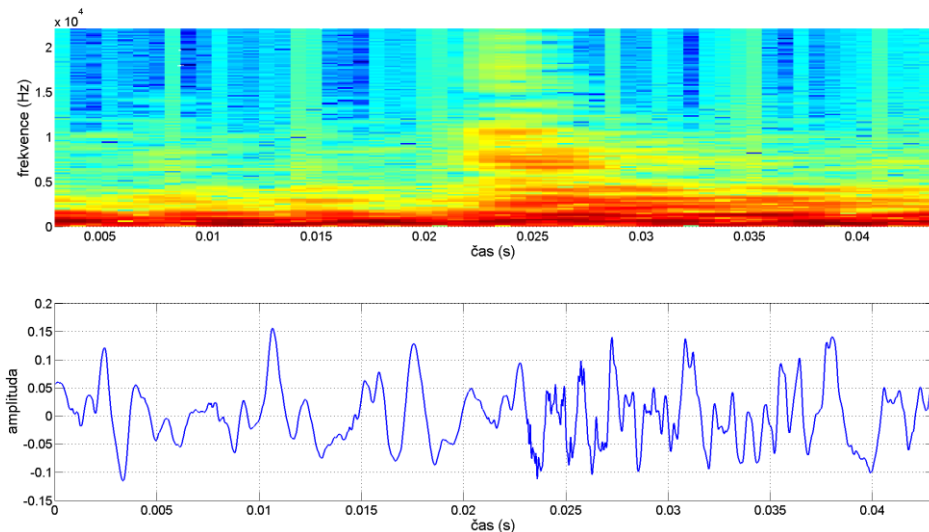
metoda	AR model		Vlnková transformace		Derivace	
	počet nalezených / % nedetekovaných vad	celkový počet / % falešné detekce	počet nalezených / % nedetekovaných vad	celkový počet / % falešné detekce	počet nalezených / % nedetekovaných vad	celkový počet / % falešné detekce
signálA	11 / 21,4	21 / 47,6	13 / 7,1	37 / 64,9	13 / 7,1	29 / 55,2
signálB	30 / 9,1	38 / 21,1	31 / 6,1	43 / 27,9	31 / 6,1	41 / 24,4
signálC	19 / 75	19 / 0	42 / 44,7	52 / 19,2	34 / 55,3	40 / 15

Konkrétní výsledky pro jednotlivé vady v seznamu pro signálA, signálB a signálC jsou pak uvedeny v tabulkách 6.3, 6.4 a 6.5.

6.5 Vyhodnocení

Jak je vidět z konečných výsledků uvedených v tabulce 6.2, tak všechny algoritmy po zařazení ověřovacího kroku poskytují poměrně dobré výsledky a našly podstatné vady vyskytující se v testovaných signálech. Z výsledků dále vidíme, že i po ověřovacím kroku zbyly ve výsledcích nějaké falešné detek-

ce. Tyto detekce jsou ve většině případů menší vady, které nejsou uvedeny v seznamech, ale co se týče vážnosti, řadily by se do skupiny nejméně vážných vad. Výjimkou pak jsou některé falešná detekce, které nebyly vyřazeny u detekce pomocí vlnkové transformace. Tyto oblasti obsahují krátkou pasáž, která má složky i na vyšších frekvencích, a při ověřování má obdobné vlastnosti jako skutečné vady. Ilustrace této falešné detekce je na obr. 6.16. Tato skutečnost by mohla být problémem u nějakých signálů, které obsahují obdobné krátké úseky s obsahem na vyšších frekvencích.



