

Algoritmy pro odhad výkonových spektrálních hustot náhodných signálů

Bc. Michal Ostřanský

Diplomová práce
2020



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav informatiky a umělé inteligence

Akademický rok: 2019/2020

ZADÁNÍ DIPLOMOVÉ PRÁCE (projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Michal Ostránský**
Osobní číslo: **A17296**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Informační technologie**
Forma studia: **Kombinovaná**
Téma práce: **Algoritmy pro odhad výkonových spektrálních hustot náhodných signálů**
Téma práce anglicky: **Algorithms for the Estimation of the Power Spectral Densities of Random Signals**

Zásady pro vypracování

1. Vypracujte rešerši algoritmů a metod vhodných pro výpočet odhadu výkonových spektrálních hustot náhodných signálů.
2. Vybrané algoritmy realizujte v prostředí Matlab/Simulink.
3. Vyhodnoťte algoritmy z hlediska výpočetní náročnosti, přesnosti, rozlišení spektra a dalších statistických aspektů.
4. Navrhněte a formulujte zadání vhodné úlohy pro předmět Identifikace a modelování náhodných signálů.
5. Vypracujte vzorový protokol.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. ZAPLATÍLEK, Karel a Bohuslav DOŇAR. *MATLAB: začínáme se signály*. Praha: BEN – technická literatura, 2006. 271s., ISBN 80-7300-200-0.
2. DINIZ, Paulo, DA SILVA, Eduardo, NETTO, Sergio. *Digital Signal Processing*. 2nd ed. Cambridge University Press, 2010. 889s. ISBN 978-0-521-88775-5.
3. OPPENHEIM, Alan., WILLSKY, Alan. *Signals and Systems*. N.J. USA: Prentice-Hall, Englewood Cliffs, 1997. 957s. ISBN 0-13-814757-4.
4. LI, Tan. *Digital Signal Processing, Fundamentals and Applications*. Elsevier, 2008. 816 s. ISBN 978-0-12-374090-8.
5. CANDY, James. *Model Based Signal Processing*. John Wiley & Sons, 2006. 677 s., ISBN 978-0-471-23632-0.

Vedoucí diplomové práce:

doc. Ing. Marek Kubalčík, Ph.D.

Ústav řízení procesů

Datum zadání diplomové práce: 28. listopadu 2019
Termín odevzdání diplomové práce: 15. května 2020



doc. Mgr. Milan Adámek, Ph.D.
děkan

prof. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

Ve Zlíně dne 9. prosince 2019

Jméno, příjmení:

Název diplomové práce:

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

MICHAL OSTRÁNSKÝ, v. r.
podpis diplomanta

ABSTRAKT

Cílem této diplomové práce je seznámit se s problematikou týkající se algoritmů a metod vhodných pro výpočet odhadu výkonových spektrálních hustot náhodných signálů. Studium odborné literatury je třeba získat znalosti potřebné k volbě metod vhodných k odhadu spektra. V teoretické části popsat tyto metody a provést jejich rozdělení dle obvyklých pravidel. Následně je zpracovat a vybrané algoritmy realizovat v prostředí MATLAB/Simulink, což se provede v části praktické. Zde se vyhodnotí tyto algoritmy z hlediska výpočetní náročnosti, přesnosti, rozlišení spektra a dalších statistických aspektů. Nakonec se navrhne a zformuluje zadání vhodné úlohy pro předmět Identifikace a modelování náhodných signálů a vypracuje vzorový protokol.

Klíčová slova: výkonová spektrální hustota, Fourierova transformace, periodogram, MATLAB, Identifikace a modelování náhodných signálů

ABSTRACT

The aim of this master's thesis is to learn the problems related to algorithms and methods suitable for calculating the estimation of power spectral densities of random signals. Based on learning a specialised literature gain the knowledge needed to select methods suitable for spectrum estimation. In the theoretical part describe these methods and divide them according to the usual rules. Subsequently, they are processed and the selected algorithms are implemented in the MATLAB / Simulink environment, which is done in the practical part. Here, these algorithms are evaluated in terms of computational complexity, accuracy, spectrum resolution and other statistical aspects. Finally, the assignment of a suitable task for the subject Identification and modeling of random signals is designed and formulated and a sample protocol is developed.

