

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2017

Bc. Jindřich Pevný



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY**

**A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

**ÚSTAV TELEKOMUNIKACÍ**

DEPARTMENT OF TELECOMMUNICATIONS

## **METODY PRO SPEKTRÁLNÍ ANALÝZU S VYSOKÝM ROZLIŠENÍM**

METHODS FOR HIGH RESOLUTION SPECTRAL ANALYSIS

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. Jindřich Pevný**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. Petr Frenštátský**

**BRNO 2017**

# Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

**Student:** Bc. Jindřich Pevný

**ID:** 155217

**Ročník:** 2

**Akademický rok:** 2016/17

**NÁZEV TÉMATU:**

## **Metody pro spektrální analýzu s vysokým rozlišením**

**POKYNY PRO VYPRACOVÁNÍ:**

Seznamte se s metodami spektrálního zaměření pro spektrální analýzu s vysokým rozlišením, jako je spektrální odhad, komplexní demodulace a Zoom FFT. Porovnejte dané metody na základě dosažitelné přesnosti a výpočetní náročnosti. Nejvhodnější metoda pro spektrální analýzu v reálném čase bude implementována v jazyce C++ ve formě spektrogramu.

**DOPORUČENÁ LITERATURA:**

[1] SYSEL, P.; SMÉKAL, Z. Číslicové filtry. Brno: Vysoké učení technické v Brně, 2012. s 1-148. ISBN 978-8-214-4454-6

[2] OPPENHEIM, A., V., SCHAFER, R. W. Digital Signal Processing. London: Prentice-Hall, Inc., 1975. s.1 - 585. ISBN 0-13-214635-5

**Termín zadání:** 1.2.2017

**Termín odevzdání:** 24.5.2017

**Vedoucí práce:** Ing. Petr Frenštátský

**Konzultant:**

**doc. Ing. Jiří Mišurec, CSc.**  
*předseda oborové rady*

**UPOZORNĚNÍ:**

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

## **ABSTRAKT**

Tato práce se zabývá problematikou spektrální analýzy ve vysokém rozlišení. V první části práce jsou představeny vybrané metody, které jsou následně na základě implementace v programovém prostředí aplikace Matlab porovnány. Zároveň je řešena i problematika redukce interferenčních složek v kvadratických časově–frekvenčních distribucích. Druhá část práce se věnuje implementaci a optimalizaci algoritmu pro výpočet vyhlazené Wignerovy distribuční funkce v reálném čase.

## **KLÍČOVÁ SLOVA**

spektrální analýza, ZoomFFT, Wignerova distribuční funkce, kvadratické časově–frekvenční distribuce

## **ABSTRACT**

This thesis deals with the topic of high resolution spectral analysis. In the first part, selected methods are presented and afterwards compared based on the Matlab implementations. The problematics of reduction of crossterms in quadratic time–frequency distributions is also covered. The second part is focused on the implementation and optimization of the algorithm for real-time computation of smoothed Wigner distribution function.

## **KEYWORDS**

spectral analysis, ZoomFFT, Wigner distribution function, quadratic time–frequency distributions

PEVNÝ, Jindřich *Metody pro spektrální analýzu ve vysokém rozlišení*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2017. 63 s. Vedoucí práce byl Ing. Petr Frenštátský



## PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Metody pro spektrální analýzu ve vysokém rozlišení“ jsem vypracoval(a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně 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(-ky)

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Petru Frenštátskému za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno .....

.....

podpis autora(-ky)



Faculty of Electrical Engineering  
and Communication  
Brno University of Technology  
Purkynova 118, CZ-61200 Brno  
Czech Republic  
<http://www.six.feec.vutbr.cz>

## PODĚKOVÁNÍ

Výzkum popsany v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno .....

.....

podpis autora(-ky)



EVROPSKÁ UNIE  
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ  
INVESTICE DO VAŠÍ BUDOUCNOSTI



# OBSAH

Úvod	12
<b>1 Fourierova transformace</b>	<b>13</b>
1.1 Definice	13
1.1.1 Krátkodobá Fourierova transformace	13
1.1.2 Rychlá Fourierova transformace (FFT)	14
<b>2 Metody pro spektrální přibližování založené na Fourierově transformaci</b>	<b>16</b>
2.1 Komplexní demodulace	16
2.2 Vernier FFT	17
<b>3 Kvadratické časově-frekvenční distribuce a možnosti jejich využití pro spektrální analýzu ve vysokém rozlišení</b>	<b>19</b>
3.1 Kvadratické časově-frekvenční distribuce	19
3.2 Wignerova distribuční funkce	19
3.3 Variace WDF	21
3.4 Metody redukce interferenčních spektrálních složek	22
3.4.1 Vyhlazená Wignerova distribuční funkce (SWD,SPWD)	22
3.4.2 Filtrování v ambiguitní doméně	22
3.4.3 S-Metoda	23
3.4.4 B-distribuce	24
<b>4 Adaptivní metody vyhlazování kvadratických časově-frekvenčních distribucí</b>	<b>26</b>
4.1 Adaptivní a Vysoce adaptivní směrová TFD	26
4.2 Metoda výběru optimálního tvaru jádra pomocí algoritmu Region Growing	28
<b>5 Porovnání časově-frekvenčních analýz</b>	<b>30</b>
5.1 Spektrální přibližování	30
5.2 Porovnání metod redukce interferenčních spektrálních složek u WVD	30
5.2.1 Poznámky k výsledkům testování implementovaných metod	31
5.3 Úpravy a testování navržené metody AOK-RGA	33
5.3.1 Úpravy navržené metody	33
5.3.2 Výsledky testování	35
5.4 Hledisko teoretické výpočetní náročnosti	35
5.4.1 Komplexní demodulace	35

5.4.2	Kvadratické frekvenční distribuce . . . . .	37
5.5	Shrnutí výsledků první části práce . . . . .	37
5.5.1	Výběr metody pro implementaci . . . . .	37
<b>6</b>	<b>Implementace</b>	<b>39</b>
6.1	Algoritmus pro výpočet Wigner-Villeho distribuční funkce . . . . .	39
6.1.1	Knihovna FFTW . . . . .	39
6.1.2	Výpočet analytického signálu . . . . .	40
6.1.3	Výpočet Wignerovy distribuční funkce . . . . .	40
6.2	Implementace metody ADTFD . . . . .	40
6.2.1	Struktura a popis bloku . . . . .	41
6.2.2	Přímá implementace . . . . .	41
6.2.3	Využití paralelizace na procesoru s knihovnou Thread Building Blocks . . . . .	41
6.2.4	Rychlá 2D FFT konvoluce . . . . .	43
6.2.5	Implementace výpočtu na grafické kartě . . . . .	43
6.2.6	Separabilita kernelů a její využití při optimalizaci . . . . .	46
6.3	Implementace metody SPWD . . . . .	49
6.4	Testovací aplikace . . . . .	49
6.4.1	Problematika proměnlivé frekvenční nejistoty . . . . .	49
6.4.2	Měření a práce s testovací aplikací . . . . .	49
<b>7</b>	<b>Výsledky implementace</b>	<b>51</b>
7.1	Srovnání rychlostí konvolučních algoritmů v metodě ADTFD . . . . .	51
7.2	Testování implementovaných metod pro analýzu v reálném čase . . . . .	52
7.2.1	Výpočet Wignerovy distribuční funkce . . . . .	53
7.2.2	Výpočet vyhlazené WDF pomocí metody ADTFD . . . . .	53
7.2.3	Výpočet vyhlazené WDF pomocí metody SPWD . . . . .	54
7.3	Testovací konfigurace . . . . .	55
<b>8</b>	<b>Závěr</b>	<b>56</b>
	<b>Literatura</b>	<b>58</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>61</b>
	<b>Seznam příloh</b>	<b>62</b>
<b>A</b>	<b>Obsah elektronické přílohy</b>	<b>63</b>

# SEZNAM OBRÁZKŮ

1.1	Výpočet DFT délky $N$ pomocí dvou DFT poloviční délky. [2] . . . . .	15
1.2	Radix-2 FFT algoritmus. [2] . . . . .	15
2.1	Blokové schéma algoritmu komplexní demodulace [11]. . . . .	17
2.2	Znázornění algoritmu Vernier FFT. . . . .	18
3.1	WDF kombinace stacionárního signálu ( $f = 1500$ Hz) a lineárně modulovaného chirpu ( $f_0 = 3000$ Hz). . . . .	20
3.2	Ilustrace polohy frekvenčních složek pro A) SAF B) WDF . . . . .	23
4.1	Blokové schéma navrženého algoritmu AOK-RGA. . . . .	29
5.1	Demonstrace metody spektrálního přibližování aplikované na kombinaci logaritmických chirpů. . . . .	30
5.2	S-Metoda s různými hodnotami parametru $L_d$ aplikovaná na echolokační signál netopýra hnědého. . . . .	32
5.3	Výsledky vyhlazování WDF echolokačního signálu netopýra hnědého (a-c) a kombinace stacionárního signálu ( $f=2150$ Hz) a kvadraticky modulovaného ( $f_0=3000$ Hz) chirpu (d-f) pomocí B-distribuce a gaussovského filtru v ambiguitní doméně. . . . .	34
5.4	Výsledky vyhlazování WDF echolokačního signálu netopýra hnědého (a-c) a kombinace stacionárního signálu ( $f=1800$ Hz) a lineárně modulovaného ( $f_0=2000$ Hz) chirpu (d-f) pomocí AOK-RGA. . . . .	36
5.5	Výsledky porovnání vyhlazovací metody ADTFD a navržené metody AOK-RGA. . . . .	38
6.1	Struktura výpočetního bloku Wigner-Villeho distribuční funkce. . . . .	39
6.2	Implementovaný algoritmus pro výpočet ADTFD. . . . .	42
6.3	Struktura 2-D výpočetních operací v OpenCL. . . . .	47
6.4	Blokové schéma výpočtu směrově filtrovaného obrazu pomocí dílčích bází. . . . .	48
6.5	Demonstrační MATLAB aplikace. . . . .	50
7.1	Srovnání relativních zrychlení implementovaných algoritmů pro výpočet dílčích konvolucí v metodě ADTFD. . . . .	51
7.2	Testování real-time funkcionality bloku pro výpočet Wignerovy distribuční funkce. . . . .	52
7.3	Srovnání kritických a dosažených výpočetních časů metody ADTFD. . . . .	53
7.4	Testování real-time funkcionality bloku pro výpočet Wignerovy distribuční funkce . . . . .	54

## SEZNAM TABULEK

7.1	Specifikace testovací sestavy . . . . .	55
7.2	Parametry grafické karty ASUS Dual RX460 O2G . . . . .	55
7.3	Parametry procesoru AMD Ryzen 7 1700 . . . . .	55

## SEZNAM VÝPISŮ

6.1	Příklad OpenCL kernelu. . . . .	44
6.2	Příklad OpenCL kernelu. . . . .	46



# ÚVOD

Ačkoliv se běžná Rychlá Fourierova transformace může zdát ve většině případů jako plně dostačující nástroj pro spektrální analýzu nestacionárních signálů, existují vědní obory, které vyžadují vysoké rozlišení časově-frekvenční analýzy, kterého je možné dosáhnout pouze za použití specializovaných metod. Tato práce si klade za cíl představit tyto metody, implementovat je a zhodnotit jejich vlastnosti. Čtenář bude proveden touto oblastí zpracování signálů od Fourierovy transformace, přes metody spektrálního přibližování z ní přímo vycházejících až po oblast bilineárních časově-frekvenčních distribucí.

Před samotným ponořením se do definicí a vysvětlováním základních principů je však vhodné pozastavit se a zamyslet se nad vlastní motivací pro zavádění těchto specializovaných metod. Potřebu vysokého rozlišení časově-frekvenční analýzy můžeme pocítit například ve zdravotnictví při ultrazvukové analýze krevního oběhu. Dokonce i v hudebních odvětvích oboru zpracování signálů rostou požadavky na přesnost analýzy. Příkladem může být analýza hudebních témbřů pro přesnou re-syntézu reálných zvuků. Stejně tak můžeme nalézt podobné analyzátoři i ve strojírenství při zjišťování stavu součástí nebo v neposlední řadě v oborech biologie, při zkoumání tonálních posloupností zpěvu velryb či struktury echolokačních sonarových pulsu netopýřů velkých.

Je tedy zřejmé, že každý nástroj pro takovouto spektrální analýzu je ceněnou záležitostí v řadě vědních oborů. Obzvláště pokud je využitelný i pro analýzu v reálném čase. Posledním cílem této práce je tedy výběr jedné z představených metod a tu následně implementovat tak, aby ji bylo možné využít pro spektrální analýzu v reálném čase.

# 1 FOURIEROVA TRANSFORMACE

Fourierova transformace je známý matematický aparát pro převod signálů z časové oblasti do oblasti frekvenční – ať už v případě stacionárních signálů, kdy je aplikovaná přímo a nebo v případě signálů nestacionárních, kdy je používána její segmentální varianta – Krátkodobá Fourierova transformace. Díky poměrně snadnému a výpočetně nenáročnému způsobu implementace se tak stala ve své diskrétní podobě zřejmě nejpoužívanější metodou spektrální analýzy signálů. Pro případy, kdy je nutné výrazně zvýšit její rozlišení, však nedosahuje dobrých výsledků, neboť je výbornou ukázkou Heisenbergova principu neurčitosti vztaženého na čas a frekvenci. To tedy v praxi znamená, že pro zvýšení frekvenčního rozlišení je potřeba delší časový úsek analyzovaného signálu, čímž je zároveň snižováno rozlišení časové. Přesto však bude výchozím bodem této práce, ať už pro to, že některé později představené metody na ni přímo navazují v podobě její optimalizace pro vysoké rozlišení a nebo proto, že v některých algoritmech je využita jako matematický nástroj.

## 1.1 Definice

Fourierova transformace (FT) je jedním z fourierovských zobrazení, které je, narozdíl od Fourierovy řady, vztažené na signály jednorázové, někdy zobecněně označované jako signály aperiodické. Pro Fourierovu transformaci signálu definovaného ve spojitě časové oblasti platí vztah:

$$S(\omega) = \int_{-\infty}^{\infty} s(t)e^{-j\omega t} dt, \quad (1.1)$$

kde  $S(\omega)$  je spektrální funkce představující Fourierův obraz signálu  $s(t)$ . [1]

V diskrétní časové oblasti tak můžeme definovat diskrétní Fourierovu transformaci (DFT) signálu  $s(n)$  jako:

$$S(e^{j\omega}) = \sum_{-\infty}^{\infty} s[n]e^{-j\omega n}. \quad (1.2)$$

### 1.1.1 Krátkodobá Fourierova transformace

Při analýze nestacionárních (časově proměnlivých) signálů však nelze pouze aplikovat FT na celou délku signálu a očekávat kýžený výsledek v podobě časově-frekvenčního rozdělení. Pro tyto případy je nutné využít FT v její krátkodobé podobě (STFT). Ta vychází z předpokladu, že analyzovaný nestacionární signál je rozdělen na segmenty, které jsou považované za stacionární a nad těmito segmenty je aplikována FT. Segmenty jsou navíc váženy zvolenou funkcí časového okna pro

redukci negativních jevů jako je např. spektrální prosakování. STFT signálu  $s(t)$  můžeme tedy definovat ve spojitém čase vztahem:

$$STFT(\tau, \omega) = \int_{-\infty}^{\infty} s(t)w(t - \tau)e^{-j\omega t} dt. \quad (1.3)$$

Alternativně pro diskrétní signál  $s[n]$  pak můžeme psát:

$$STFT(n, \omega) = \sum_k w[k] s[n + k]e^{-j\omega k}. \quad (1.4)$$

Nejčastější reprezentací spektra časově variantního signálu je spektrogram – soubor spekter definovaný jako druhá mocnina modulu STFT daného signálu:

$$SPEC(n, k) = |STFT(n, k)|^2. \quad (1.5)$$

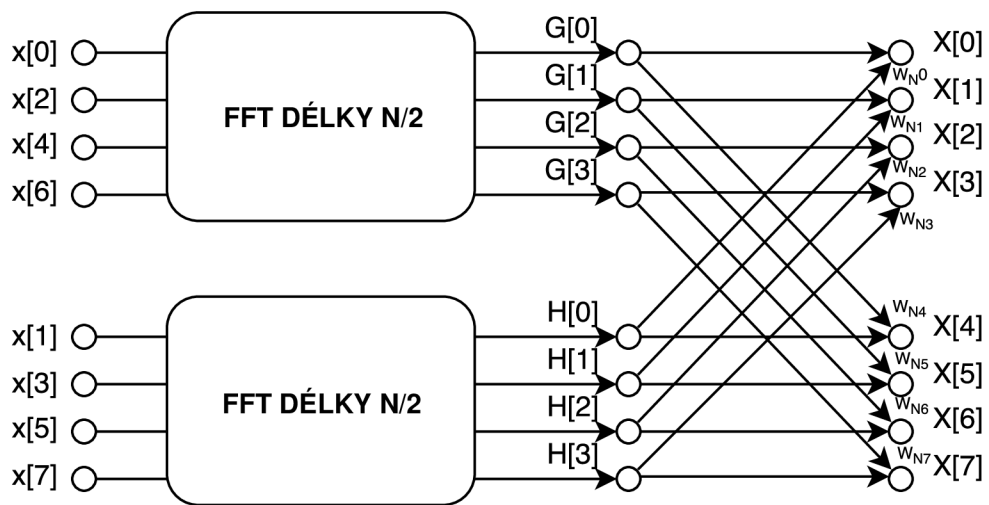
STFT a potažmo i spektrogram je možné zařadit do skupiny lineárních časově-frekvenčních distribucí.