Obr. 6.16: Falešná detekce, která nebyla vyřazena při ověřovacím kroku

Z hlediska reálné implementace je také důležitá výpočetní náročnost metod. V tabulce 6.6 jsou uvedeny časové údaje pro jednotlivé metody. Algoritmy byly testovány na počítači s procesorem Core i5 a pamětí RAM 8GB. Údaje odpovídají délce výpočtu pro signál dlouhý 15 minut. Nejrychleji proběhl výpočet u metody AR modelu, dále následovala metoda derivace a nejpomaleji proběhl výpočet u metody vlnkové transformace. U metod derivace a vlnkové transformace je délka výpočtu ovlivněna délkou segmentů, která byla v tomto testu nastavena na délku 88200 vzorků. Pro nejrychlejší výpočet by bylo nejvhodnější detekci provést rovnou na celém signálu, ale to bylo příliš náročné na paměť. Zpracování po segmentech, ale může být použito při zpracování v reálném čase, proto byla zvolena pro test tato délka. Tyto dvě metody pak jsou výpočetně náročnější zejména kvůli výpočtu mediánového filtru.

Tab. 6.6: Časová náročnost algoritmů

metoda	čas výpočtu
AR model	110 s
derivace	253s
vlnková transformace	570 s

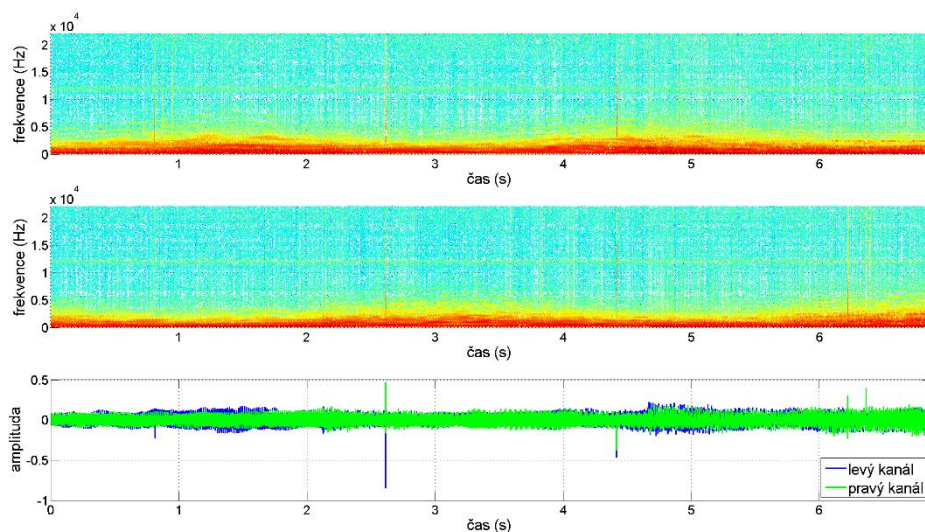
Když vezmeme v potaz výpočetní náročnost a zmíněnou falešnou detekci u vlnkové transformace, tak se tato metoda nejeví jako nejvhodnější. Už z výsledků v tabulce 6.1 je vidět, že u této metody dochází k opravdu velkému počtu falešných detekcí, které se pak nemusí podařit všechny odstranit. Dále je trochu problematické nastavení parametrů detekce z hlediska různé délky vad. Pro největší vady je třeba nastavit délku okna filtru pro výpočet prahu na větší hodnotu, která pak může způsobovat zmíněnou falešnou detekci zobrazenou na obr. 6.16. Když pak dále porovnáme metodu

AR modelu a metodu derivace, tak AR model vychází výpočetně méně náročný a když porovnáme výsledky bez ověření v tabulce 6.1, tak AR model vykazuje také méně falešné detekce. Z těchto hledisek tedy vychází AR model v porovnání k ostatním dvěma metodám jako nejlepší. Výběr metody může také ovlivnit, jestli chceme detekovat i menší vady. K tomu se spíše hodí metoda derivace, u které má detekční signál výraznou amplitudu i v případě menších vad. U metody AR modelu se k takové detekci musí snížit práh a to vede k falešným detekcím v místech podobným, jako je zobrazeno na obr. 6.16. U metody derivace se tento problém nevyskytoval. V tomto případě by šlo také uvažovat o spojení obou metod a zvýšení spolehlivosti. Jak je vidět z výsledků v tabulkách 6.3, 6.4 a 6.5, tak u některých vad dojde k detekci jen u jedné z metod a spojením výsledků by se dosáhlo lepší úspěšnosti i u detekce menších vad. Pokud chceme detekovat jen ty podstatné vady, tak metoda AR modelu vychází nejlépe, protože je nejrychlejší a pomocí hodnoty prahu lze poměrně dobře ovládat míru detekce.