Keywords: power spectral density, Fourier transform, periodogram, MATLAB, Identification and modeling of random signals

Děkuji tímto vedoucímu mé diplomové práce doc. Ing. Marku Kubalčíkovi, Ph.D. za všechny čas, připomínky a rady, které mi věnoval během konzultací.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 ZÁKLADY ZPRACOVÁNÍ SIGNÁLŮ	11
1.1 SPOJITÉ A DISKRÉTNÍ SIGNÁLY	11
1.2 FOURIEROVA TRANSFORMACE.....	12
1.3 DISKRÉTNÍ A RYCHLÁ FOURIEROVA TRANSFORMACE.....	14
1.4 DETERMINISTICKÉ A NÁHODNÉ SIGNÁLY.....	15
1.5 VÝKONOVÁ SPEKTRÁLNÍ HUSTOTA	16
1.6 OKÉNKOVÁNÍ	18
2 METODY PRO ODHAD SPEKTRA - TEORIE	21
2.1 SPEKTRÁLNÍ ANALÝZA	21
2.2 NEPARAMETRICKÉ METODY PRO ODHAD SPEKTRA	21
2.2.1 Metoda periodogramu	22
2.2.2 Metoda Bartlett.....	23
2.2.3 Metoda Blackman-Tukey.....	23
2.2.4 Metoda nezkresleného odhadu	25
2.2.5 Metoda Blackman–Tukey 2	25
2.2.6 Metoda Welch	25
2.2.7 Metoda minimálního rozptylu.....	26
2.2.8 Metoda diskrétní Fourierovy transformace	27
2.3 PARAMETRICKÉ METODY PRO ODHAD SPEKTRA	27
2.3.1 Metoda Burg.....	27
II PRAKTICKÁ ČÁST	30
3 METODY PRO ODHAD SPEKTRA - ALGORITMY	31
3.1 METODA PERIODOGRAMU	31
3.2 METODA BARTLETT	33
3.3 METODA BLACKMAN–TUKEY	34
3.4 METODA NEZKRESLENÉHO ODHADU	35
3.5 METODA BLACKMAN–TUKEY 2	36
3.6 METODA WELCH.....	37
3.7 METODA MINIMÁLNÍHO ROZPTYLU	38
3.8 METODA DISKRÉTNÍ FOURIEROVY TRANSFORMACE	40
3.9 METODA BURG	41
4 METODY PRO ODHAD SPEKTRA - VYHODNOCENÍ	43
4.1 INDIVIDUÁLNÍ VYHODNOCENÍ METOD	43
4.2 VZÁJEMNÉ SROVNÁNÍ METOD	49
4.3 SROVNÁNÍ VÝPOČETNÍ SLOŽITOSTI A PŘESNOSTI.....	52
4.4 VYHODNOCENÍ METOD - ZÁVĚR	53
5 NÁVRH VZOROVÉHO PROTOKOLU	55

5.1	ZADÁNÍ	55
5.2	METODA PERIODOGRAMU	55
5.3	METODA BLACKMAN-TUKEY.....	56
6	VYPRACOVÁNÍ VZOROVÉHO PROTOKOLU	57
6.1	METODA PERIODOGRAMU	57
6.1.1	Definice signálu	57
6.1.2	Algoritmus výpočtu.....	58
6.2	METODA BLACKMAN-TUKEY.....	61
6.2.1	Definice signálu	61
6.2.2	Algoritmus výpočtu.....	62
6.3	VYHODNOCENÍ METOD PERIODOGRAMU A BLACKMAN-TUKEY	65
6.4	ZÁVĚR.....	67
6.5	ZDROJOVÝ KÓD	67
	ZÁVĚR	68
	SEZNAM POUŽITÉ LITERATURY.....	70
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	71
	SEZNAM OBRÁZKŮ	72
	SEZNAM TABULEK.....	74
	SEZNAM PŘÍLOH.....	75