### 1.1.2 Rychlá Fourierova transformace (FFT)

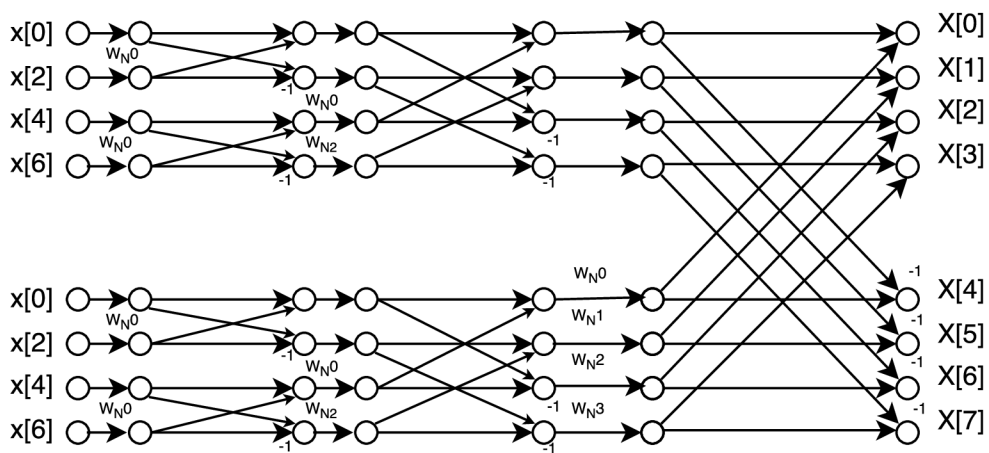
Jako FFT (Fast Fourier Transform) označujeme optimalizované algoritmy pro výpočet diskrétních Fourierových transformací. Tyto algoritmy obecně pracují na principu opakovaného využívání výsledků několika kratších mezivýpočtů. Princip metody ilustrujeme na nejjednoduším z rodiny FFT algoritmů – Radix-2 FFT. Vycházíme z předpokladu, že Fourierovu transformaci délky  $N$  můžeme vyjádřit pomocí dvou dílčích transformací délky  $\frac{N}{2}$  dle vztahu: [2]

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x(2n)e^{-i\frac{2\pi nk}{N/2}} + e^{i\frac{2\pi nk}{N}} \sum_{n=0}^{\frac{N}{2}-1} x(2n+1)e^{-i\frac{2\pi nk}{N/2}}, \quad (1.6)$$

kde vstupem jedné z transformací jsou tedy sudé vstupní vzorky a vstupem druhé jsou vzorky liché. Výraz  $e^{i\frac{2\pi nk}{N}}$  pak označujeme jako *twiddle faktor*  $W_t$ . Algoritmizaci tohoto vztahu můžeme ilustrovat pomocí schématu 1.1. Pokud tento princip aplikujeme rekurzivně, docházíme k algoritmu Radix-2 FFT znázorněnému schématem 1.2.



Obr. 1.1: Výpočet DFT délky N pomocí dvou DFT poloviční délky. [2]



Obr. 1.2: Radix-2 FFT algoritmus. [2]

## 2 METODY PRO SPEKTRÁLNÍ PŘIBLIŽOVÁNÍ ZALOŽENÉ NA FOURIEROVĚ TRANSFORMACI

Jelikož rozlišení FT analyzovaného signálu přímo závisí na délce analyzovaných dat, může (v závislosti na požadovaném rozlišení) taková analýza zároveň znamenat velmi vysokou výpočetní a paměťovou náročnost. Metody pro spektrální přibližování předpokládají, že v případě nutnosti analýzy spektra signálu ve vysokém rozlišení bývá oblastí zájmu pouze určité pásmo. Tohoto předpokladu je v těchto metodách různými způsoby využíváno k rapidnímu snížení výpočetní náročnosti při výrazném zvýšení rozlišení výsledku analýzy.

### 2.1 Komplexní demodulace

Tato metoda spočívá v modulaci pásma zájmu (tj. přibližované pásmo) o délce  $f_s/K$ , kde  $K$  je přibližovací faktor určující míru přiblížení, do základního pásma. Toho je docíleno komplexní modulací určenou vztahem: [10]

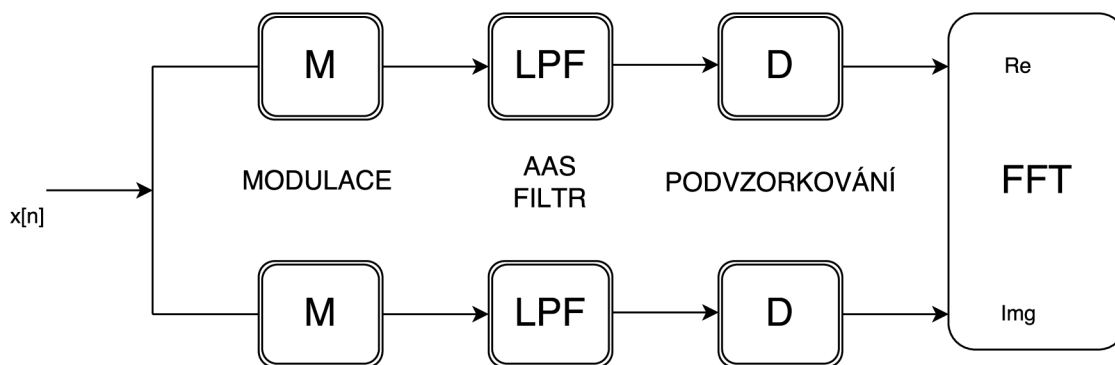
$$x_m[n] = x[n] e^{\frac{j2\pi f_0 n}{f_s}}. \quad (2.1)$$

Tím je přibližované pásmo přeneseno do základního pásma, kdy je střední frekvence pásma  $f_0$  modulována jako stejnosměrná složka signálu.

Signál je dále přiveden na vstup antialiasingového filtru a poté je podvzorkován faktorem  $K$ . Filtr musí mít v nepropustném pásmu útlum minimálně 110-120 dB na oktávu, aby bylo předcházeno nežádoucím vlivům aliasingu. [10]

Podvzorkováním je docíleno snížení redundance v analyzovaném signálu a je provedena jeho  $M$ -bodová rychlá Fourierova transformace, kdy  $M$  je  $K$ -krát menší než původní délka analyzovaného signálu.

Pro lepší představu o funkčnosti tohoto algoritmu uvedme následující příklad. Uvažujme signál se vzorkovací frekvencí 10 MHz, u kterého požadujeme analýzu pouze úzké části jeho spektra (uvažujme např. 1 kHz) avšak s dostatečným rozlišením mezi jednotlivými frekvenčními biny alespoň 10 Hz. Potom bychom při přímé spektrální analýze tohoto signálu potřebovali provést FFT přes 1 milion vzorků. Využijeme-li ale toho, že zbytek oblasti spektra kromě námi určeného pásma zájmu můžeme považovat za redundantní a využijeme algoritmus pro komplexní demodulaci s faktorem  $K = 4096$ , bude po translaci do základního pásma a podvzorkování potřeba FFT délky pouze 256 vzorků, což je samozřejmě výpočetně mnohem méně náročné. [11]



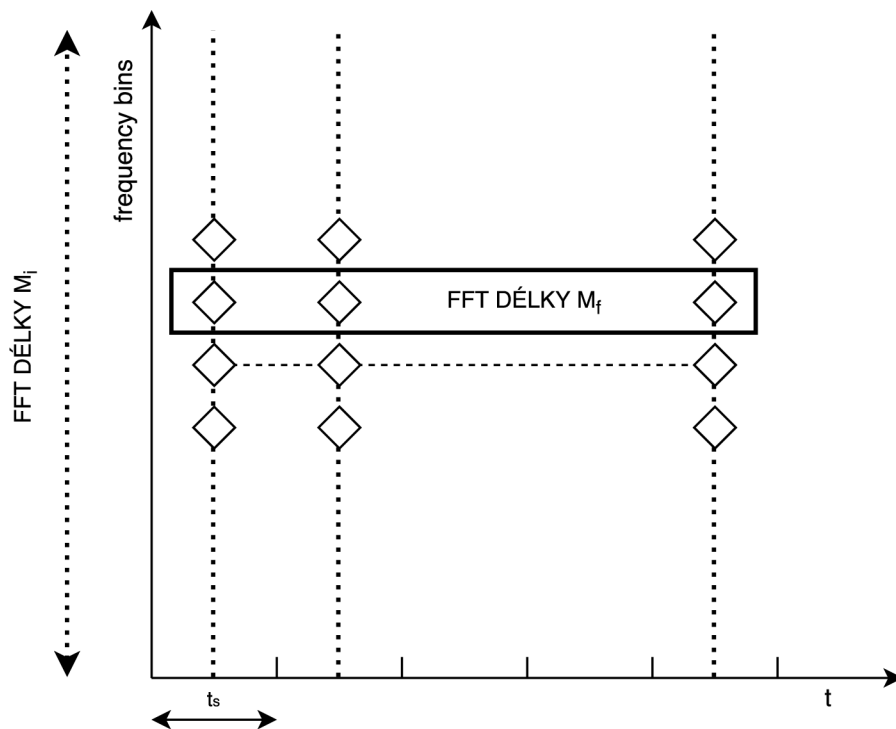
Obr. 2.1: Blokové schéma algoritmu komplexní demodulace [11].

Možný způsob reálné implementace znázorňuje blokové schéma v ilustraci 2.1.

## 2.2 Vernier FFT

Algoritmus označovaný jako Vernier FFT můžeme popsat těmito kroky: [3]

1. Vstupem je segmentovaný signál. Segmenty jsou váženy oknem.
2. Je provedena FFT každého segmentu.
3. Z každého spektra segmentu jsou vybrána ta data (frekvenční biny), které odpovídají zájmové oblasti analýzy.
4. Na tato data je ještě jednou aplikována FFT (tentokrát velmi krátké délky, neboť délka vstupu se rovná počtu segmentů v kroku 1).



Obr. 2.2: Znázornění algoritmu Vernier FFT.

Pro lepší představu uveďme příklad demonstrující princip metody. Uvažujme vstupní harmonický signál o frekvenci 256 Hz s nulovou fází. Ten je rozdělen kupříkladu na osm segmentů a na každý takový segment je aplikována FFT. Z FFT každého segmentu je zachována pouze hodnota frekvenčního binu lokalizovaného kolem 256 Hz. Po aplikaci FFT (s délkou 8) na tato data je získána Vernier FFT, kde nenulových hodnot nabývá pouze modul prvního (resp. nultého) frekvenčního binu. Pokud budeme uvažovat původní signál, nyní ovšem s frekvencí 256,5 Hz, budou mít frekvenční biny kolem 256 Hz a 257 Hz střídavou hodnotu amplitudy. Výstup Vernier FFT pak bude mít nenulovou hodnotu modulu pouze ve 4. frekvenčním binu (tedy uprostřed výstupního pole hodnot). Pro ostatní frekvenční biny bude situace podobná, jen hodnota modulu bude mít klesající tendenci v závislosti na vzdálenosti od binu odpovídajícímu 256 Hz. [3]

Přestože metoda demonstruje poměrně zajímavý způsob spektrálního přibližování, z důvodu výrazné redukce časového rozlišení způsobeného řetězením hodnot zájmových frekvenčních binů byla z fáze implementace a testování vyřazena a zástupcem metod pro spektrální přibližování byla zvolena pouze aktuálnější metoda komplexní demodulace.

# 3 KVADRATICKÉ ČASOVĚ-FREKVENČNÍ DISTRIBUCE A MOŽNOSTI JEJICH VYUŽITÍ PRO SPEKTRÁLNÍ ANALÝZU VE VYSOKÉM ROZLIŠENÍ

## 3.1 Kvadratické časově-frekvenční distribuce

Krátkodobá Fourierova transformace aplikovaná jako nástroj pro analýzu nestacionárního signálu vychází z předpokladu, že takovýto signál je možné chápat jako sumu stacionárních signálů o krátké době trvání. Základním nedostatkem tohoto přístupu je fakt, že frekvenční rozlišení je přímo ovlivněné dobou trvání těchto stacionárních segmentů, které samozřejmě určují i časové rozlišení výsledného časově-frekvenčního rozdělení [4]. Tento nedostatek obcházejí kvadratické časově-frekvenční distribuce navazující na práci E.Wignera, který v roce 1932 zavádí Wignerovu distribuční funkci, původně jako nástroj pro řešení matematických problémů kvantové mechaniky.

Bilineární (kvadratické) časově-frekvenční distribuce od té doby nacházejí uplatnění při řešení problémů v řadě vědeckých oborů od strojírenství, přes zdravotnictví či například v oblasti rozpoznávání objektů na základě dat získaných z radarových senzorů. [9]

Tato kapitola se věnuje nastínění základní problematiky aplikace těchto distribucí v oboru spektrální analýzy nestacionárních signálů a navrhuje možnosti řešení jejich negativních vlastností.

## 3.2 Wignerova distribuční funkce

Wigner definuje Wignerovu distribuční funkci (WDF) jako:

$$W(t, \omega) = \int R_t(\tau) e^{-j\omega\tau} d\tau, \quad (3.1)$$

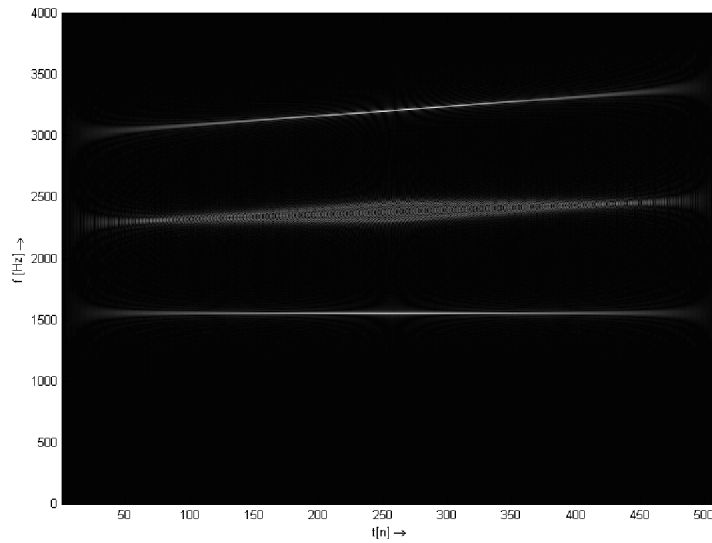
kde  $R_t(\tau)$  je lokální autokorelační funkce signálu  $s(t)$ :

$$R_t(\tau) = s\left(t + \frac{\tau}{2}\right) s^*\left(t - \frac{\tau}{2}\right). \quad (3.2)$$

Algoritmus pro výpočet WDF v její diskrétní podobě je pak definován vztahem: [4]

$$W(m\Delta t, k\Delta\omega) = \Re \left\{ 2\Delta t \text{FFT} \left[ \hat{R}(i) \right] \right\}, \quad (3.3)$$





Obr. 3.1: WDF kombinace stacionárního signálu ( $f = 1500$  Hz) a lineárně modulovaného chirpu ( $f_0 = 3000$  Hz).

s diskrétní lokální autokorelační funkcí:

$$\hat{R}(i) = s(m + i - 1)s^*(m - i + 1), \quad m \geq i. \quad (3.4)$$

#### Vlastnosti WDF [4]

- WDF je reálná funkce.
- Integrace WDF vzhledem k frekvenci odpovídá okamžitému výkonu signálu.
- Integrace WDF vzhledem k času odpovídá výkonové spektrální hustotě signálu.
- Časový nebo frekvenční posun signálu odpovídá stejnému posunu ve WDF.
- WDF je pro daný signál symetrická v čase.
- WDF může nabývat i záporných hodnot.
- Integrál druhé mocniny WDF odpovídá druhé mocnině integrálu výkonu signálu vzhledem k času.

Jak může být patrné z ilustrace 3.1, výsledné časově-frekvenční rozdělení trpí na přítomnost interferenčních spektrálních složek, které jsou nejvíce markantní při

aplikaci WDF na multikomponentní signál. Tato negativní vlastnost pramení z kvadratické povahy WDF a možnosti její eliminace či alespoň významné redukce budou předvedeny v několika následujících kapitolách.

### 3.3 Variace WDF

#### Wigner-Villeho distribuční funkce (WVD)

Pro zlepšení vlastností WDF včetně potlačení interferencí v oblasti základního pásma zavádíme modifikaci původní distribuční funkce. Metoda spočívá v předzpracování analyzovaného signálu pomocí Hilbertovy transformace: [4]

$$s_a(t) = s(t) + j\mathcal{H}\{s(t)\}. \quad (3.5)$$

Hilbertova transformace je definována jako:

$$\mathcal{H}\{s(t)\} = s(t) * h(t), \quad (3.6)$$

kde konvoluční jádro odpovídá:

$$h(t) = \frac{2 \sin^2(\frac{\pi t}{2})}{\pi t}. \quad (3.7)$$

V diskrétní podobě můžeme Hilbertovu transformaci zapsat jako:

$$\mathcal{H}\{s_r[n]\} = \sum_{m=-\infty}^{\infty} h[n-m]s_r[m]. \quad (3.8)$$

Výsledkem předzpracování je pak tedy analytický signál neobsahující záporné frekvenční komponenty. Ten je poté přiveden na vstup Wignerovy d.f. a tím je získána Wigner-Villeho d.f. [4]

#### Pseudo Wignerova distribuční funkce (pWDF)

V případě analýzy reálných signálů konečné délky, kdy není možné integrovat v rozsahu nekonečných integračních mezí, zavádíme metodu známou jako Pseudo Wignerova distribuce [14]. Ta spočívá v omezení WDF časovým oknem  $w$  a je definována jako:

$$W_p(t, f) = \int_{-\infty}^{\infty} R_w(\tau) R_t(\tau) e^{-j2\omega\tau f} d\tau, \quad (3.9)$$

kde  $R_w$  označuje autokorelační funkci časového okna  $w$ :

$$R_w(\tau) = w\left(\frac{\tau}{2}\right)w^*\left(-\frac{\tau}{2}\right). \quad (3.10)$$

V oblasti diskrétního času můžeme tedy pWDF definovat vztahem: [14].

$$W_p(n, \omega) = 2 \sum_{k=-L+1}^{L-1} R_w[k] R_t[k, n] e^{-j2k\omega}, \quad (3.11)$$

kde  $R_w$  nyní označuje diskrétní autokorelační funkci časového okna:

$$R_w[k] = w[k] w * [-k]. \quad (3.12)$$

### 3.4 Metody redukce interferenčních spektrálních složek

Přítomnost interferencí plynoucích z kvadratické povahy WDF výrazně omezuje možnosti reálného využití této metody pro spektrální analýzu složitějších signálů. Obzvláště v případě signálů s dvěma a více spektrálními složkami, kdy pozorovateli navíc není předem známý odhad rozložení vlastních spektrálních složek, se může výsledná matice časově-frekvenční distribuce stát naprosto nečitelnou. Pro eliminaci či alespoň minimalizaci vlivu nežádoucích interferenčních složek proto zavádíme metody založené na různých typech filtrování základní WDF popř. WVD.