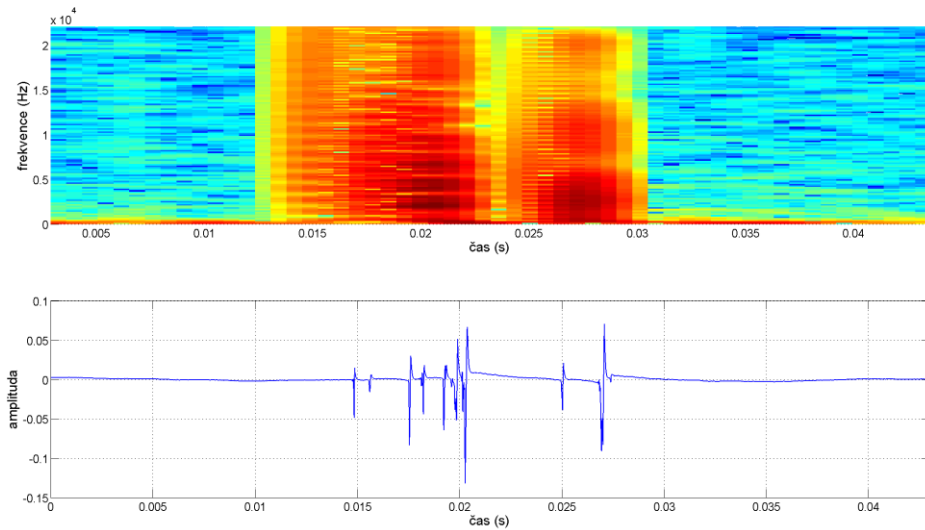
6.6 Možnosti klasifikace

Možnosti klasifikace byly zkoumány na ukázkách poskytnutých firmou GZ Media, a.s, které obsahovaly některé vady uvedené v tabulce 2.1. Jednalo se o krátké úseky, které obsahovaly danou vadu. Z ukázek vyplynulo, že se všechny vady projevují typickým překmitem, který byl uveden v úvodu na obr. 2.2. Z tohoto hlediska se tedy vady příliš rozdělit nedají. Možný způsob klasifikace pak je vzít v úvahu časové rozmístění vad, jestli se periodicky opakují nebo jsou umístěny blízko sebe, případně se objevují v nějakém shluku.

Vzhledem k principu záznamu lze dobře pomocí časové posloupnosti klasifikovat vady, které se projevují na otáčku, a oddělit tak minimálně v klasifikaci vady, které se tak projevují. Jednou z vad, které se opakují na otáčku, je škrábanec, který vytvoří lupance, které se periodicky opakují. Ukázka tohoto případu je na obr. 6.17. Jedná se o posloupnost čtyř lupanců, které mohly být způsobeny zmíněným škrábanec, nebo pak škráblou matricí. Z ukázek se tyto dva případy nelišily, takže odlišit by se dali až případně testem další desky. Další vadou, která se obvykle takto periodicky opakuje, jsou nedolisy. Ty se zároveň vyznačují tak, že narušují pravý kanál signálu. Ukázka této vady je zobrazena na obr. 6.18. Jak je dále patrné z obrázku, tak vada se nevyznačuje jen jedním překmitem jak je obvyklé u ostatních vad, ale je jich přítomno více v těsné blízkosti u sebe. Toho lze využít pro případné odlišení od škrábance, kde můžou být lupance také jen v pravém kanálu. Z popsaných hledisek by tedy šlo tyto vady od sebe oddělit.