ÚVOD

Zpracování signálů má za sebou dlouhou a bohatou historii. Tento obor vždy těžil z úzkého propojení teorie, aplikací a technologií pro implementaci systémů zpracování signálu. Zpracování signálů se zabývá zobrazením, transformací a úpravou signálů a informací, které obsahují. Patří sem např. vylepšení některé ze složek signálu nebo oddělení dvou a více signálů, které byly nějakým způsobem zkombinovány. K získání původních složek složeného nebo třeba zašuměného signálu, slouží právě jedna z částí tohoto oboru, která se zabývá spektrální analýzou. Sem patří i vše, co se týká výkonové spektrální hustoty, algoritmů a metod, které slouží k jejímu odhadu. Cílem této diplomové práce je seznámit čtenáře s danou problematikou a ukázat práci s metodami, které slouží k odhadu výkonové spektrální hustoty.

Teoretická část diplomové práce se zabývá základy zpracování signálů, Fourierovou transformací (Fourier Transform - FT), diskrétní Fourierovou transformací (Discrete Fourier Transform - DFT) a její implementací – algoritmy rychlé Fourierovy transformace (Fast Fourier Transform - FFT). Dále pak deterministickými a náhodnými signály, přechodem z časové do frekvenční oblasti, autokorelační funkcí a konečně výkonovou spektrální hustotou (Power Spectral Density - PSD). Druhá část teorie se pak přímo věnuje metodám, které slouží k odhadu PSD. Patří sem metoda periodogramu, Bartlett, Blackman-Tukey, metoda nezkráceného odhadu, Blackman-Tukey 2, Welch, metoda minimálního rozptylu, metoda DFT a Burg.

V praktické části jsou zpracovány všechny výše zmíněné metody, v prostředí MATLAB jsou naprogramovány jejich algoritmy a zobrazeny jejich výsledky v podobě PSD. Algoritmy jsou porovnány z hlediska výpočetní náročnosti, přesnosti, rozlišení spektra, zkrácení a dalších statistických aspektů jako je třeba rozptyl. Závěr praktické části obsahuje návrh a formulaci zadání vhodné úlohy pro předmět Identifikace a modelování náhodných signálů a je vypracován vzorový protokol.

I. TEORETICKÁ ČÁST

1 ZÁKLADY ZPRACOVÁNÍ SIGNÁLŮ

Signál může být definován jako jakákoli fyzikální veličina, která se mění v závislosti na čase, prostoru nebo jakékoli jiné proměnné nebo proměnných. Signály zprostředkovávají informace o hodnotách fyzikálních veličin. Práce s těmito informacemi zahrnuje získávání, ukládání, přenos a transformaci signálů.

Signály například přenášejí informace o stavu nebo chování systému. Pro zjednodušení analýzy a návrhu systémů zpracování signálů je téměř vždy nutné reprezentovat signály pomocí matematických funkcí jedné nebo více nezávislých proměnných. Například zvukový signál může být modelován pomocí matematické funkce času. Nezávislé proměnné v matematických modelech signálů mohou být spojité nebo diskrétní. Spojité signály jsou spojité v čase. Nezávisle proměnná je tedy spojitá. Signály diskrétní jsou definovány v diskrétních časech a nezávislá proměnná nabývá diskrétních hodnot v určitých diskrétních časech. To znamená, že diskrétní časové signály tvoří posloupnost čísel. Rozdíl mezi signálem spojitým a diskrétním má velký význam v matematických nástrojích používaných pro jejich reprezentaci a analýzu. [5]

1.1 Spojité a diskrétní signály

Kromě nezávislých proměnných, které mohou být spojité nebo diskrétní, také amplituda signálu může být diskrétní nebo spojitá. Digitální signály jsou třídou, která je diskrétní jak v čase, tak v amplitudě. Podobně jako signály je možné kategorizovat také systémy. Spojité časové systémy zahrnují systémy, jejichž vstup a výstup jsou spojité a analogicky platí, že diskrétní časové systémy jsou systémy, které mají vstup i výstup diskrétní. Podobně pro digitální systémy platí, že jejich vstupem a výstupem jsou digitální signály. [5]