#### 3.4.1 Vyhlazená Wignerova distribuční funkce (SWD, SPWD)

Nejvíce přímočarým přístupem k redukování vlivu interferenčních spektrálních složek ve WDF je filtrování distribuce Gaussovským 2D průměrovacím filtrem. Tento filtr můžeme definovat jako:

$$G(t, \omega) = \frac{1}{2\pi\sigma_t\sigma_\omega} e^{-\left[\left(\frac{t^2}{2\sigma_t^2}\right) + \left(\frac{\omega^2}{2\sigma_\omega^2}\right)\right]}, \quad (3.13)$$

kde  $\sigma_t, \sigma_\omega$  je rozptyl okna v odpovídajících směrech. Vyhlazenou WDF pak můžeme zapsat pro spojitou časovou oblast vztahem:

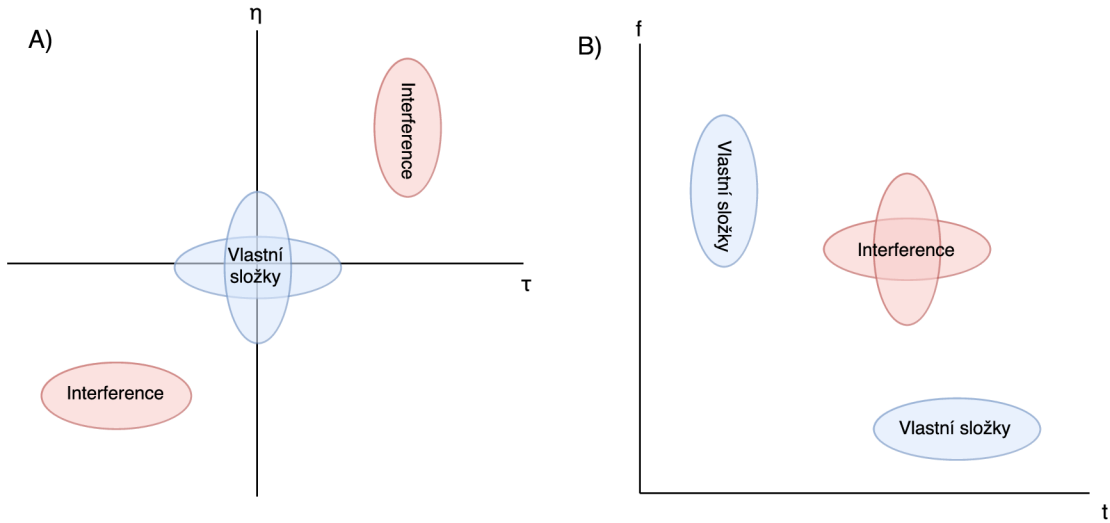
$$W_s(t, \omega) = \frac{1}{2\pi} \int \int W(t', \omega') G(t - t', \omega - \omega') dt' d\omega'. \quad (3.14)$$

SWD diskrétního signálu pak definujeme jako:

$$W_s[l, m] = \frac{\Delta t \Delta \omega}{2\pi} \sum_{p=l-j}^{l+j} \sum_{q=m-k}^{m+k} W[p, q] G[p - l, q - m]. \quad (3.15)$$

#### 3.4.2 Filtrování v ambiguitní doméně

Odlíšný přístup odfiltrování interferenčních složek nabízí metody založené na filtraci v ambiguitní (dvoznačné) doméně. Definujme Sussmanovu ambiguitní funkci



Obr. 3.2: Ilustrace polohy frekvenčních složek pro A) SAF B) WDF

(SAF) vztahem:

$$A(\theta, \tau) = \int_{-\infty}^{+\infty} f(t + \frac{\tau}{2})f^*(t - \frac{\tau}{2})e^{i\theta t} dt. \quad (3.16)$$

Tuto funkci můžeme také získat přímo z Wignerovy distribuční funkce vztahem: [7]

$$A(\theta, \tau) = \mathcal{F}_t \mathcal{F}_f^{-1} \{W(t, f)\}. \quad (3.17)$$

Jak můžeme pozorovat na ilustraci 3.2, v ambiguitní doméně dochází ke koncentraci vlastních spektrálních složek ve středu matice ambiguitní funkce, zatímco interferenční složky jsou lokalizovány po stranách. Tato separace umožňuje poměrně snadno a efektivně odfiltrovat interference vynásobením matice ambiguitní funkce maskou v podobě filtračního jádra.

### 3.4.3 S-Metoda

Další metoda pro eliminaci či redukci interferenčních složek vychází ze vztahu mezi STFT analyzovaného signálu a jeho PWD funkcí: [8]

$$W_p(t, \Omega) = \frac{1}{\pi} \int_{-\infty}^{\infty} STFT(t, \Omega + \theta)STFT^*(t, \Omega - \theta)d\theta. \quad (3.18)$$

Časově-frekvenční distribuce, která je označovaná podle jejího autora L.J. Stankovice jako S-Metoda, je pak definována vztahem: [8]

$$W_{SM}(t, \Omega) = \frac{1}{\pi} \int_{-L_d}^{L_d} P(\theta) STFT(t, \Omega + \theta)STFT^*(t, \Omega - \theta)d\theta, \quad (3.19)$$

kde  $P(\Omega)$  je konečné obdélníkové okno o délce  $2L_d$  definované ve frekvenční oblasti. V diskrétní časové oblasti pak můžeme definovat S-Metodu jako: [9]

$$W_{SM}[n, k] = \sum_{i=-L_d}^{L_d} P[i] STFT[n, k + i] STFT^*[n, k - i]. \quad (3.20)$$

Z předchozího vztahu je patrné, že pro  $L_d = 0$  je výstupem S-Metody spektrogram analyzovaného signálu. Zjednodušeně lze tedy princip této metody chápat jako „obohacování“ běžného spektrogramu o jistou míru Wignerovy distribuce, kde tato míra je určena velikostí okna  $L_d$

S-Metoda mimo svůj zjevný přínos v podobě kontrolovatelné redukce interferenčních složek přináší i zajímavý pohled na vztah mezi lineárními a kvadratickými TDF, kdy postupným inkrementováním délky obdelníkového okna  $L_d$  můžeme pozorovat plynulý přechod od spektrogramu analyzovaného signálu až po jeho Wignerovu distribuční funkci.

### 3.4.4 B-distribuce

Obecná definice jakékoliv bilineární časově frekvenční distribuce je generalizována vztahem: [5]

$$P(t, \omega) = W(t, \omega) * \phi(t, \omega). \quad (3.21)$$

Z tohoto vztahu tedy plyne, že jakoukoliv bilineární distribuci můžeme získat pomocí filtrování Wignerovy distribuce  $W(t, \omega)$  daným filtračním jádrem  $\phi(t, \omega)$  označovaným také jako jádro časově -frekvenční distribuce.

Pokud vyjdeme z poznatků přednesených v kapitole 3.5.2, můžeme takovouto distribuci definovat v ambiguitní doméně: [5]

$$C(\theta, \tau) = A(\theta, \tau) \phi(\theta, \tau). \quad (3.22)$$

V závislosti na tvaru jádra  $g(t, \tau)$  rozlišujeme celou řadu těchto pokročilejších č.f. distribucí. Pro demonstraci rozdílu mezi takto speciálně filtrovanou WDF a původní WDF byla v této práci vybrána tzv. B-distribuce jejíž jádro je určeno vztahem: [7]

$$G(t, \tau) = |\tau|^\beta \cosh^{-2\beta} t, \quad (3.23)$$

kde rozptyl propustné oblasti filtračního jádra je určen parametrem  $\beta$ . Stejně jako v předchozích případech je možné pozorovat, že s větší mírou filtrace (tzn. užším jádrem filtru) sice vzrůstá míra potlačení nežádoucích interferenčních složek, ale

zároveň se zhoršuje i frekvenční rozlišení jednotlivých vlastních spektrálních složek analyzovaného signálu.

## 4 ADAPTIVNÍ METODY VYHLAZOVÁNÍ KVADRATICKÝCH ČASOVĚ-FREKVENČNÍCH DISTRIBUCÍ

Z poznatků z předchozích kapitol můžeme snadno vypožorovat, že schopnost jednotlivých metod redukovat přítomnost interferenčních spektrálních složek je přímo vztahovaná k charakteru analyzovaného signálu. Při implementaci jsme pak tedy v podstatě postaveni před dvě možnosti nastavení parametrů použité vyhlazovací metody. Budto zvolíme parametry tak, aby metoda podávala co nejlepší výkon pro úzce specializovanou množinu signálů, a nebo zvolíme kompromis a získáme univerzálnější spektrální analyzátor, který však nebude dosahovat ideálních výsledků. V úvahu však připadá i třetí možnost a to využít algoritmu pro adaptivní změnu parametrů vyhlazovacích filtrů na základě vlastností vstupního signálu.

### 4.1 Adaptivní a Vysoce adaptivní směrová TFD

Ačkoliv těchto metod opět existuje větší množství, v této práci bude představena metoda Adaptivní směrové TFD (ADTFD) a na ni navazující metoda Vysoce adaptivní TFD (HADTFD). Obě tyto metody fungují na v předchozích kapitolách představeném principu filtrování časově-frekvenční distribuce speciálním 2D filtrem  $\gamma_\theta(t, f)$ . Výslednou TFD můžeme tedy definovat jako:

$$TFD_{adapt}(t, f) = TFD(t, f) *_t *_f \gamma_\theta(t, f), \quad (4.1)$$

kde  $TFD(t, f)$  označuje kvadratickou časově-frekvenční distribuci získanou z WVD konvolucí s 2D vyhlazovacím jádrem  $\gamma(t, f)$ :

$$TFD(t, f) = WVD(t, f) *_t *_f \gamma(t, f). \quad (4.2)$$

Pokud dosadíme za  $\gamma(t, f)$  všepropustné filtrační jádro, bude výchozím bodem adaptivní metody Wigner-Villeho distribuce. [7]

Filtrační jádro  $\gamma_\theta(t, f)$  je určeno druhou derivací dvojrozměrného Gaussovského okna. Hlavním důvodem pro použití tohoto specifického filtru je fakt, že mimo rozptylu v obou osách je možné parametrizovat i natočení filtru a tím docílit přesnějších výsledků. Vztah pro výpočet jádra  $\gamma_\theta(t, f)$  je: [13]

$$\gamma_\theta(t, f) = \frac{ab}{2\pi} \frac{d^2}{df_\theta^2} e^{-a^2 t_\theta^2 - b^2 f_\theta^2}, \quad (4.3)$$

kde parametry  $a$ ,  $b$  určují rozptyl filtru ve směru příslušných os a  $t_\theta$ ,  $f_\theta$  určují úhel  $\Theta$  natočení jádra filtru vztahy [13] :

$$t_\theta = t \cos(\theta) + f \sin(\theta), \quad f_\theta = -t \sin(\theta) + f \cos(\theta). \quad (4.4)$$

Princip optimalizace natočení vyhlazovacího filtračního jádra  $\gamma_\theta(t, f)$  je založen na výběru takového úhlu  $\Theta$ , pro který je korelace směrového jádra filtru a modulu TFD maximální [12] :

$$\theta(t, f) = \arg \max_{\theta} | | TFD(t, f) | *_t *_f \gamma_\theta(t, f) | . \quad (4.5)$$

### Vysoce adaptivní TFD

Vysoce adaptivní TFD metoda navíc optimalizuje i parametry  $a$ ,  $b$ , tedy rozptyl v příslušných osách filtračního jádra. Toho je docíleno sérií operací, kdy je nejprve vypočítána množina  $TFD_i$  filtrovaných TFD:

$$TFD_i(t, f) = TFD(t, f) *_t *_f \gamma_{a_i, b_i, \theta}(t, f), \quad (4.6)$$

kde  $i$  je proměnná jdoucí od 1 do L. Konstanta L určuje počet možných variací parametrů  $a_i$ ,  $b_i$  volených tak, aby se vzrůstající hodnotou  $i$  rostla i míra efektu vyhlazování dané TFD. [13]

Následně je pro každé dvě po sobě jdoucí filtrované TFD z množiny  $TFD_i$  vypočítána hodnota  $\Delta TFD_i$  jako rozdíl:

$$\Delta TFD_i(t, f) = TFD_{i+1}(t, f) - TFD_i(t, f). \quad (4.7)$$

Parametry vedoucí k nejmenšímu rozdílu mezi dvěma filtrovanými TFD jsou pak vybrány jako optimální rozptyly vyhlazovacího filtru pro dané  $t, f$  [13] :

$$(a, b) = \arg \min_{\theta} | \Delta TFD_i(t, f) | . \quad (4.8)$$

Výsledný algoritmus pak můžeme zapsat v několika postupných krocích:

### Adaptivní TFD

1. Výpočet WVD analyzovaného signálu.
2. Získání výchozí kvadratické TFD filtrováním WVD (v případě, že výchozím č.f. rozdělením má být zvolena WVD, je tento krok vynechán).
3. Konvoluce modulu TFD s L různě natočenými filtračními jádry  $\gamma_\theta$ .
4. Výběr optimálních hodnot úhlu natočení  $\Theta(t, f)$ .



## Vysoce Adaptivní TFD

5. Výpočet množiny distribucí  $TFD_i$ .
6. Výpočet hodnot  $\Delta TFD_i$ .
7. Výběr optimálních parametrů  $a, b$ .

## 4.2 Metoda výběru optimálního tvaru jádra pomocí algoritmu Region Growing

V rámci této práce byla navržena i metoda adaptivního filtrování interferenčních složek založená na principu výběru optimálního tvaru maskovacího jádra pro filtrování signálu v ambiguitní doméně. Pro účely této práce bude metoda nadále nazývána jako AOK-RGA (Adaptive Optimal Kernel by Region Growing Algorithm).

Vycházejme z kapitoly 3.4.2 a předpokladu, že vlastní spektrální složky signálu jsou v ambiguitní doméně lokalizovány ve středu matice AF. Narozdíl od předchozích představených metod není tvar filtračního jádra definován předem a až následně upravován pomocí adaptace rozptylu či natočení, ale je vytvářen přímo výběrem oblasti výskytu vlastních spektrálních složek.

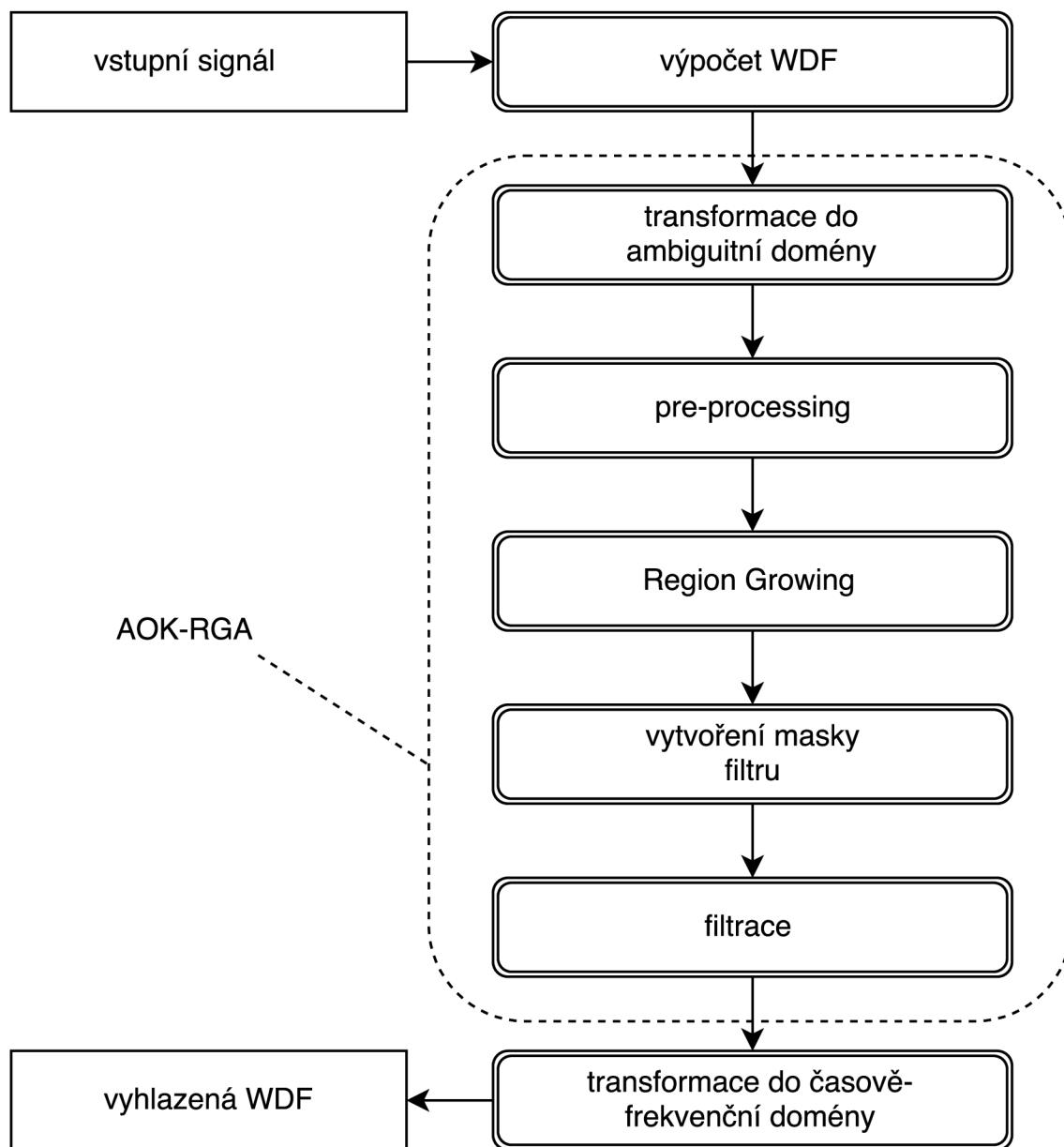
Tento výběr je realizován pomocí algoritmu Region Growing s počáteční pozicí nastavenou do středu matice AF. Tento výchozí bod označme jako *seed*. Algoritmus Region Growing je poměrně běžným nástrojem oboru zpracování obrazu a funguje na principu porovnávání intenzit pixelů (v našem případě hodnotu modulu  $AF$  v daném bodě  $n, k$ ) v iterativně posouváném okolí výchozího *seed* bodu. [6] Rozhodovací úroveň pro zastavení posunu okolí je určena hodnotou prahového parametru  $T$ . Můžeme tedy říci, že:

$$M(n, k) = \begin{cases} 1, & \text{pro } |AF(n, k)| > T \\ 0, & \text{pro } |AF(n, k)| < T \end{cases},$$

kde  $M(n, k)$  je binární maska vymežující danou oblast. [6]

Ideálním výstupem algoritmu je binární maska, která je nulová pro všechny oblasti, ve kterých se nevyskytují vlastní spektrální složky analyzovaného signálu. Tato maska pak tvoří optimální tvar jádra vyhlazovacího filtru  $\gamma_{\text{opt}}$ .

Použití nijak nezpracované binární masky přímo jako jádra vyhlazovacího filtru však nepřináší nejlepší možné výsledky právě kvůli binárnímu charakteru jejích hodnot (tj. ostrých přechodů mezi oblastí definovaného regionu a oblastí výskytu interferenčních složek).