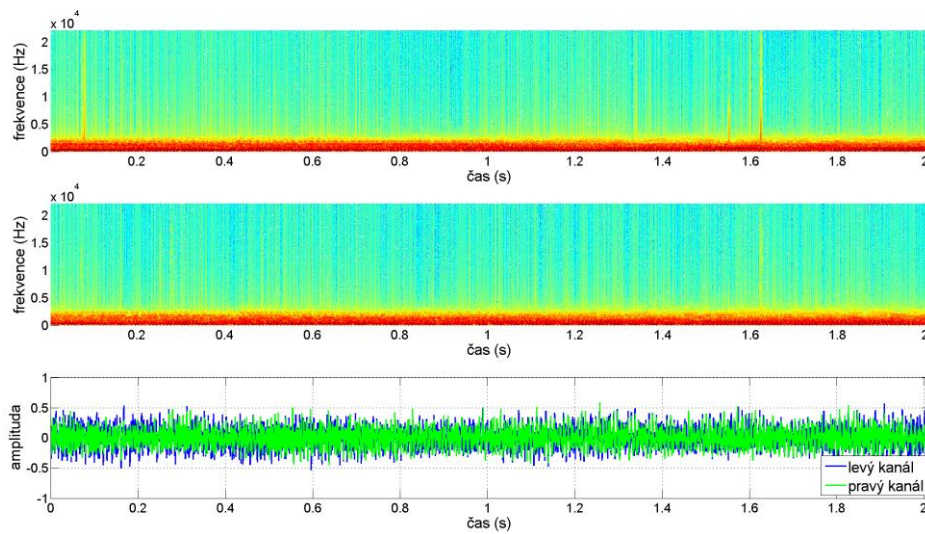


Obr. 6.17: Spektrogram(nahoře levý kanál) a časový průběh lupanců opakujících se na otáčku



Obr. 6.18: Spektrogram a časový průběh pravého kanálu obsahující nedolisy

Dalším typem vady je praskot, který se projevuje větší posloupností lupanců v krátkém časovém úseku. Ukázka takového případů je na obr. 6.19. Když tedy detekujeme větší množství vad v nějakém krátkém časovém úseku a vada se neopakuje na otáčku, případně se na otáčku opakuje a zároveň k tomu jsou přítomny i další lupance, jedná se nejspíše o nějakou vadu, která způsobuje praskot, jako jsou návalky.



Obr. 6.19: Spektrogram(nahore levý kanál) a časový průběh praskotu

7 Závěr

Práce se zabývala problematikou mechanického záznamu a jeho vad. V první části práce byl popsán princip záznamu a jeho výroby a uveden přehled typických vad u tohoto záznamu, které vznikají při jeho výrobě. Jednalo se zejména o impulsní vady. V další části byly popsány metody, pomocí kterých lze detekovat tyto vady. Konkrétně se jedná o metodu AR modelu, vlnkové transformace a derivace. Dále byly uvedeny možnosti, jak eliminovat falešnou detekci pomocí spektrálních a časových charakteristik. Zde bylo využito typického projevu impulsních vad v signálu. V prostředí Matlab byly dále tyto metody implementovány v podobě skriptů, které jsou přiloženy k práci. Vytvořené algoritmy byly dále otestovány na třech reálných signálech, které byly záznamem celých desek v délkách 20 a 15 minut. Nejdříve byly nalezeny pro všechny metody vhodné parametry, které zajistily požadovanou úroveň detekce. Poté byl proveden samotný test a výsledky detekce pak byly porovnány se softwarem firmy GZ Media, a.s. Z výsledků vyšlo, že u všech tří metod lze dosáhnout dobrých výsledků a podařilo se najít všechny podstatné vady a i některé menší. Dále byly uvedeny možnosti, jak klasifikovat některé vady podle jejich časového rozmístění v signálu.

Jako nejlepší algoritmus vychází z výsledků metoda AR modelu, která nevykazuje takové množství falešné detekce a také lze u ní dobře ovládat parametry z hlediska nastavení detekce. Oproti tomu metoda vlnkové transformace a derivace je z hlediska nastavení parametrů trochu problematická kvůli různé délce vad. Zejména u metody vlnkové transformace kvůli tomu docházelo k velkému počtu falešných detekcí. Metoda AR modelu také vychází lépe z hlediska výpočetní náročnosti, která byla testována. U metod vlnkové transformace a derivace je totiž třeba provádět výpočetně náročnou mediánovou filtraci. Z hlediska výstupní kontroly, kde je třeba najít hlavně ty nejvážnější vady, je metoda AR modelu také nejvhodnější, protože jde na rozdíl od ostatních dvou metod lépe ovládat míru detekce pomocí nastaveného prahu.