Zpracování diskrétních časových signálů v průběhu času značně pokročilo. Tento pokrok nepřišel ze dne na den, ale byl učiněn postupně. V 17. století vyvinuli vědci a inženýři fyzikální modely pro fyzikální jevy založené na diferenciálních rovnicích a spojitých proměnných. V následujících stoletích vývoj v této oblasti pokračoval a posléze v roce 1965 dva američtí matematici James William Cooley a John Wilder Tukey navrhli sadu optimálních algoritmů pro výpočet Fourierovy transformace zvané FFT. [4]

Diskrétní časové signály jsou matematicky znázorněny pomocí posloupnosti čísel. Posloupnost čísel x , ve které $x[n]$ představuje n – tý vzorek této posloupnosti je následující: [4]

$$x = \{x[n]\}, \quad -\infty < n < \infty.$$

N je celé číslo. V praxi lze tyto posloupnosti čísel získat periodickým vzorkováním analogového signálu. To znamená, že číselná hodnota n – *tého* vzorku této posloupnosti se rovná hodnotě analogového signálu v čase nT : [4]

$$x[n] = x(nT), \quad -\infty < n < \infty.$$

T je perioda vzorkování a převrácená hodnota vzorkovací frekvence. Aby bylo možné spojitý časový signál zpětně zrekonstruovat po jeho převodu na čas diskrétní, je nutné, aby frekvence vzorkování byla vyšší než dvojnásobek nejvyšší harmonické složky vzorkovaného signálu a frekvence rovná polovině vzorkovací frekvence se nazývá Nyquistova frekvence. [4]

1.2 Fourierova transformace

Již bylo řečeno, že je velmi běžné znázorňovat signály pomocí časové funkce nezávislé proměnné. V souladu s tím lze tuto funkci převést do frekvenční oblasti. To jako první provedl francouzský matematik Joseph Fourier. Fourierovou transformací funkce je funkce komplexní, jejíž amplituda představuje hodnotu funkce na dané frekvenci a její argument pak udává fázi sinusového signálu na této frekvenci.

K dispozici je také zpětná Fourierova transformace, která rekonstruuje časovou funkci signálu z jeho kmitočtových složek. Pokud nazveme $X(j\omega)$ Fourierovou transformací spojitěho časového signálu $x(t)$, pak Fourierova transformace a inverzní Fourierova transformace jsou následující: [5]

$$X(j\omega) = \int_{-\infty}^{+\infty} x(t)e^{-j\omega t} dt, \quad (1)$$

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(j\omega)e^{j\omega t} d\omega. \quad (2)$$

Dvě výše uvedené rovnice (1) a (2) se nazývají Fourierova transformační dvojice, funkce $X(j\omega)$ se nazývá Fourierova transformace nebo Fourierův integrál funkce $x(t)$ a syntézní rovnice (2) se nazývá inverzní Fourierova transformace. Syntézní rovnice pro periodické

signály je rovna Fourierově řadě. Jak pro Fourierovu transformaci, tak pro Fourierovu řadu platí, že jsou lineární kombinací komplexních exponenciálních funkcí.

Transformace $X(j\omega)$ aperiodického signálu se nazývá spektrum $x(t)$. Informace potřebné k jeho popisu jsou k dispozici jako například lineární kombinace sinusových signálů s různými kmitočky. [5]

K lepšímu porozumění principům Fourierovy transformace poslouží následující příklad. Vezme se v úvahu signál $x(t) = e^{-at} u(t)$ a vzhledem k rovnici (1) platí [5]

$$X(j\omega) = \int_0^{\infty} e^{-at} e^{-j\omega t} dt = -\frac{1}{a + j\omega} e^{-(a+j\omega)t} \Big|_0^{\infty}.$$