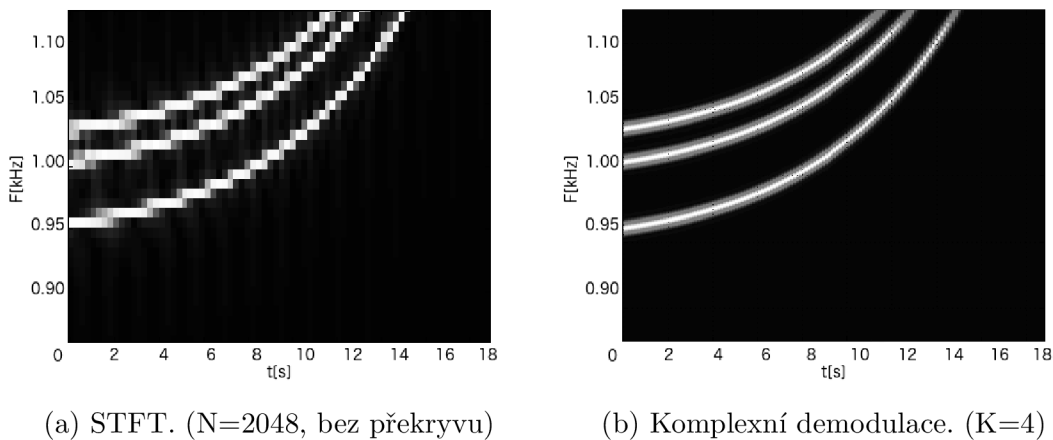
Obr. 4.1: Blokové schéma navrženého algoritmu AOK-RGA.

## 5 POROVNÁNÍ ČASOVĚ-FREKVENČNÍCH ANALÝZ

Po úvodní rešerši byly představené metody implementovány pomocí skriptovacího jazyka aplikace Matlab a následně byly zkoumány jejich vlastnosti s cílem vybrat metodu pro pozdější implementaci v jazyce C++. Mimo samotného zkoumání vlastností zadaných metod byl kladen důraz zejména na studium metod pro redukcí interferenčních spektrálních složek kvadratických časově-frekvenčních distribucí.

### 5.1 Spektrální přibližování

Metoda komplexní demodulace pro spektrální přibližování vykazuje, jak je možné pozorovat na ilustraci 5.1, velmi dobré výsledky, co se týče zvýšení frekvenčního rozlišení. Nicméně neřeší problém s časovým rozlišením, čímž se např. pro analýzu velmi krátkých signálů, narozdíl od Wignerovy distribuce a jejích mutací, nehodí. Dalším faktorem, který je třeba zvážit, je fakt, že tato metoda umožňuje analýzu pouze přesně vymezeného pásma o šířce definované faktorem přiblížení  $K$ .



Obr. 5.1: Demonstrace metody spektrálního přibližování aplikované na kombinaci logaritmických chirpů.

### 5.2 Porovnání metod redukce interferenčních spektrálních složek u WVD

Jednotlivé metody byly v průběhu implementace a testování aplikovány na celou řadu syntetických testovacích signálů. Pro zachování přehlednosti a stručnosti jsou

však v příložených ilustracích doloženy spektrogramy pouze těch signálů, které dobře demonstrierují negativní i pozitivní vlastnosti představených metod.

### 5.2.1 Poznámky k výsledkům testování implementovaných metod

#### SPWD

SPWD vykazuje poměrně dobrou redukci interferencí v oblastech inflexí vlastních spektrálních složek nelineárně modulovaných signálů. Interferenční složky mezi dvěma signály však redukuje minimálně. Vzhledem k poměrně jednoduché implementaci a relativní výpočetní nenáročnosti je možné zvážit použití tohoto druhu vyhlazování pro vyhlazování v reálném čase.

#### S-Metoda

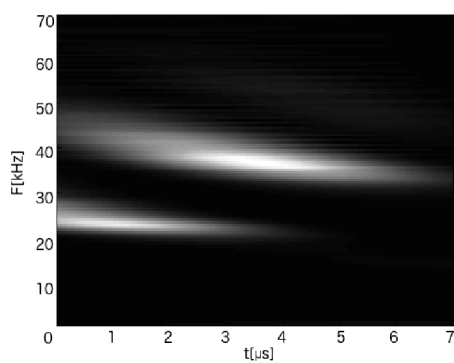
S-metoda nachází své uplatnění obzvláště při redukci vlivu spektrálního prosakování klasické STFT aplikované na signál definovaný v krátkém časovém úseku. Zvolením délky okna  $L_d$  lze dosáhnout dobrého kompromisu mezi rozlišením výsledné distribuce a přítomností interferenčních spektrálních složek. Bohužel však tato metoda není vhodná pro implementaci v reálném čase, neboť již z její definice je patrné, že pro výpočet S-Metody je nutné nejprve spočítat STFT daného signálu. Díky tomu, že základem výpočtu je STFT, je také možné pozorovat silnou nepřímou úměru mezi časovým a frekvenčním rozlišením. Ilustrace 5.2 ukazuje vliv hodnoty parametru  $L_d$  – se zvyšující se hodnotou  $L_d$  se výsledná časově-frekvenční distribuce kvalitativně blíží Wignerově distribuční funkci (včetně přítomnosti interferenčních složek).

#### Radiální filtr v ambiguitní doméně

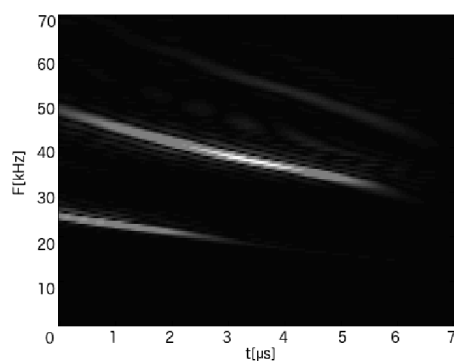
Tato metoda se vyznačuje poměrně konstantní mírou kvality produkovaných TFD pro různé druhy signálů, nicméně toto je, jak může být patrné z ilustrací 5.3c a 5.3f, vykoupeno výrazným vlivem filtrování na přesnost definice spektra signálů. Lepších výsledků by bylo možné dosáhnout použitím eliptického 2D filtru – tedy Gaussovského filtru s kontrolovatelným rozptylem ve směru obou os.

#### B-distribuce

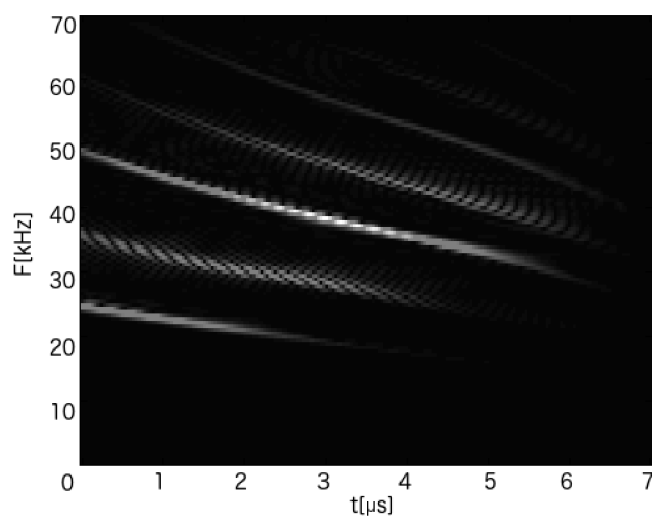
Z příložených ilustrací 5.3b a 5.3e je patrné, že ačkoliv B-distribuce vykazuje poměrně dobré výsledky pro syntetické testovací signály, pro reálný signál echolokačního pulsu netopýra nejenže neredukuje interferenční složky, ale také výrazně znehodnocuje výslednou TFD. Po bližší analýze AF testovaného signálu bylo vypořazováno, že



(a) Spektrogram ( $L_d=0$ )



(b)  $L_d=4$



(c)  $L_d=14$

Obr. 5.2: S-Metoda s různými hodnotami parametru  $L_d$  aplikovaná na echolokační signál netopýra hnědého.

tento jev byl způsoben opačnou orientací modulu AF echolokačního signálu oproti jádru B-distribuce, které ve své základní definici parametrizovaný náklon (rotaci kernelu) neumožňuje. Překryv AF signálu filtračním jádrem byl tedy nedokonalý. V případě, že bylo jádro filtru nakloněno pomocí funkce *imrotate* v prostředí aplikace Matlab, bylo již dosaženo výsledků podstatně lepších. Z tohoto problému lze usoudit, že B-distribuce sama o sobě sice dokáže produkovat velmi kvalitní vyhlazené TFD, ale její výkon je silně spřažený s charakterem spektra analyzovaného signálu a pro její optimální využití by bylo třeba parametrizovat i náklon jejího jádra, který by byl navíc adaptivně přizpůsobován vstupnímu signálu.

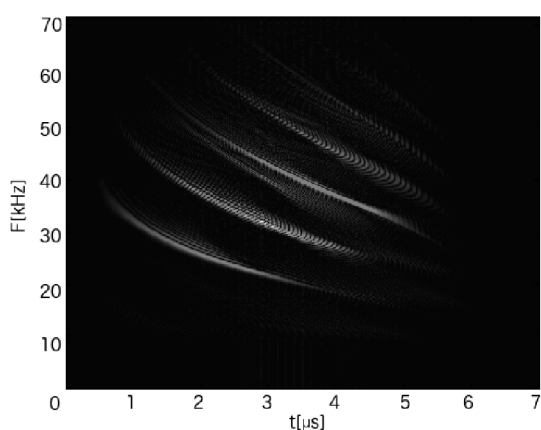
## 5.3 Úpravy a testování navržené metody AOK-RGA

### 5.3.1 Úpravy navržené metody

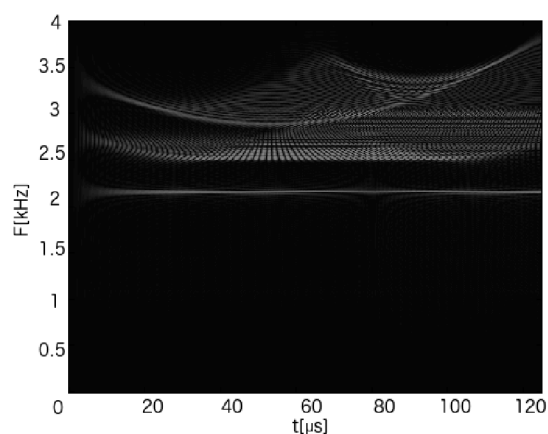
Jak již bylo řečeno v kapitole představující základní princip navržené metody, při přímém použití výstupní matice RG algoritmu jako maskovacího jádra  $\gamma_{\text{opt}}$  AF funkce analyzovaného signálu není dosaženo nejlepších možných výsledků – výstupní TFD je znehodnocena přítomností šumu. Bylo zjištěno, že výsledky dosahují mnohem vyšší kvality, pokud jsou ostré přechody binární masky před samotnou filtrací vyhlazeny. Toho je v tomto případě dosaženo velmi jednoduchým způsobem a to rozostřením masky pomocí 2D filtru (Matlab funkce *imfilter* s parametrem *disk* ).

Testováním bylo vyzorováno, že pokud se podobným způsobem předzpracuje i modul AF před samotnou aplikací RG algoritmu, je dosahováno díky zjemnění dynamického rozsahu AF lépe definovaných tvarů maskovací funkce  $M$ . Míra vyhlazení však musí být výrazně nižší než v případě diskutovaném v předchozím odstavci, neboť je nutné zachovat separaci vlastních a interferenčních spektrálních složek analyzovaného signálu.

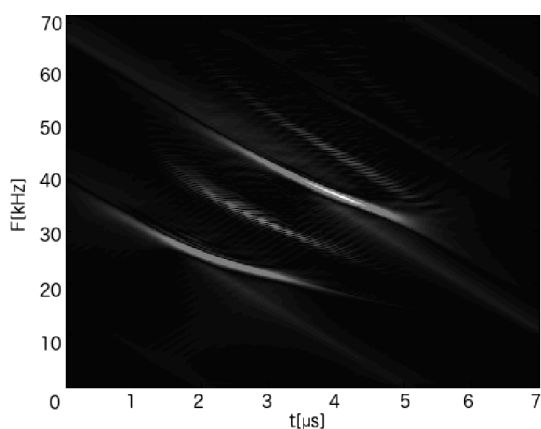
Druhým problémem, který byl při testování funkčnosti algoritmu odhalen, je neschopnost redukce interferenčních složek pro dvě stacionární harmonické složky lokalizované ve frekvenční doméně velmi blízko u sebe. Prvním problémem je fakt, že kombinace dvou stacionárních složek produkuje nespojitou AF v oblasti vlastních spektrálních složek. Nespojitost je do jisté míry redukována vyhlazením AF zmíněném v předchozím odstavci, nicméně pokud zvolíme takovou míru vyhlazení, abychom byli schopni eliminovat nespojitost v oblasti vlastních složek, téměř vždy tím zákonitě zasáhneme i do separace vlastních a interferenčních složek AF. Funkce metody je také silně spřažena s nastavením prahovacího parametru  $T$  RG algoritmu.



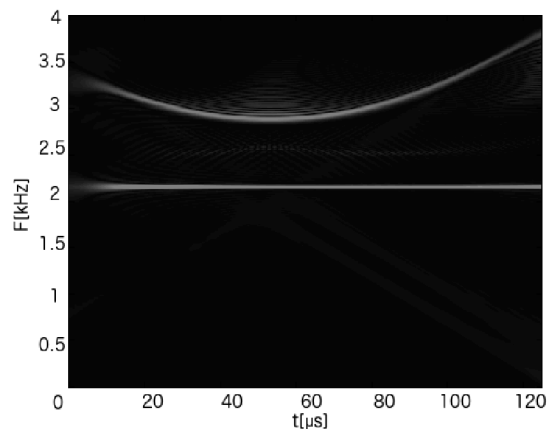
(a) WVD echolokačního signálu.



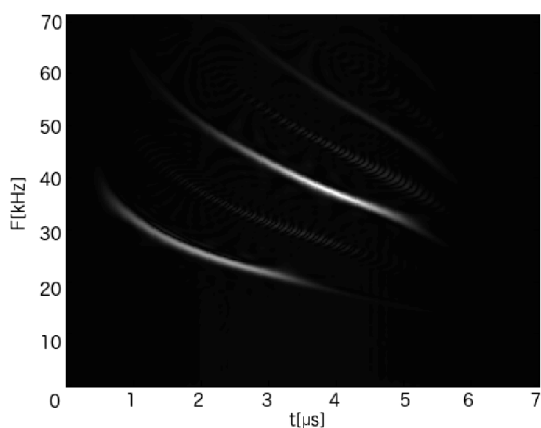
(d) WVD kombinace signálů.



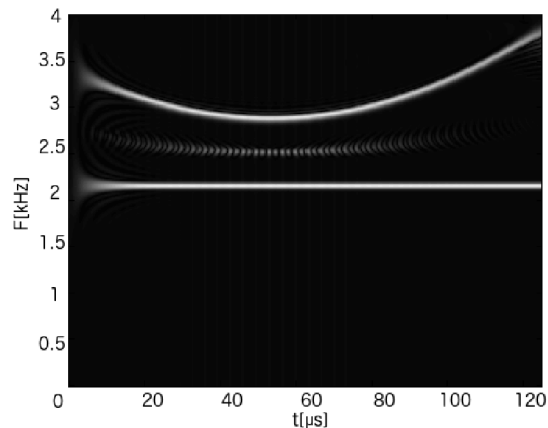
(b) B-distribuce ( $\beta = 0.01$ ).



(e) B-distribuce ( $\beta = 0.01$ ).



(c) Radiální filtr v amb. doméně.



(f) Radiální filtr v amb. doméně.

Obr. 5.3: Výsledky vyhlazování WDF echolokačního signálu netopýra hnědého (a-c) a kombinace stacionárního signálu ( $f=2150$  Hz) a kvadraticky modulovaného ( $f_0=3000$  Hz) chirpu (d-f) pomocí B-distribuce a gaussovského filtru v ambiguitní doméně.

### 5.3.2 Výsledky testování

V ilustraci 5.5 je možné pozorovat jak TFD před a po filtrování, tak i odpovídající tvar filtračního jádra získaný algoritmem RG. Dále bylo provedeno srovnání s implementovanou metodou ADTFD. Obě metody byly aplikovány na množinu testovacích signálů sestávající z:

- **Syntetické signály (512 vzorků,  $f_s = 8.0$  kHz):**
  1. multikomponentní harmonický signál ( $f_1 = 200$  Hz,  $f_2 = 500$  Hz,  $f_3 = 1000$  Hz,  $f_4 = 1500$  Hz,  $f_5 = 2000$  Hz,  $f_6 = 3000$  Hz),
  2. kombinace stacionárního a FM signálu ( $f_1 = 700$  Hz,  $f_c = 3000$  Hz,  $f_m = 12$  Hz),
  3. dva blízké lineárně modulované signály ( $f_{10} = 800$  Hz,  $f_{11} = 1800$  Hz,  $f_{20} = 1000$  Hz,  $f_{21} = 2000$  Hz).
- **Reálné signály (400 vzorků,  $f_s = 142.0$  kHz):**
  4. echolokační signál netopýra hnědého (tři nelineární chirpy klesající z frekvencí cca 65, 60 and 40 kHz).

Výsledky získané pomocí metody AOK-RGA byly v rámci množiny signálů srovnatelné s výsledky metody ADTFD. V některých případech byla redukce interferenčních složek dokonce více markantní. V multikomponentním signálu ADTFD nedosahovala míry redukce algoritmu AOK-RGA ani s většími velikostmi filtračního kernelu. U spektrogramu echolokačního signálu bylo v porovnání s ADTFD dosaženo opět objektivně lepších výsledků, nicméně metoda ADTFD vykazovala vyšší míru redukce “vlastních interferenčních složek” lokalizovaných v oblastech zakřivení jednotlivých chirpů.

V případě dvou blízkých signálů byla, vzhledem k nedostatečné separaci v ambiguitní doméně, pozorována buď nulová redukce interferencí, nebo v případě vyšší hodnoty prahovacího parametru, výrazné zhoršení frekvenčního rozlišení.