Co se týče falešné detekce a její eliminace, tak použitá metoda, která využívá spektrálních charakteristik, počet falešné detekce snížila na poměrně přijatelnou míru, ale stále je přítomna. Druhá metoda, která využívá časových charakteristik, se pak příliš neosvědčila. Z hlediska další práce by tedy bylo vhodné použít metodu AR modelu pro detekci a najít další kritéria, která by spolehlivěji tuto detekci ověřovala.

Literatura

- [1] DUŠEK, Karel. *Záznam a reprodukce zvuku*. 5 vydání. Praha: Vydavatelství ČVUT, 1982.
- [2] Směry záznamu stereofonních kanálů v drážce. In: *ebastlirna* [online]. [cit. 2016-05-18]. Dostupné z: <http://www.ebastlirna.cz/modules.php?name=News&file=article&sid=203>
- [3] *Manufacturing GZ Vinyl* [online]. [cit. 2016-02-01]. Dostupné z: <http://www.gzvinyl.com/Manufacturing.aspx>
- [4] OUDRE, Laurent. Automatic Detection and Removal of Impulsive Noise in Audio Signals. *Image Processing On Line*. 2015, 5, 267-281.
- [5] VASEGHI, S.V. a P.J.W. RAYNER. A new application of adaptive filters for restoration of archived gramophone recordings. In: *ICASSP-88., International Conference on Acoustics, Speech, and Signal Processing*. 1988, s. 2548-2551.
- [6] GODSILL, Simon J a Peter J RAYNER. *Digital audio restoration: a statistical model based approach*. New York: Springer, 1998, xix, 328 p. ISBN 3540762221.
- [7] ESQUEF, Paulo AA, Luiz WP BISCAINHO, Paulo SR DINIZ a Fhabio P FREELAND. A double-threshold-based approach to impulsive noise detection in audio signals. In: *X European Signal Processing Conference (EUSIPCO 2000), Tampere, Finland*. 2000, s. 2041-2044.
- [8] NONGPIUR, R. C. a D. J. SHPAK. Impulse-noise suppression in speech using the stationary wavelet transform. *The Journal of the Acoustical Society of America*. 2013, **133**(2), 866–879.
- [9] JAN, Jiří. *Číslicová filtrace, analýza a restaurace signálů*. 2. upr. a rozš. vyd. Brno: VUTIUM, 2002. ISBN 80-214-1558-4.
- [10] RAJMÍČ, Pavel a Jaroslav KLÍMEK. Removing crackle from an LP record via wavelet analysis. In: *Conference on Digital Audio Effects (DAFx'04), Naples, Italy*. 2004, s. 38-41.
- [11] BEDNAR, J. BEE a TERRY L. WATT. Alpha-trimmed means and their relationship to median filters. *IEEE Transactions on Acoustics, Speech and Signal Processing*. 1984, **32**(1), 145-153.
- [12] KAUPPINEN, Ismo. Methods for detecting impulsive noise in speech and audio signals. In: *International Conference on Digital Signal Processing*. 2002, s. 967-970.
- [13] BERGL, P. Porovnání metod pro měření odlišnosti spekter signálů. In: *Konference Technical Computing Prague 2006*. 2006, s. 17.