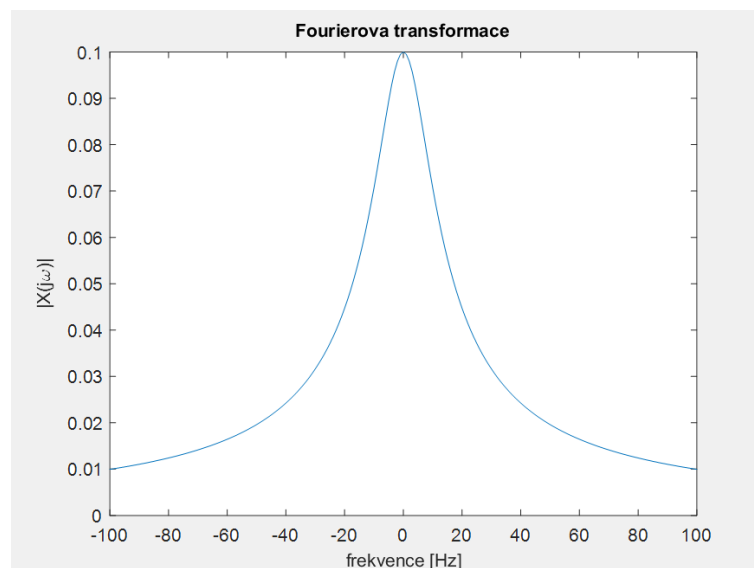
Pak

$$X(j\omega) = \frac{1}{a + j\omega} \quad a > 0.$$

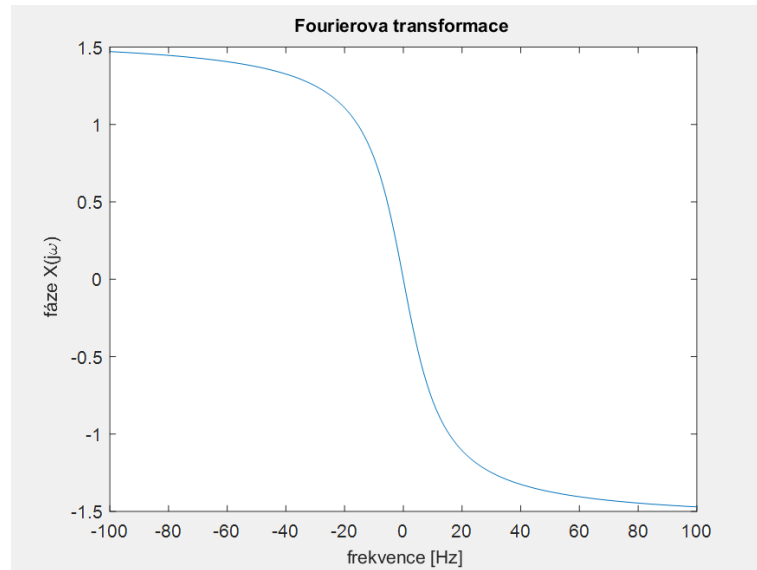
Protože je tato Fourierova transformace komplexní, získá se amplituda a fáze:

$$|X(j\omega)| = \frac{1}{\sqrt{a^2 + \omega^2}}, \quad \angle[X(j\omega)] = -\tan^{-1}\left(\frac{\omega}{a}\right).$$

Na obrázku (Obr. 1) a (Obr. 2) je vidět graf amplitudy (sudá funkce) a fáze (lichá funkce) pro $a = 10$.



Obr. 1. Amplituda FT.



Obr. 2. Fáze FT.

Dalším příkladem může být získání spektra jednotkového impulsu:

$$x(t) = \delta(t).$$

Pak

$$X(j\omega) = \int_{-\infty}^{\infty} \delta(t) e^{-j\omega t} dt = 1.$$

To znamená, že Fourierova transformace jednotkového impulsu je tvořena ze složek stejné velikosti na všech frekvencích. [5]

Spektrum diskrétních časových signálů a stejně tak i spojitých, lze zobrazit ve frekvenční oblasti pomocí Fourierovy transformace. Rovněž Fourierova transformace signálu diskrétního představuje rozklad signálu na nekonečnou sumu komplexních diskrétních sinusovek.