## 5.4 Hledisko teoretické výpočetní náročnosti

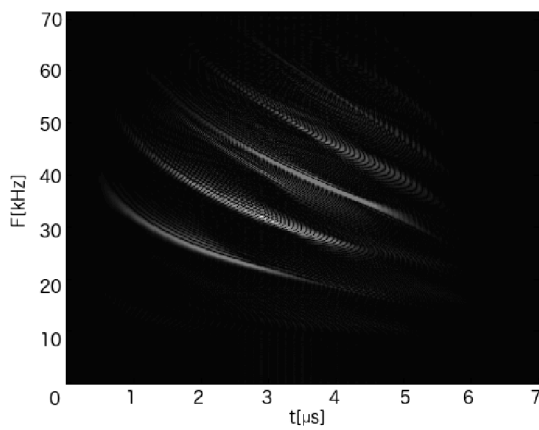
### 5.4.1 Komplexní demodulace

Teoretický počet operací násobení potřebných pro realizaci této metody pro segment signálu o velikosti  $N$  je možné definovat jako:

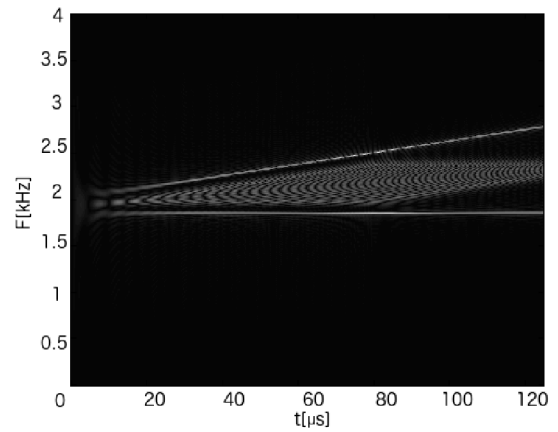
$$4NKF + 4NK + 2N \log_2 N, \quad (5.1)$$

kde  $N$  je délka analyzovaného signálu,  $K$  je faktor přiblížení a  $F$  je řád antialiasingového filtru. [10]

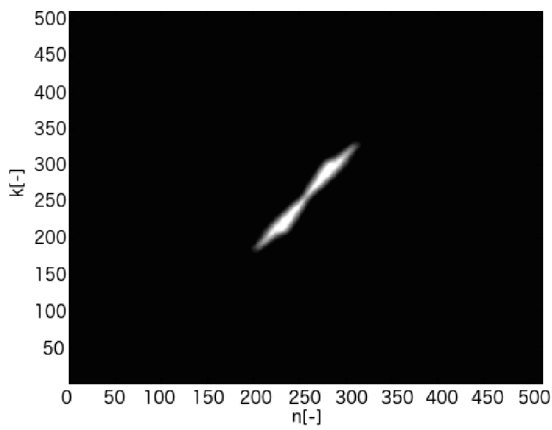




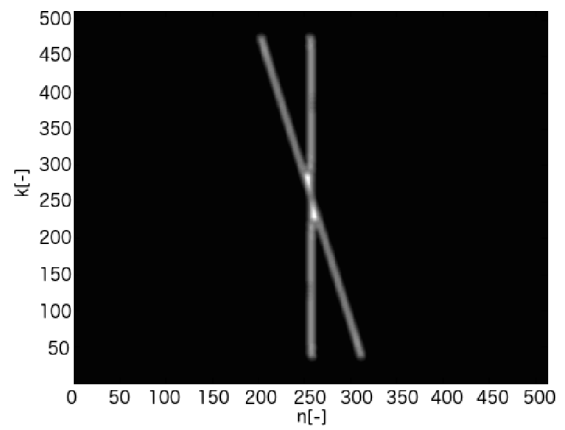
(a) WVD echolokačního signálu.



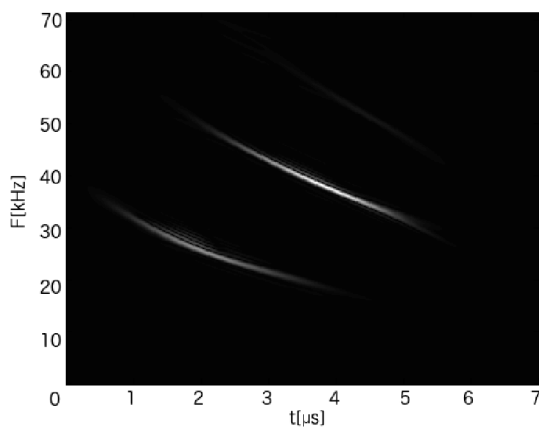
(d) WVD kombinace signálů.



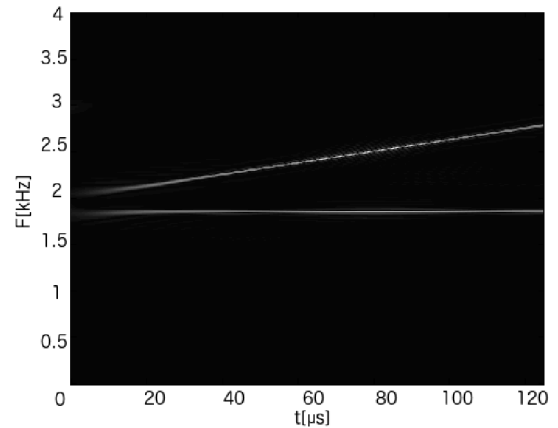
(b) Tvar vyhlazovacího jádra AOK-RGA



(e) Tvar vyhlazovacího jádra AOK-RGA.



(c) Filtrovaná WVD.



(f) Filtrovaná WVD.

Obr. 5.4: Výsledky vyhlazování WDF echolokačního signálu netopýra hnědého (a-c) a kombinace stacionárního signálu ( $f=1800$  Hz) a lineárně modulovaného ( $f_0=2000$  Hz) chirpu (d-f) pomocí AOK-RGA.

## 5.4.2 Kvadratické frekvenční distribuce

Již ze samotné teoretické podstaty metod odvozených od WDF je patrné, že počet operací provedený při výpočtu kvadratické TFD je výrazně větší než při výpočtu běžné STFT. Pokud budeme uvažovat počet nutných operací jako součet operací násobení a sčítání a zároveň se zaměříme na počet provedených DFT (které přispívají k celkové náročnosti největším dílem), můžeme definovat výpočetní náročnost výpočtu WVD jako:

$$12N_h^2 \log_2 N, \quad (5.2)$$

kde  $N$  je délka analyzovaného signálu (segmentu) a  $N_h$  odpovídá  $2N$ . [17]

V případě existujících adaptivních metod vyhlazování výpočetní náročnost strmě vzrůstá, protože je třeba spočítat navíc i sadu konvolucí pro výpočet ideálního natočení a rozptylu jádra filtru. Počet realizovaných konvolucí tak závisí na zvolené míře kvantizace (tj. na kroku natočení filtru atd.).

Výpočetní náročnost představeného algoritmu AOK-RGA je těžce definovatelná, neboť závisí na spektrálním charakteru signálu a tedy i velikosti oblasti vlastních spektrálních složek v ambiguitní doméně. V případě dobře separovaných vlastních a interferenčních spektrálních složek však bylo dosahováno dobrých výsledků.

## 5.5 Shrnutí výsledků první části práce

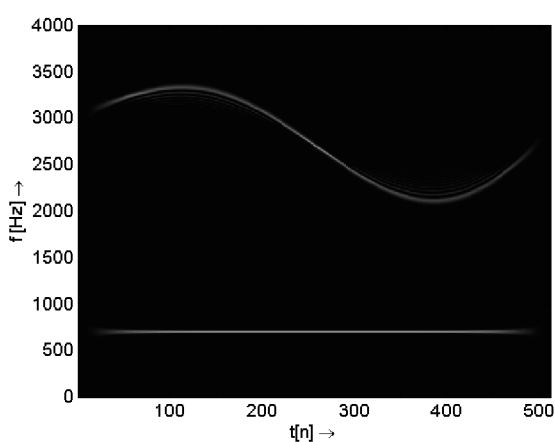
### 5.5.1 Výběr metody pro implementaci

Metoda komplexní demodulace představuje praktickou, poměrně jednoduše implementovatelnou metodu pro redukci výpočetní náročnosti spektrální analýzy ve vysokém rozlišení. Toho je docíleno přiblížením omezeného pásma celkového spektra analyzovaného signálu. Představuje tedy spíše optimalizační úlohu a přínos případné implementace by byl minimální. Oproti tomu oblast kvadratických časově-frekvenčních distribucí nabízí mimo vyššího frekvenčního rozlišení i výrazné zjemnění časového rozlišení a tedy i značnou redukci efektu Heisenbergova principu neurčitosti. Přesto, že tyto distribuce přinášejí i řadu problémů, které by mohly komplikovat jejich implementaci (vyšší výpočetní náročnost, přítomnost interferenčních spektrálních složek), byla pro další fázi práce, vzhledem k uvedeným kladným vlastnostem a obecnému přínosu implementace, zvolena Wignerova distribuční funkce coby zástupce kvadratických časově-frekvenčních distribucí.

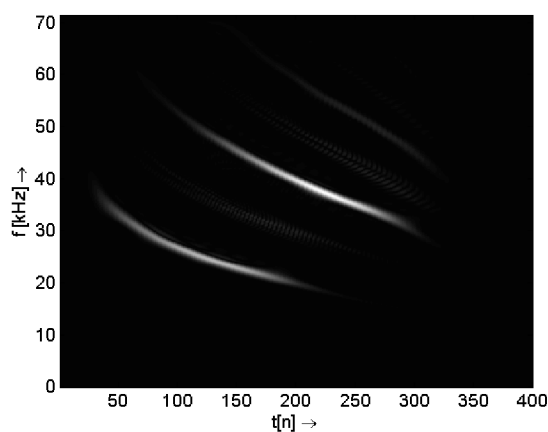
## Výběr metody pro redukci vlivu interferenčních složek

Přestože navržená metoda AOK-RGA vykazovala dobré výsledky na testovací množině signálů a to i v přímém porovnání s metodou ADTFD, její efektivita v případě aplikace na komplexnější signály je sporná. Metoda ADTFD navíc kromě přínosu řešení optimalizačních úloh souvisejících s její implementací představuje i výchozí bod pro případné rozšíření na metodu HADTFD, která v [13] vykazovala velmi vysokou adaptibilitu a míru redukce interferenčních složek. Z těchto důvodů byla pro další fázi práce, implementaci v jazyce C++, zvolena metoda ADTFD.

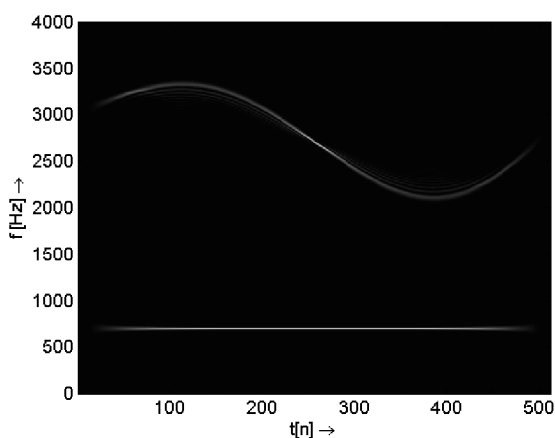
Jako zástupce neadaptivních metod byla zvolena metoda SPWD, zejména z důvodu výrazně nižší výpočetní náročnosti.



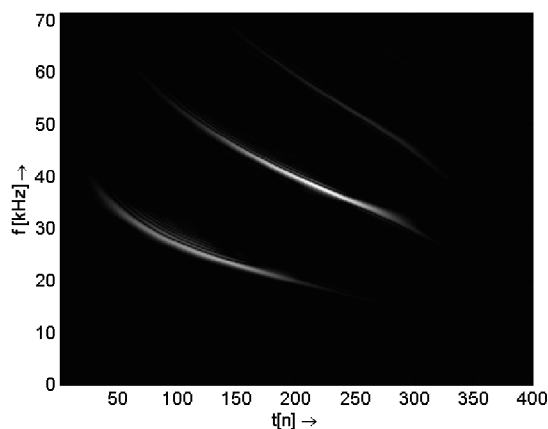
(a) ADTFD (Signál 2).



(c) ADTFD (Signál 4).



(b) AOK-RGA (Signál 2).



(d) AOK-RGA (Signál 4).

Obr. 5.5: Výsledky porovnání vyhlazovací metody ADTFD a navržené metody AOK-RGA.

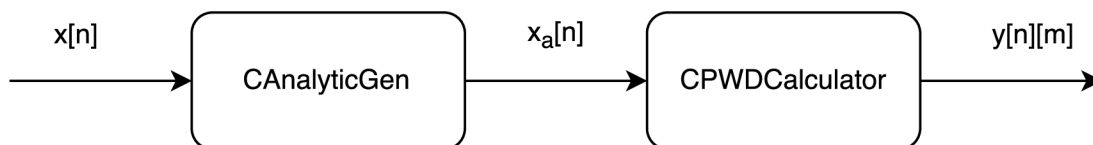
## 6 IMPLEMENTACE

Tato kapitola se věnuje samotné implementaci knihovny pro výpočet vyhlazené Wignerovy distribuční funkce. Vzhledem k důrazu na rychlost výpočtu a optimalizacím, nutným obzvláště v pozdější fázi práce, kdy byl implementován vyhlazovací algoritmus ADTFD, bylo jazykem pro implementaci zvoleno C++. Vývoj probíhal v prostředí Visual Studio 2015 IDE s využitím řady knihoven, které budou představeny chronologicky podle místa výskytu v jednotlivých výpočetních blocích.

Hlavní algoritmus je strukturován pomocí objektů reprezentujících jednotlivé výpočetní bloky. Základní struktura tříd je děděna pomocí rozhraní `IProcessingBlock` předepisujícího společné a opakující se vlastnosti výpočetních bloků (názvy výpočetních metod, proměnné atd.)

### 6.1 Algoritmus pro výpočet Wigner-Villeho distribuční funkce

Jádro knihovny tvoří blok výpočtu Wignerovy distribuční funkce (tedy takové WDF, ve které jsou přítomny interferenční spektrální složky). Strukturu bloku můžeme ilustrovat pomocí schématu 6.1 a dílčích subbloků, které budou dále popsány podrobněji.



Obr. 6.1: Struktura výpočetního bloku Wigner-Villeho distribuční funkce.

#### 6.1.1 Knihovna FFTW

Základní operací používanou na několika místech implementovaného algoritmu je DFT. Návrh vlastního algoritmu pro výpočet Rychlé Fourierovy transformace zasahuje hluboko mimo problematiku, které se tato práce věnuje, a proto bylo nutné zvolit vhodnou knihovnu, na kterou by bylo možné tyto výpočty delegovat. K tomuto účelu byla vybrána knihovna FFTW. Tato knihovna je vytvořena pomocí jazyka C a mimo FFT výpočtů nabízí i řadu dalších funkcí, jako je například vlastní reprezentace komplexních čísel (datový typ `fftw_complex`) či možnosti multithreadingu. Samotný výpočet FFT probíhá pomocí tzv. plánů, které jsou po inicializaci dle aktuální potřeby vykonány pomocí metody `fftw_execute_plan` [19].

### 6.1.2 Výpočet analytického signálu

Jak je patrné z kapitoly 3.3, pro získání Wigner-Villeho distribuční funkce je třeba nejprve získat analytický signál z analyzovaného segmentu vzorků. Ten lze získat přímou implementací Hillbertovy transformace podle vztahu 3.8. V praxi však požadujeme, kromě jednoduchosti implementace, i co nejméně časově náročný algoritmus. Vyjděme z všeobecné definice analytického signálu, jakožto signálu neobsahujícího záporné spektrální komponenty:

$$S_a(f) = \begin{cases} 2S(f), & \text{pro } f > 0, \\ S(f), & \text{pro } f = 0, \\ 0, & \text{pro } f < 0, \end{cases} \quad (6.1)$$

kde  $S_a(f)$  je Fourierův obraz analytického signálu a  $S(f)$  je obraz původního, vstupního signálu.

Klíčovou operací výpočtu analytického signálu je tedy odfiltrování záporných spektrálních složek. Jedním ze způsobů jak získat signál splňující uvedenou definici je použití Diskrétní Fourierovy transformace pro převod do frekvenční oblasti, následné vynulování frekvenčních binů odpovídajících zápornému spektru a zpětná transformace zpět do časové oblasti.

Při konkrétní implementaci bylo pro dopřednou a zpětnou DFT využito knihovny FFTW. Za výpočet analytického signálu zodpovídá instance třídy `CAlyticCalculator`, kde samotný výpočet probíhá v metodě `process` přijímající za argumenty ukazatel na paměť vstupního bufferu, výstupního bufferu a hodnotu reprezentující počet vzorků v jednom bufferu.

### 6.1.3 Výpočet Wignerovy distribuční funkce

Výpočet Wignerovy distribuční funkce byl realizován přímou implementací vztahu 3.3. K výpočtu DFT byla opět využita knihovna FFTW. V bloku `PWDCalculator` je nejprve vypočítána matice lokální autokorelační funkce vstupního signálu a na tuto je následně aplikována po řádcích FFT (`fftw_plan_many_dft`). Výsledná WDF je pak zapsána do výstupního bufferu a předána dalšímu bloku řetězce.

## 6.2 Implementace metody ADTFD

Implementace adaptivního vyhlazovacího algoritmu ADTFD představuje značnou výzvu, zejména v důsledku předpokládané vysoké výpočetní náročnosti související s nutností provádět  $L$  2-D filtrací (kde  $L$  reprezentuje počet různých natočení derivací gaussovského filtru a tedy přímo ovlivňuje přesnost vyhlazovací metody) pro každý

analyzovaný blok. V této kapitole bude nastíněný postup implementace s důrazem na popis jednotlivých optimalizovaných algoritmů, které byly při snaze dosáhnout kritického výpočetního času navrženy. Bylo navrženo několik algoritmů, které se liší metodou použitou pro výpočet dílčích konvolucí nebo použitou metodou paralelizace výpočtů. Hlavní důraz je tedy kladen na co nejvyšší míru optimalizace výpočtu dílčích konvolucí.

### 6.2.1 Struktura a popis bloku

Výpočet metody ATFD byl opět zapouzdřen do nezávislého objektu třídy `CADTFDCalculator` (popř. `CADTFDCalculatorCL`, pro algoritmus využívající funkci knihovny `OpenCL`). Jednotlivé algoritmy lze volit při inicializaci objektu nastavením hodnoty enumerátoru `Algorithm`. Objekt je v blokovém schématu zařazen přímo za blok `PWVDCalculator` a metoda `process` tedy přijímá jako vstupní argument ukazatel na blok paměti velikosti  $2N \times N$  a ukazatel na výstupní buffer stejné velikosti.