Příloha A

Tab. 6.3: Seznam vad pro signálA a výsledky detekce

začátek vady	AR model	Vlnková transformace	Derivace
1236165	1	1	1
2505611	1	1	1
2584982	1	1	1
4765438	0	1	1
5324097	1	1	1
5752509	0	1	1
6151225	1	1	1
10231622	1	0	0
30325418	0	1	1
34010830	1	1	1
38490589	1	1	1
39681419	1	1	1
39760807	1	1	1
39840198	1	1	1

Tab. 6.4: Seznam vad pro signálB a výsledky detekce

začátek vady	AR model	Vlnková transformace	Derivace
206444	1	1	1
1999438	1	1	1
4711618	1	1	1
6220012	1	1	1
6226034	1	1	1
6281922	1	1	1
6286925	1	1	1
6292316	1	1	1
6297351	1	1	1
6305036	1	1	1
6353154	1	1	1
6618984	1	1	1
6684288	1	1	1
6692336	1	1	1
6711818	1	1	1
7013442	1	1	1
7249746	1	1	1
7483322	1	1	1
7648752	1	1	1
8027490	1	1	1
8438401	1	1	1
12186970	1	1	1
12266200	1	1	1
12490254	1	1	1
12504348	1	1	1
13364175	0	0	0
36129772	1	1	1
37090128	0	0	0
39120027	1	1	1
42003013	1	1	1
43794762	1	1	1
45229109	0	1	1
51332368	1	1	1

Tab. 6.5: Seznam vad pro signálC a výsledky detekce

začátek vady	AR model	Vlnková transformace	Derivace				
				21207044	0	1	0
				21286431	0	0	0
778405	0	0	0	21324256	0	1	1
1603319	0	0	0	21365811	1	1	1
1924018	0	0	0	21838597	0	1	1
2193286	0	0	0	21993095	0	1	1
2463640	1	1	1	22469000	0	1	1
2914128	0	1	1	22714500	0	0	0
4020034	1	1	1	22758414	0	1	0
4822305	0	0	0	22925123	1	1	0
5656129	0	0	0	23191161	0	0	0
6162327	0	0	0	24544803	0	0	0
6502845	0	0	0	24745006	1	1	1
6517351	0	1	0	25647889	1	1	1
6540019	0	1	1	26378236	0	0	0
7132490	0	1	0	26516480	0	1	1
7194173	0	0	0	26520217	0	0	1
7203807	0	0	0	26885456	0	1	1
7690974	0	1	1	27262345	0	0	0
7755624	0	0	0	29495626	1	1	1
7798123	0	0	0	29811161	1	0	0
8644418	0	0	1	30697934	0	0	0
9780723	0	0	0	30739113	0	0	0
11356019	0	1	0	31603626	0	1	1
11682468	0	0	0	32377894	0	1	0
12026855	0	0	0	32763897	0	0	0
12144705	0	1	1	33491944	1	1	1
12224087	1	1	1	33829097	0	0	0
12303464	1	1	1	34069755	1	1	1
12382506	1	1	1	34182105	0	0	0
12461907	1	1	1	34228550	0	1	0
12883201	1	1	1	34663509	0	0	0
16506884	1	1	1	35049605	0	1	1
16767149	0	1	1	35756968	0	0	0
16932630	1	1	1	36088348	1	1	1
17611307	0	1	0	36778714	0	0	0
18758854	0	1	0	36925172	0	0	0
20651305	0	0	0	37224666	0	0	0
20767461	0	1	1	37258876	0	1	1
20801140	1	1	1	40013778	0	0	0

Příloha B

Obsah přiloženého CD

skript_vlnkova_detekce.m – detekce pomocí vlnkové transformace
skript_derivace_detekce.m – detekce pomocí derivace
skript_AR_detekce.m – detekce pomocí AR modelu
skript_matched_detekce.m – detekce pomocí AR modelu + matched filtru

vlnkova_segment.m – funkce pro detekci pomocí vlnkové transformace
derivace_segment.m – funkce pro detekci pomocí derivace
AR_segment.m – funkce pro detekci pomocí AR modelu
matched_segment.m – funkce pro detekci pomocí AR modelu + matched filtru

spojeni_chyb.m – funkce pro úpravu detekčního vektoru
souradnice_chyb.m – funkce pro nalezení souřadnic vad
overeni_chyb.m – funkce pro ověření detekce vad
trimmeanfilt.m – funkce realizující trimmean filtraci

DP.pdf – práce v elektronickém formátu