### 6.2.2 Přímá implementace

Výchozím bodem byla přímá implementace diskrétní 2D konvoluce dle vzorce:

$$(I * f)(x, y) = \sum_{-\infty}^{+\infty} \sum_{-\infty}^{+\infty} I(x - u, x - v) f(u, v), \quad (6.2)$$

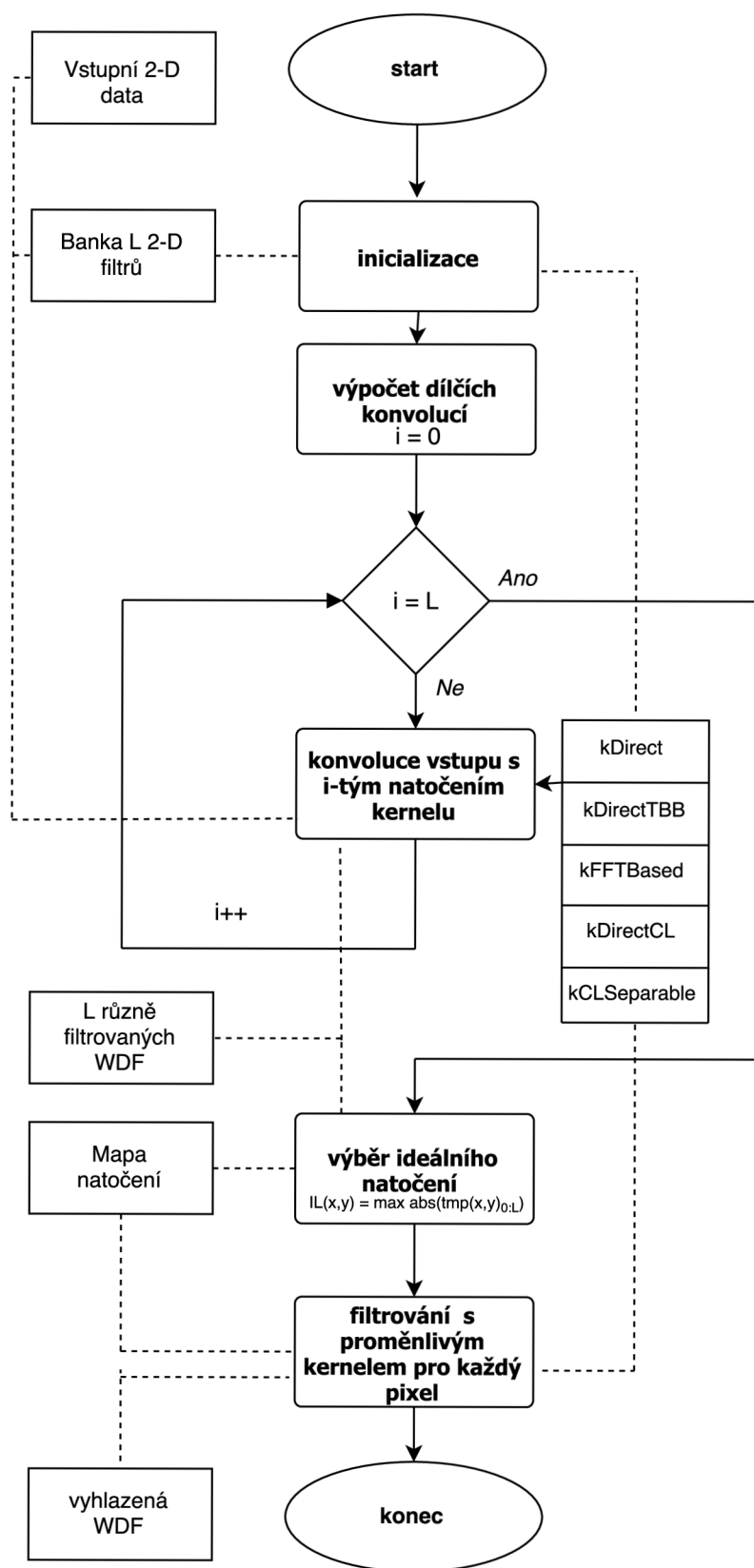
kde počet vykonaných operací pro vstupní data rozměru  $N \times M$  a filtrem velikosti  $K$  je přibližně:

$$NMK^2. \quad (6.3)$$

Tato metoda je tedy prostá jakýchkoliv optimalizací a slouží jako výchozí bod pro měření relativního zrychlení v porovnání s ostatními algoritmy v části práce s diskuzí výsledků. Výběr tohoto algoritmu je možné provést nastavením hodnoty `kDirect` proměnné `Algorithm`.

### 6.2.3 Využití paralelizace na procesoru s knihovnou Thread Building Blocks

Využití multijádrových kapacit výpočetních jednotek patří k základním úkonům při optimalizaci algoritmů. V této práci je paralelizace výpočtů na CPU zastoupena využitím paralelních smyček knihovny `Thread Building Blocks` (TBB) společnosti Intel. Výhodou této knihovny je poměrně jednoduchá možnost využití potenciálu multijádrových CPU, např.: pomocí funkce `parallel_for`, která umožňuje rozdělení dílčích kroků mezi jednotlivá jádra podle nastaveného blokového rozsahu `blocked_range`



Obr. 6.2: Implementovaný algoritmus pro výpočet ADTFD.

[20]. Využitím této funkce lze tedy teoreticky dosáhnout lepších výpočetních časů než v případě použití jednoduchých, neoptimalizovaných smyček.

### 6.2.4 Rychlá 2D FFT konvoluce

Odlíšným přístupem je výpočet filtrovaného obrazu ve frekvenční oblasti. Zde se využívá zejména faktu, že operace konvoluce je ve frekvenční doméně převedena na operaci běžného násobení. V úvahu však musíme také vzít cenu převodu do frekvenční domény (FFT) a také cenu převodu zpětného (IFFT). Metodu výpočtu pro obraz velikosti  $N \times N$  a velikosti filtru  $K \times K$  je možné popsat následujícími kroky:

1. doplnění matice kernelu na velikosti  $(N + k) \times (N + k)$ ,
2. 2-D FFT vstupního obrazu,
3. 2-D FFT matice z kroku 1,
4. komplexní násobení matic z kroku 1 a 2,
5. inverzní FFT produktu multiplikace.

Algoritmus rychlé konvoluce implementovaný v této práci minimalizuje i výpočetní redundanci např. výpočtem spekter kernelů již při inicializaci aplikace. Výsledná výpočetní náročnost je pak tedy dána pouze jedním výpočtem spektra vstupního obrazu,  $L$  operacemi komplexního násobení a  $L$  zpětnými FFT transformacemi. Pro výpočet FFT je použita opět knihovna FFTW.

### 6.2.5 Implementace výpočtu na grafické kartě

Vzhledem k tomu, že problematika vyhlazování kvadratických časově frekvenčních distribucí nás zavádí od zpracování jednorozměrných signálů k oblasti zpracování obrazových dat, je poměrně logickou úvahou delegovat dílčí operace na výpočetní jádro grafické karty.

#### Knihovna OpenCL

Množství knihoven umožňující přístup k výpočetnímu hardwaru grafických karet je označováno pojmem “vendor specific”, jedná se tedy o knihovny, které spolupracují pouze s kartami daného výrobce (např. knihovna NVidia CUDA). V této práci však bylo požadováno co nejuniverzálnější řešení a proto byla zvolena knihovna Intel OpenCL. Tato knihovna nabízí jak možnosti výpočetní paralelizace na CPU, tak na grafických jednotkách většiny výrobců. Delegace výpočtu pak probíhá prostřednictvím tzv. kernelů, které jsou vytvářeny programovacím jazykem založeným na jazyce



C (OpenCL C). Ty je možné načítat z binárních souborů nebo je kompilovat přímo ze zdrojových kódů za běhu hostitelské aplikace.

Inicializace knihovny OpenCL spočívá ve výběru dostupné OpenCL platformy a zařízení, na které budou výpočtu delegovány. Výčet dostupných zařízení je možné filtrovat pomocí příslušných flagů. Dále je nutné, jak již bylo zmíněno, zkompilovat použité kernely a vytvořit objekty typu `Kernel` pro jejich reprezentaci na straně hostitelské aplikace. Při požadavku na použití daného kernelu je pak jeho objektu nastaven příslušný počet parametrů (metoda `setKernelArg`) a kernel je zařazen do fronty (`enqueueNDRangeKernel`).

```
const sampler_t sampler = CLK_NORMALIZED_COORDS_FALSE |
    CLK_ADDRESS_CLAMP | CLK_FILTER_NEAREST;

void kernel convolve(
    __read_only image2d_t input,
    __global float *filter_kernel,
    int filter_kernel_size,
    __write_only image2d_t output)
{

    const int2 index = {get_global_id(0), get_global_id(1)};
    float sum = 0.f;

    int dim = get_global_size(0);

    for (int i = -filter_kernel_size / 2; i <
        filter_kernel_size / 2; i++)
    {
        for (int j = -filter_kernel_size / 2; j <
            filter_kernel_size / 2; j++)
        {
            int2 coordAct = index+(int2)(i,j);

            float val = read_imagef(input,sampler,coordAct)
                .x;
            int kernelPosY = i+(filter_kernel_size/2);
            int kernelPosX = j+(filter_kernel_size/2);
```

```

        sum+=val*filter_kernel[kernelPosY + kernelPosX*
            filter_kernel_size];

    }

}

float val = sum;
write_imagef(output,index,val);
}

```

Výpis 6.1: Příklad OpenCL kernelu.

**Základní koncepty práce s OpenCL [21]** Ačkoliv si práce záměrně neklade za cíl interpretovat celou dokumentaci knihovny OpenCL, je nutné vysvětlit alespoň základní pojmy související s vykonáváním výpočetních kernelů zmíněných v odstavci výše.

- **Výpočetní jednotka (Computing unit)**  
OpenCL zařízení má jednu či více výpočetních jednotek. Na nich se vykonává vždy jedna pracovní skupina. Výpočetní jednotka je složena z jednoho či více výpočetních elementů a lokální paměti.
- **Zařízení (Device)**  
Označení pro soubor výpočetních jednotek.
- **Příkazová fronta (Command queue)**  
Fronta použitá pro předávání příkazů OpenCL zařízení. Mezi tyto příkazy patří např. požadavek na vykonání výpočetního kernelu nebo čtení a zápis do paměťových objektů.
- **Pracovní jednotka (Work item)**  
Jednotka ze souboru paralelních volání výpočetního kernelu na daném zařízení. Pracovní jednotka je explicitně určena pomocí globálního a lokálního ID.
- **Pracovní skupina (Work group)**  
Soubor pracovních jednotek. Jednotky v jedné skupině vykonávají volání jednoho kernelu a sdílejí lokální paměť.

Důležitým parametrem je i velikost pracovní skupiny (`local_work_size`), jejíž hodnota ovlivňuje celkový výpočetní čas. Pro lepší ilustraci struktury OpenCL výpočtu je možné nahlédnout do ilustrace 6.3.

## Problematika lokální a globální paměti

Při delegování výpočtů na grafickou kartu se často setkáváme s pojmy globální a lokální paměť. Přístupy z CL zařízení do globálního paměťového prostoru představují časově složitější operaci oproti přístupům do paměti lokální, sdílené v rámci pracovní skupiny. Proto je vhodné minimalizovat počet přístupů do globální paměti, obzvláště v případech, kdy se jedná o přístupy nadbytečné.

Uvažujme například, že výpočetnímu kernelu dvourozměrné konvoluce předáváme ukazatel na paměťový blok s hodnotami jádra 2-D filtru o velikosti  $K \times K$ . Pak každá výpočetní jednotka pracovní skupiny počítající konvoluční sumu pro jeden určitý pixel musí provést také  $K \times K$  přístupů do globálního paměťového prostoru, kde jsou uloženy koeficienty filtru.

Pokud však vhodně využijeme paralelity výpočetních jednotek, je možné využít lokální paměť zařízení jako mezipaměť, do které je před zahájením samotné výpočetní operace uložen obsah globálního paměťového bloku s hodnotami filtru. Toho můžeme dosáhnout poměrně přímočarým způsobem, kdy využíváme hodnot `local_id` pro indexaci jednotlivých hodnot v paměti v kombinaci s mechanismem lokálních bariér `CLK_LOCAL_MEM_FENCE`. V OpenCL však existuje i mechanismus přímo řešící problematiku asynchroního kopírování dat v rámci pracovní skupiny pomocí metody `async_work_group_copy` [22]. Výpis 6.2 ilustruje použití této metody. Funkce `wait_group_events` zde zajišťuje synchronizaci mezi výpočetními jednotkami (exekuce kódu po příkazu je pozdržena do doby, kdy je kopírování paměti dokončeno).

```
__local float K[24*24];
event_t e = async_work_group_copy((__local float*)K, (
    __global float*)&filter_kernel[0], (size_t)
    filter_kernel_size*filter_kernel_size, 0);
wait_group_events(1, &e);
```

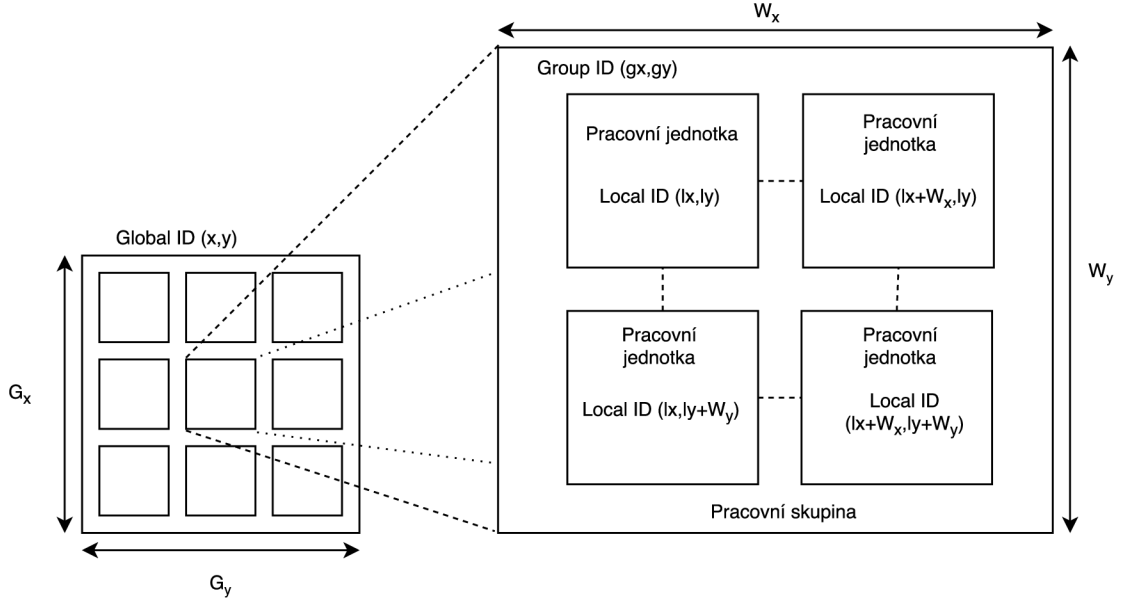
Výpis 6.2: Příklad OpenCL kernelu.

### 6.2.6 Separabilita kernelů a její využití při optimalizaci

Kernel velikosti  $N \times M$ , který je možné vyjádřit pomocí dvou vektorů o velikosti  $N$  a  $M$  vztahem:

$$K(x, y) = K_v(x) \otimes K_h(y) \quad (6.4)$$

kde  $K(x, y)$  představuje matici kernelu a  $K_v$ ,  $K_h$  jsou vertikálně a horizontálně orientované vektory, nazýváme kernelem separabilním. Ověřit separabilitu kernelu můžeme pomocí výpočtu hodnoty matice filtru – pokud je hodnota rovna jedné,



Obr. 6.3: Struktura 2-D výpočetních operací v OpenCL.

kernel je separabilní. Toto vyjádření umožňuje rozdělit proces 2-D filtrace na dvě po sobě jdoucí 1-D konvoluce ve vertikálním a následně horizontálním směru. Pokud tedy definujeme počet nutných operací  $N_O$  pro výpočet konvoluce vstupních dat o rozměru  $N \times M$  s filtračním kernelem o rozměru  $P \times Q$  přímou metodou jako:

$$N_O = MNPQ, \quad (6.5)$$

a počet operací  $N_{OS}$  potřebných pro výpočet separabilní verze stejné konvoluce jako:

$$N_{OS} = MNP + MNQ = MN(P + Q), \quad (6.6)$$

můžeme teoretický faktor zrychlení  $K$  vypočítat dle vztahu:

$$K = \frac{N_O}{N_{OS}} = \frac{PQ}{P + Q}. \quad (6.7)$$

### Separabilita říditelných filtrů

Jako říditelné filtry označujeme filtry, u kterých je možné parametricky řídit rotaci kernelu, která zároveň ovlivňuje i filtraci vstupního signálu. Do této skupiny tedy patří i filtry představené v kapitole 3 při popisu metody ADTFD. Ty však samy o sobě jednoduše separabilní nejsou. Můžeme však využít faktu, že filtrovaný signál můžeme získat pomocí filtrování lineární kombinací filtrů, které splňují podmínku, že jejich součet je rovný kernelu původního filtru. Druhou derivaci gaussovského filtru  $G_2$  lze vyjádřit pomocí tří filtrů (bází)  $G_{2a}$ ,  $G_{2b}$ ,  $G_{2c}$  jako: [23]

$$G_2^a = k_a(\theta)G_{2a} + k_b(\theta)G_{2b} + k_c(\theta)G_{2c}, \quad (6.8)$$

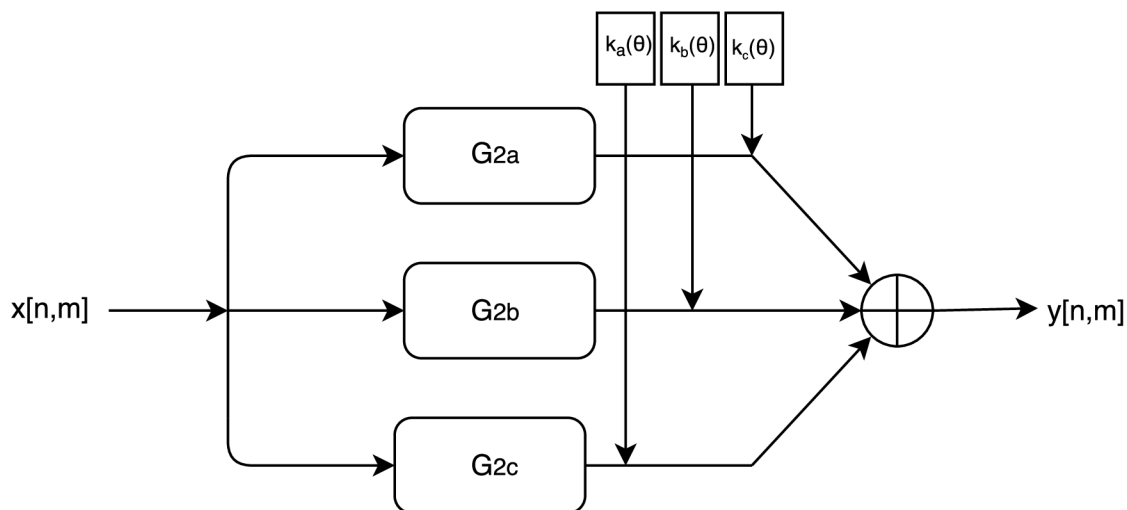
kde  $k_a(\theta)$ ,  $k_b(\theta)$ ,  $k_c(\theta)$  představují váhovací funkce závislé na úhlu natočení  $\theta$ : [23]

$$\begin{aligned} k_a(\theta) &= \cos^2(\theta), \\ k_b(\theta) &= -2 \cos(\theta) \sin(\theta), \\ k_c(\theta) &= \sin^2(\theta), \end{aligned}$$

a jednotlivé báze  $G_{2a}$ ,  $G_{2b}$ ,  $G_{2c}$  jsou definované pomocí interpolačních předpisů: [23]

$$\begin{aligned} G_{2a} &= 0.9213 (2x^2 - 1) e^{(x^2+y^2)}, \\ G_{2b} &= 1.843 x y e^{(x^2+y^2)}, \\ G_{2c} &= 0.9213 (2y^2 - 1) e^{(x^2+y^2)}. \end{aligned}$$

Tyto báze již tedy jsou separabilní. Separace na jednotlivé horizontální a vertikální vektory přímo za běhu programu je však komplexní operací a proto byl vytvořen skript v prostředí aplikace Matlab, který vytvoří  $L$  různé natočených bází a příslušné vertikální a horizontální vektory. Ty jsou uloženy do hlavičkového souboru jako pole hodnot typu `float` a načteny v aplikaci. Separace je dosaženo pomocí funkce `svd(X)`, která vykonává rozklad vstupní matice na singulární hodnoty.



Obr. 6.4: Blokové schéma výpočtu směrově filtrovaného obrazu pomocí dílčích bází.

Samotná separabilní konvoluce pak byla implementována pomocí knihovny OpenCL, kdy byly vytvořeny dva výpočetní kernely – jeden pro vertikální a druhý pro horizontální směr. Postupně jsou pro každé natočení tedy vypočteny tři dvojice 1-D konvolucí a mezivýsledky jsou následně sečteny pro získání výsledné konvoluce.

## 6.3 Implementace metody SPWD

Zároveň s adaptivní metodou ADTFD byla pro srovnání výsledků a dosažených výpočetních časů implementována i metoda statického filtrování SPWD. Bylo zde přímo využito poznatků získaných při fázi optimalizace algoritmu ADTFD a filtrování tedy bylo realizováno pomocí knihovny OpenCL s využitím separability gaussovského filtru.

## 6.4 Testovací aplikace

Vzhledem k tomu, že knihovna byla dle zadání koncipována pro analýzu v reálném čase, byla vytvořena testovací konzolová aplikace simulující real-time zpracování segmentací vstupního signálu a následného volání funkcí knihovny pro výpočet výsledné časově-frekvenční distribuce v rámci jednotlivých segmentů.

### 6.4.1 Problematika proměnlivé frekvenční nejistoty

Ze samotné definice Wignerovy distribuční funkce (popř. definice lokální autokorelační funkce) a stejně tak ze spektrogramů získaných v dílčí části práce věnované výsledkům úvodní rešerše a Matlab implementace představených metod je patrné, že frekvenční nejistota ve výsledné WDF není konstantní. Konkrétně můžeme říci, že tato nejistota se snižuje od okrajových hodnot časové osy směrem ke středu. Při zpracovávání dlouhých signálů je tato vlastnost zanedbatelná, nicméně při aplikaci WDF jako real-time metody můžeme předpokládat, že tato bude aplikována na krátké bloky s konstantní velikostí, přičemž výstupní distribuce jsou postupně řazeny za sebe do jednoho finálního spektrogramu. V tomto případě je tedy nežádoucí, aby bylo v jednotlivých blocích dosahováno výrazných rozdílů frekvenční nejistoty v krajních a středové oblasti segmentu.

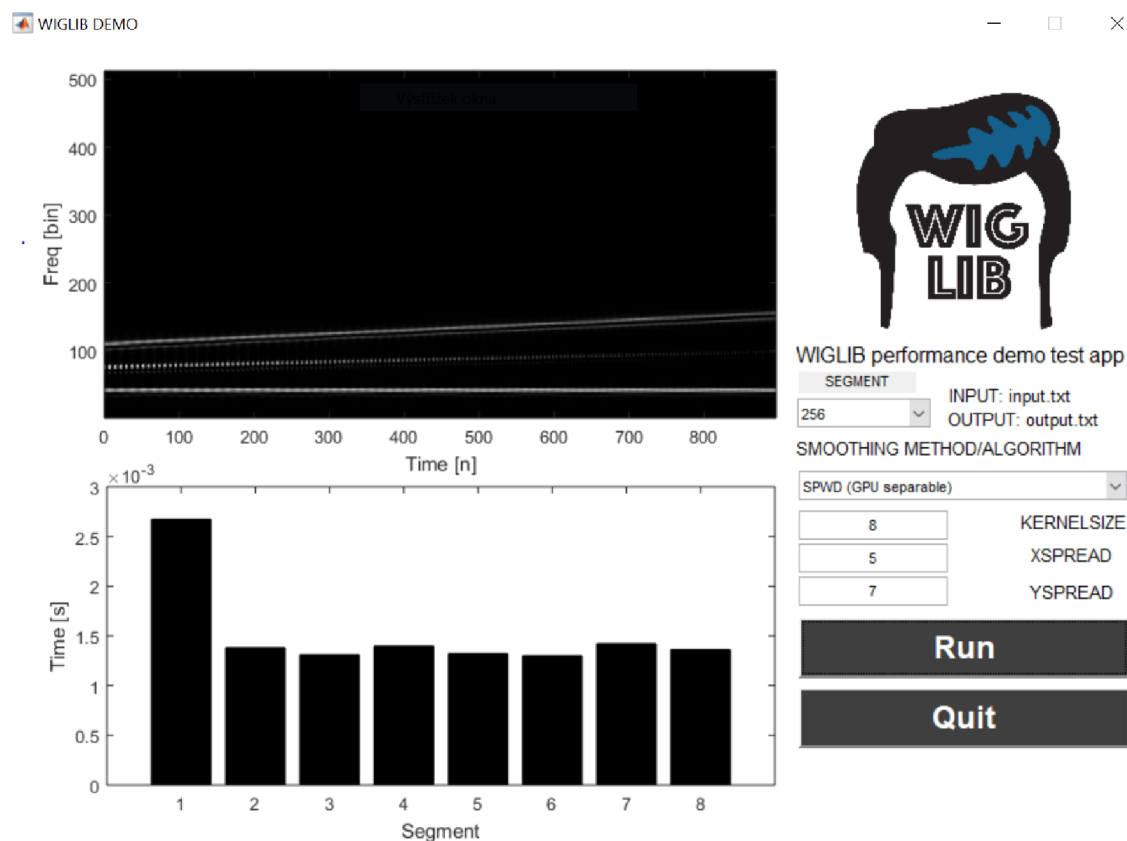
Pro účely testovací aplikace byl použit poměrně přímočarý způsob řešení tohoto problému spočívající v segmentaci vstupního signálu s polovičním překryvem a následným výběrem oblasti spektrogramu s časovou osou omezenou intervalem:  $t \in [\frac{N_s}{4}, \frac{3N_s}{4}]$ , kde  $N_s$  je délka segmentu.

### 6.4.2 Měření a práce s testovací aplikací

V implementační fázi práce byla řešena především problematika optimalizace a to pomocí řady algoritmů představených v předchozích podkapitolách. Bylo tedy nutné implementovat mechanismy pro měření a sběr výpočetních časů pro pozdější zhodnocení dosažených výsledků. Měření probíhá pomocí knihovny `chrono`, která je součástí standardní knihovny `std`. Jednotlivá měření jsou uchovávána v průběhu celého

životního cyklu příslušného objektu a po dokončení analýzy vstupního signálu jsou uložena do textového souboru pojmenovaného dle použité metody a délky segmentu.

Testovací aplikaci je možné ovládat jako běžnou konzolovou aplikaci – tedy předáním parametrů pomocí příkazové řádky systému Windows. Pro usnadnění sběru naměřených dat byl však vytvořen dávkový příkaz systému Windows umožňující alespoň z části automatizovat proces analýzy. Současně byl pro demonstrační účely vytvořen i Matlab skript rozšířený o grafické uživatelské rozhraní viz ilustrace 6.5.

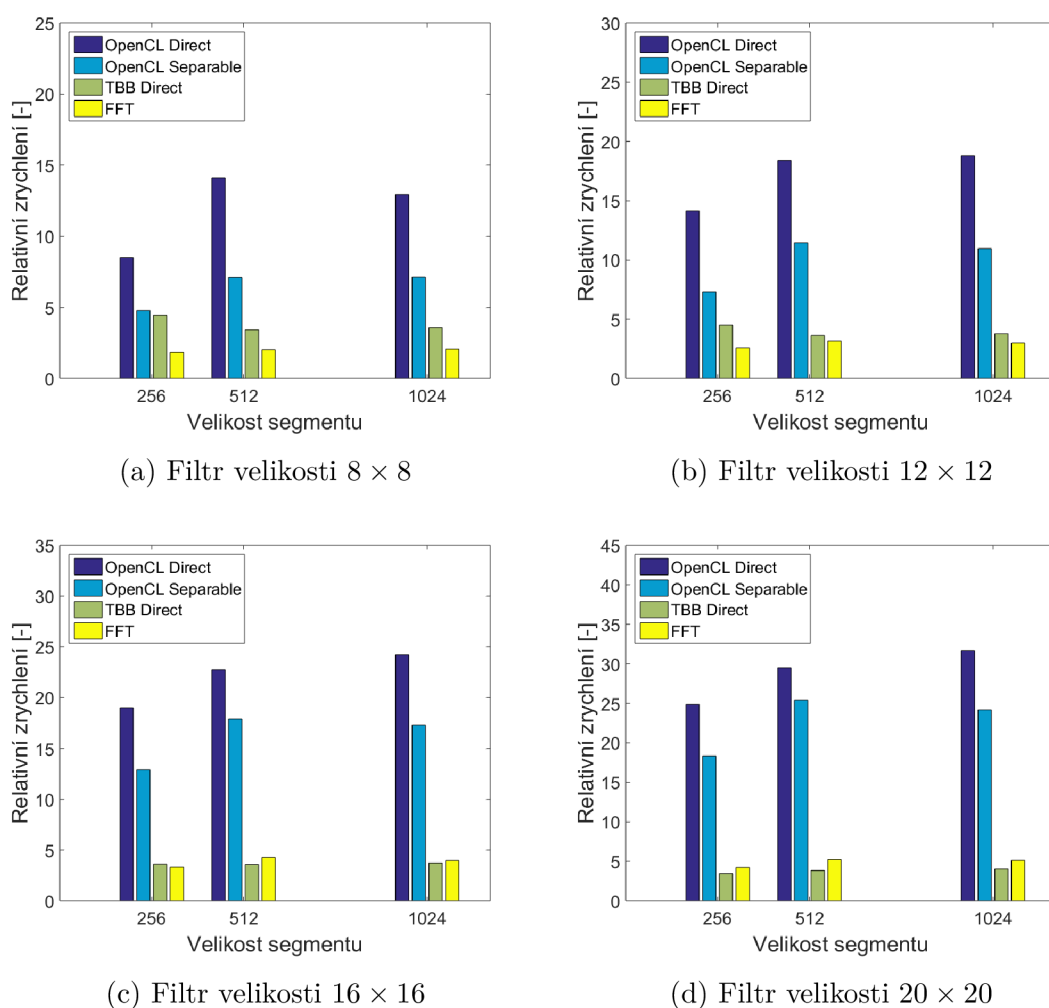


Obr. 6.5: Demonstrační MATLAB aplikace.

## 7 VÝSLEDKY IMPLEMENTACE

### 7.1 Srovnání rychlostí konvolučních algoritmů v metodě ADTFD

První část analýzy dosažených výsledků byla věnována testování jednotlivých algoritmů pro výpočet dílčích konvolucí v adaptivní metodě redukce interferenčních složek ADTFD. Měřítkem pro výběr výpočetně nejefektivnějšího algoritmu bylo zvoleno relativní zrychlení vztahené k časové náročnosti přímé, sériové implementace 2-D konvoluce dle vztahu 6.2.



Obr. 7.1: Srovnání relativních zrychlení implementovaných algoritmů pro výpočet dílčích konvolucí v metodě ADTFD.

Z přiložené ilustrace 7.1 můžeme pozorovat, že nejlepších výsledků bylo dosaženo



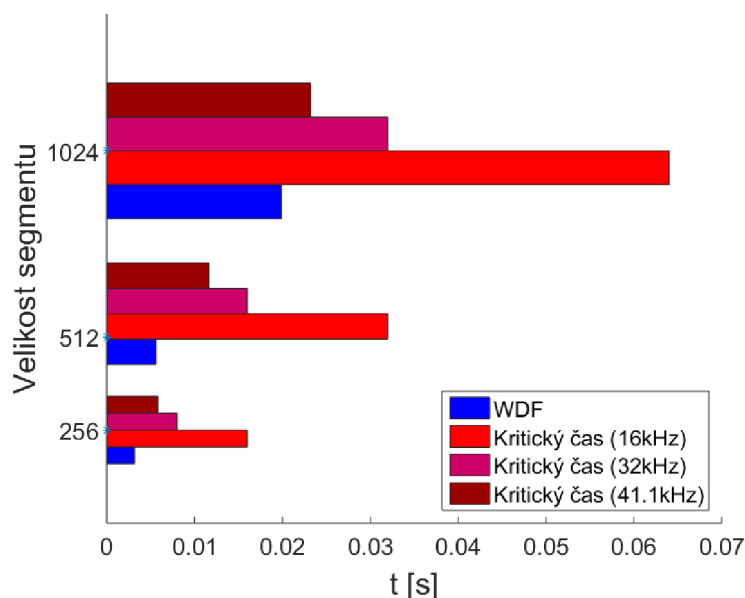
pomocí přímé implementace konvoluce na GPU. Separabilní verze GPU konvoluce nedosahovala výsledků přímé implementace zřejmě z důvodu většího množství režijních operací prováděných na CPU (sčítání meziproductů z důvodu zachování říditelnosti filtrů) a většímu počtu vykonávaných OpenCL kernelů (tři dvojice kernelů – vertikální a horizontální průchod). Implementace rychlé konvoluce pomocí FFT dosahovala lepších výsledků oproti implementaci pomocí knihovny TBB u filtračních kernelů od velikosti  $16 \times 16$  (viz. 7.1c).

## 7.2 Testování implementovaných metod pro analýzu v reálném čase

Pojem zpracování v reálném čase může být v případě spektrální analýzy poměrně zavádějící, neboť výpočet spektra vždy probíhá nad celistvým blokem (segmentem) vzorků analyzovaného signálu. Proto zavádíme pojem kritický čas  $T_k$ , jakožto dobu trvání bloku  $N_s$  vzorků v sekundách v závislosti na dané vzorkovací frekvenci  $f_s$ , definovaný vztahem:

$$T_k = N_s f_s^{-1}. \quad (7.1)$$

Algoritmus schopný pracovat v reálném čase pak definujeme jako takový algoritmus, jehož výpočetní čas nepřesáhne čas kritický (nový vstupní segment není k dispozici před dokončením výpočtu nad předešlým segmentem).



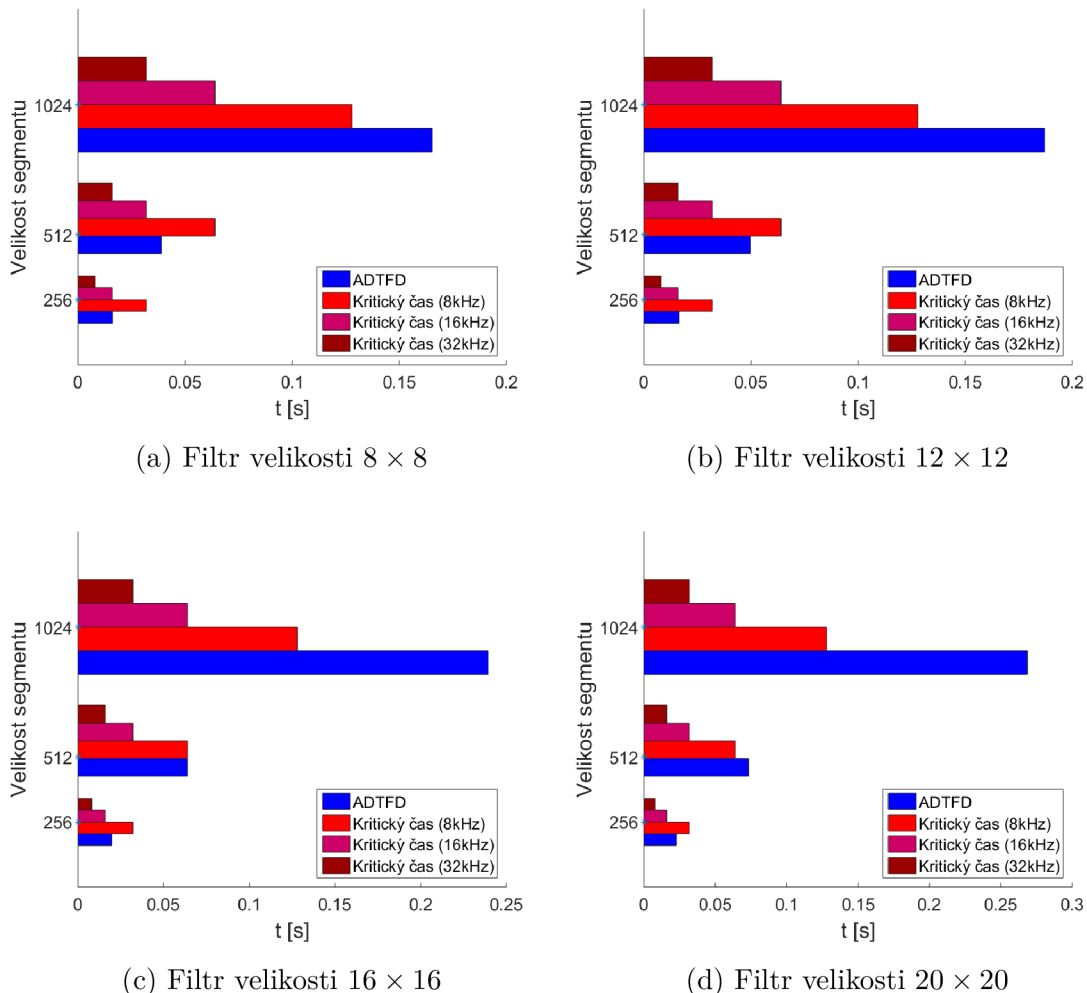
Obr. 7.2: Testování real-time funkcionality bloku pro výpočet Wignerovy distribuční funkce.

## 7.2.1 Výpočet Wignerovy distribuční funkce

Nejprve byl testován blok pro výpočet nevyhlazené Wignerovy distribuční funkce. Z grafu 7.2 můžeme pozorovat, že kritický výpočetní čas nebyl pro dané vzorkovací frekvence (16 kHz, 32 kHz, 44,1 kHz) a velikosti bloků (256, 512 a 1024 vzorků) přesažen v ani jednom z případů.

## 7.2.2 Výpočet vyhlazené WDF pomocí metody ADTFD

Adaptivní metoda vyhlazování Wignerovy distribuční funkce představuje značnou zátěž pro přidělené výpočetní prostředky. Na základě poznatků získaných v sekci 7.1 byl vybrán výpočet konvoluce přímým vztahem na GPU jakožto nejrychlejší implementovaný algoritmus.



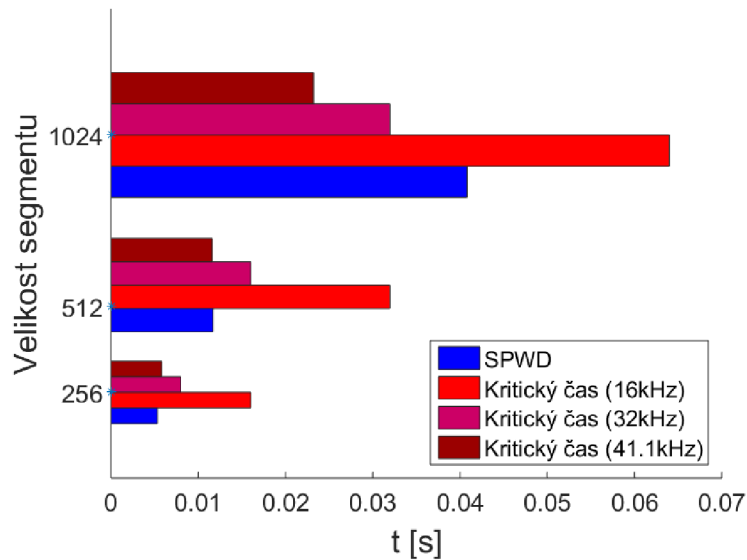
Obr. 7.3: Srovnání kritických a dosažených výpočetních časů metody ADTFD.

Tento byl testován měřením konkrétních výpočetních časů pro jednotlivé velikosti bloků a také pro různé velikosti filtrovacích kernelů (symetrické,  $N_x = N_y = 8, 12, 16, 20$ ).

Z ilustrace 7.3 můžeme pozorovat, že kritický čas pro vzorkovací frekvenci 16kHz nebyl překonán pouze pro blok velikosti 256 vzorků a velikosti filtru  $8 \times 8$  pixelů (7.3a). Pro vzorkovací frekvenci 8kHz bylo dosaženo podmínky pro zpracování v reálném čase pro bloky velikosti 256 a 512 vzorků až do velikosti filtračního jádra  $16 \times 16$  pixelů.

### 7.2.3 Výpočet vyhlazené WDF pomocí metody SPWD

Výpočetní časy dosažené při výpočtu Wigner-Villeho distribuční funkce a následného vyhlazení metodou SPWD znázorňuje graf 7.4. Dle očekávání bylo dosaženo kratších výpočetních časů oproti adaptivní metodě vyhlazování z důvodu statického nastavení parametrů filtračního jádra (tedy i absenci nutnosti výpočtu dílčích konvolučních operací). Při použití filtru velikosti  $16 \times 16$  pixelů tak bylo dosaženo pro bloky velikosti 256 a 512 vzorků dostatečně krátkého výpočetního času pro všechny testované vzorkovací frekvence.



Obr. 7.4: Testování real-time funkcionality bloku pro výpočet Wignerovy distribuční funkce

## 7.3 Testovací konfigurace

Prezentovaných výsledků bylo dosaženo testováním na běžném výpočetním hardwaru desktopové PC konfigurace specifikované v tab. 7.1 . Podrobnou specifikaci použité grafické karty lze nalézt v tab. 7.2, parametry procesoru v tab. 7.3.

<b>CPU</b>	AMD RYZEN 7 1700
<b>RAM</b>	16 GB DDR4 (2300 MHz)
<b>GPU</b>	ASUS DUAL RX460 O2G

Tab. 7.1: Specifikace testovací sestavy

<b>GPU ČIP</b>	AMD RADEON RX460
<b>FREKVENCE JÁDRA</b>	1220 MHz
<b>POČET VÝPOČET. JEDNOTEK</b>	1024
<b>VELIKOST PRACOVNÍ SKUPINY</b>	256
<b>PAMĚŤ</b>	2GB (GDDR5)
<b>FREKVENCE PAMĚTI</b>	7000 MHz

Tab. 7.2: Parametry grafické karty ASUS Dual RX460 O2G

<b>OZNAČENÍ ČIPU</b>	ZEN
<b>POČET JADER</b>	8
<b>TAKTOVACÍ FREKVENCE</b>	3.0 GHz
<b>L2 CACHE</b>	4 MB
<b>L3 CACHE</b>	16 MB

Tab. 7.3: Parametry procesoru AMD Ryzen 7 1700

## 8 ZÁVĚR

V první části práce byly nastudovány metody pro spektrální analýzu signálů ve vysokém rozlišení jako alternativy pro běžnou krátkodobou Fourierovu transformaci. Zadané metody spektrálního přibližování byly po úvodní rešerši rozšířeny i o problematiku kvadratických časově-frekvenčních distribucí. Mimo představení teoretického pozadí studovaných metod byly tyto metody implementovány v programovém prostředí aplikace Matlab a dosažené výsledky byly zhodnoceny a prezentovány prostřednictvím přiložených ilustrací.

Přímé srovnání metody komplexní demodulace a metod založených na Wignerově distribuční funkci není příliš vhodné, neboť zatímco kvadratické časově-frekvenční distribuce poskytují výrazné zjemnění rozlišení analýzy v časovém rozlišení současně s redukcí efektu Heisenbergova principu neurčitosti, komplexní demodulace je spíše optimalizační úlohou pro výpočet spektra úzkého, oboustranně omezeného pásma signálu.

Metoda komplexní demodulace představuje výpočetně efektivní a jednoduše implementovatelný nástroj, obecný přínos případné implementace by však byl minimální. Oblast kvadratických časově-frekvenčních distribucí však přináší mnohem zajímavější možnosti a to jak z hlediska implementace, tak z hlediska možné aplikace. Nese s sebou však řadu úskalí, zejména v podobě vyšší výpočetní náročnosti a přítomnosti interferenčních spektrálních složek. V práci byly, mimo jiné, zkoumány možnosti redukce těchto nežádoucích složek a na základě získaných znalostí navržena i vlastní adaptivní metoda redukce interferenčních spektrálních složek v kvadratických časově-frekvenčních distribucích.

Přestože navržený vyhlazovací algoritmus AOK-RGA vykazoval na dané množině testovacích signálů slibné výsledky i ve srovnání s existující adaptivní metodou ADTFD, vzhledem ke sporné možnosti aplikace na komplexnější signály a možnosti případné návaznosti na metodu HADTFD byla pro implementaci zvolena metoda ADTFD. Druhou implementovanou metodou řešící problematiku interferenčních spektrálních složek ve Wignerově distribuční funkci byla zvolena metoda statického filtrování SPWD.

Klíčovou fází finální implementace byla výpočetní optimalizace metod pro vyhlazování kvadratických časově-frekvenčních distribucí. Bylo navrženo několik algoritmů využívajících různých paralelních i sériových technologií a metod s cílem maximálně redukovat výpočetní časy, které, vzhledem k přesunu od jednorozměrných signálů do dvourozměrné domény spektrogramu, dosahují výrazně vyšších hodnot oproti konvenčním metodám spektrální analýzy.

Ačkoliv bylo u adaptivní metody ADTFD dosaženo výrazného relativního zrychlení oproti přímé implementaci (v případě GPU paralelizace až přibližně 30 násob-

ného), při simulaci analýzy v reálném čase bylo dosaženo pozitivního výsledku pouze pro menší vstupní bloky (256, 512 vzorků) a nízké vzorkovací frekvence (8 kHz, 16 kHz).

Metoda SPWD vykazovala při simulaci lepší výsledky díky statickému nastavení parametrů filtračního jádra bez nutnosti výpočtu dílčích filtrací. Pro bloky o velikosti 256 a 512 vzorků tak bylo, při vzorkovací frekvenci uvažované až do kmitočtu 44,1 kHz, dosaženo (pro filtr velikosti  $16 \times 16$  pixelů) výpočetních časů odpovídajících zpracování v reálném čase.

Bylo zjištěno, že obzvláště v případě zpracování větších vstupních segmentů tvoří podstatnou část celkového výpočetního času samotný výpočet Wigner-Villovy distribuční funkce. Proto by bylo vhodné zvážit optimalizaci algoritmu tohoto výpočtu, zejména z důvodu redundance použitého FFT algoritmu (výsledkem je pouze reálná část komplexní DFT). Případně by bylo možné prozkoumat možnosti paralelizace samotného výpočtu Rychlé Fourierovy transformace na GPU.

Schopnost metod pracovat v reálném čase byla určena pomocí kritických časů definovaných v kapitole 7.2. Je však nutné poznamenat, že závisí také na způsobu aplikace implementovaných metod. Kupříkladu pro vizuální analýzu v reálném čase není nutné, aby vyhlazovací algoritmy splňovaly podmínku kritického času v případě, že ji splňuje algoritmus pro výpočet výchozí Wignerovy distribuční funkce, čehož pro testované vzorkovací frekvence a velikosti bloků dosaženo bylo. V případě, že je však požadována vyhlazená distribuce již v rámci segmentu definovaného kritickým časem, použití adaptivní metody není, alespoň na běžném výpočetním hardwaru při vyšších vzorkovacích frekvencích, vhodné.

Problematiku zpracování větších bloků by v případě omezeného pásma zájmu bylo možné, na základě poznatků získaných z preimplementační studie, řešit i kombinací metody komplexní demodulace s kvadratickými časově frekvenčními distribucemi, čímž by bylo dosaženo snížení výpočetní náročnosti a zároveň by však bylo zachováno vysoké časové a frekvenční rozlišení Wigner-Villeho distribuční funkce.

Hlavní cíl práce, tedy výběr a implementace metody pro spektrální analýzu ve vysokém rozlišení v reálném čase, byl splněn. Práce pokryla širokou oblast vědních oborů od zpracování jednorozměrných signálů, přes problematiku zpracování obrazových dat až po obor aplikované informatiky v podobě optimalizačních procesů v implementační části práce. Přesto však zůstává prostor pro případná rozšíření a navazující studie. Zajímavou možností by mohlo být např. využití strojového učení pro výběr optimálního filtrovacího jádra pro redukci interferenčních složek, čímž by (při vhodně zvolených příznacích) bylo možné dosáhnout snížení výpočetní náročnosti adaptivního filtrování.

## LITERATURA

- [1] Z. SMÉKAL, *Analýza signálů a soustav-BASS*. FEKT, 2012.
- [2] D. L. JONES, *Decimation-in-time (DIT) Radix-2 FFT*. OpenStax CNX. 2006 [cit. 2017-05-19]. Dostupné z URL: <<http://cnx.org/contents/ce67266a-1851-47e4-8bfc-82eb447212b4@7>>.
- [3] W.F. MICHAEL, Jr. *Constant-Q Spectral Analysis by Means of an Approximate Fast Fourier Transform Technique*. NAVAL UNDERWATER SYSTEMS CENTER NEW LONDON CONN NEW LONDON LAB, 1977.
- [4] YOUNG S. SHIN, JAE-JIN JEON. *Pseudo Wigner–Ville Time-Frequency Distribution and Its Application to Machinery Condition Monitoring*. Department of Mechanical Engineering Naval Postgraduate School, 1993.
- [5] R. G. BARANIUK, D. L. JONES. *Optimal Kernels for Time-Frequency Analysis* San Diego, CA | July 08, 1990, 1993.
- [6] D.H. BALLARD, C. M. BROWN. *Computer vision* Englewood Cliffs, N.J.: Prentice-Hall, c1982. ISBN 0131653164.
- [7] B.BOASHASH *Time-frequency Signal Analysis and Processing A Comprehensive Review* Academic Press, 2013. ISBN 9780123984999
- [8] S. THEODORIDIS,R. CHELLAPPA. *Academic Press Library in Signal Processing, 1st Edition Array and Statistical Signal Processing* Academic Press, 2014. ISBN 9780124115972
- [9] J. STANKOVIC, T. THAYAPARAN, M. DAKOVIC, V. POPOVIC. *S-Method in radar imaging* 14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy, September 4-8, 2006
- [10] A. V. PRABHU, V.K. AATRE, T. SOUMINI, S. A. KARIPPEL *Frequency Zooming Techniques for High Resolution Spectrum Analysis* Naval Physical & Oceanographic Laboratory, 1985
- [11] K. MURUGAN , R. RAMESH. *Electrocardiogram signal analysis using zoom FFT* 2012 ISSNIP Biosignals and Biorobotics Conference: Biosignals and Robotics for Better and Safer Living, 2012
- [12] KHAN, Nabeel Ali a Boualem BOASHASH *Multi-component instantaneous frequency estimation using locally adaptive directional time frequency distributions*. International Journal of Adaptive Control and Signal Processing [online].

- Bognor Regis: Wiley Subscription Services, 2016, 30(3), 429-442 [cit. 2016-12-07]. DOI: 10.1002/acs.2583. ISSN 08906327.
- [13] MOHAMMADI, Mokhtar, Ali POUYAN a Nabeel KHAN. *A highly adaptive directional time–frequency distribution*. Signal, Image and Video Processing [online]. London: Springer London, 1610, 10(7), 1369-1376 [cit. 2016-12-08]. DOI: 10.1007/s11760-016-0901-x. ISSN 1863-1703.
- [14] C. J. HOPE, D. J. FURLONG. *Time-Frequency Distributions for Timbre Morphing: The Wigner Distribution versus the STFT* Proceedings of the SBCMIV, (4th Symposium of Brazilian Computer Music), 1997.
- [15] R. LYONS. *Using Zoom FFT for spectral analysis* [online]. Embedded.com 2011, poslední aktualizace 22.02.2011 [cit. 6.12.2016]. Dostupné z URL: <<http://www.embedded.com/design/other/4213370/Using-Zoom-FFT-for-spectral-analysis>>.
- [16] SHIREEN ROMANA, RAJENDRA CHINCHOLI. *Zoom FFT Algorithm in Ultrasonic Blood Flow Analysis Using MATLAB* International Journal of Scientific and Research Publications, Volume 4, Issue 10, October 2014 ISSN 2250-3153
- [17] J.M. O'TOOLE, M. MESBAH, B. BOASHASH. *Efficient algorithms for discrete time-frequency distributions* Proceedings of the WSEAS International Conference on Applied Computing Conference, Istanbul, [310-315], 2008.
- [18] CAO, Jia-Lin a Guang-Hua CHEN. *Fast computation of wigner-ville distribution*. Proceedings of the WSEAS International Conference on Applied Computing Conference, Istanbul, [310-315], 2008. Journal of Shanghai University (English Edition) [online]. 2003, 7(3), 265-269 [cit. 2016-12-11]. DOI: 10.1007/s11741-003-0036-5. ISSN 1007-6417. Dostupné z URL: <<http://link.springer.com/10.1007/s11741-003-0036-5>>.
- [19] FRIGO, Matteo a Steven G. JOHNSON. *FFTW: Manual for version 3.3.6-pl1* [online]. 2017 [cit. 2017-05-19]. Dostupné z URL: <<http://www.fftw.org/fftw3.pdf>>.
- [20] INTEL CORPORATION. *Intel® Threading Building Blocks: Tutorial* [online]. 2007 [cit. 2017-05-19]. Dostupné z URL: <<https://software.intel.com/sites/default/files/m/d/4/1/d/8/tutorial.pdf>>.
- [21] INTEL CORPORATION. *OpenCL Optimization Guide: for Intel Atom and Intel Core processors with Intel Graphics* [online]. 2014 [cit. 2017-05-19].



- Dostupné z URL: <<https://software.intel.com/sites/default/files/managed/72/2c/gfxOptimizationGuide.pdf>>.
- [22] AMD. *Introduction to OpenCL Programming: Training Guide* [online]. 2010 [cit. 2017-05-19]. Dostupné z URL: <[http://developer.amd.com/wordpress/media/2013/01/Introduction\\_to\\_OpenCL\\_Programming-Training\\_Guide-201005.pdf](http://developer.amd.com/wordpress/media/2013/01/Introduction_to_OpenCL_Programming-Training_Guide-201005.pdf)>.
- [23] FREEMAN, William a ADELSON, Edward. *The Design and Use of Steerable Filters*. IEEE Trans. Pattern Anal. Mach. Intell. 13, 9 (1991), 891-906.

## SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

STFT	Short Time Fourier Transform (Krátkodobá Fourierova transformace)
FFT	Fast Fourier Transform (Rychlá Fourierova transformace)
WDF	Wignerova distribuční funkce
TFD	Time frequency distribution (časově-frekvenční distribuce)
WVD	Wigner-Villeho distribuce
PWD	Pseudo-Wignerova distribuce
SAF	Sussmanova ambiguitní funkce
ADTFD	Adaptive Directional Time-Frequency Distribution
FADTFD	Fully Adaptive Directional Time-Frequency Distribution
AOK-RGA	Adaptive Optimal Kernel by Region Growing
AF	Ambiguitní (dvojnásobná) funkce
RG	Region Growing (algoritmus)

# SEZNAM PŘÍLOH

A Obsah elektronické přílohy

63

# A OBSAH ELEKTRONICKÉ PŘÍLOHY

Příložené DVD je rozděleno na tři adresáře:

- 
- Adresář **MATLAB**  
-v jednotlivých podadresářích jsou uloženy soubory Matlab skriptů použitých v části práce věnované srovnání jednotlivých metod
- 
- Adresář **SOURCE**  
-obsahuje repozitář zdrojových kódů vytvořené knihovny
- 
- Adresář **TESTAPP**  
-obsahuje zkompilované binární soubory demonstrační aplikace
- 

**Závislosti** Pro správnou funkci demonstrační aplikace popř. pro přeložení zdrojových kódů je nutné aby byly na cílovém systému přítomny tyto knihovny:

- FFTW (v. 3.3.5)
- Intel TBB 2017 (Update 5)
- OpenCL 2.0