1.3 Diskrétní a rychlá Fourierova transformace

Pro N po sobě jdoucích vzorků $x(n)$, $0 \leq n \leq N - 1$ periodické nebo aperiodické posloupnosti je N - bodová diskrétní Fourierova transformace (DFT) $X(k)$, $0 \leq k \leq N - 1$ definována jako [3]

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}kn}. \quad (3)$$

Dáno N DFT koeficientů $X(k)$, $0 \leq k \leq N - 1$, N hodnot vzorků $x(n)$, $0 \leq n \leq N - 1$ mohou být získány pomocí $N - \text{bodové}$ inverzní DFT (IDFT) dané [3]

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j\frac{2\pi}{N}kn}. \quad (4)$$

Protože $X(k)$ je funkcí diskrétního frekvenčního indexu k , který odpovídá diskrétní sadě frekvencí $\omega k = (2\pi/N)$, $k = 0, 1, \dots, N - 1$, tvoří rovnice (3) a (4) diskrétní Fourierovu transformační dvojici. Tento název se používá k zdůraznění skutečnosti, že máme Fourierovu transformaci, která je diskrétní jak v čase, tak ve frekvenci. [3]

DFT je mocný reverzibilní nástroj pro konečný počet prvků diskrétní časové posloupnosti. DFT a její inverze lze efektivně spočítat pomocí rodiny algoritmů známých jako rychlá Fourierova transformace neboli FFT. Objevení algoritmů FFT stanovilo DFT jako jeden ze základních nástrojů v oblasti digitálního zpracování signálů. [3]

Výpočetní složitost přímého DFT algoritmu je N^2 nebo $O(N^2)$. Landauova notace (též notace velké O nebo notace omikron) je notace používaná v matematické informatice pro porovnání asymptotické časové nebo prostorové složitosti algoritmů. Protože výpočetní čas pro $O(N^2)$ algoritmy velmi rychle narůstá pro velké hodnoty N , bylo nutno se zabývat vývojem algoritmů, které snižují počet operací a tím výpočetní čas. Z tohoto důvodu se začala používat metoda FFT namísto algoritmu DFT. Joseph Fourier s využitím periodicity a komplexně sdružených symetrických vlastností $e^{j\frac{2\pi}{N}kn}$ založil rodinu $O(N \log_2 N)$ algoritmů, známých jako algoritmy rychlé Fourierovy transformace (FFT). [3]

1.4 Deterministické a náhodné signály

Zatím byly zmíněny především deterministické signály. Existují i jiné typy signálů, které spadají do kategorie náhodných procesů. Pojem náhodného procesu vychází z pojmu náhodné proměnné. Náhodná proměnná je zobrazení výsledku experimentu do množiny reálných čísel. Tímto způsobem lze získat číselné informace o daném problému. Náhodný proces je pak uspořádaná množina náhodných proměnných. Nejběžnějším způsobem uspořádání množiny náhodných proměnných je přiřazení každé z nich k odlišnému časovému okamžiku, což vede ke společné interpretaci náhodného procesu jako množiny náhodných proměnných uspořádaných v čase. [3]

Hlubší implikace plyne z rozdílu mezi deterministickými a náhodnými signály. Chování deterministických signálů je zcela předvídatelné, zatímco chování náhodných signálů je spojeno s určitou mírou nejistoty. Aby byl tento rozdíl zřetelnější, je třeba vyjít z předpokladu, že známe všechny minulé hodnoty signálu až do současnosti. Pokud je možno pomocí minulých hodnot přesně předpovědět budoucí hodnoty signálu, pak takový signál je deterministický. Na druhou stranu, pokud nelze přesně předpovědět budoucí hodnoty signálu, označuje se takový signál jako náhodný. V praxi není rozdíl mezi těmito dvěma typy signálu příliš zřetelný, protože každý signál je zkreslen určitým množstvím nežádoucího náhodného šumu. Nicméně pro šetření matematické reprezentace signálů bylo široce přijato dělení na deterministické a náhodné signály. Deterministické signály mohou být popsány, alespoň v zásadě, matematickými funkcemi. Tyto funkce mohou mít často tvar explicitních matematických vzorců. [3]

Existují však deterministické signály, které nelze popsat jednoduchými rovnicemi. V zásadě lze říct, že každý deterministický signál je popsán funkcí $x(t)$, i když není k dispozici explicitní matematický vzorec. Naproti tomu náhodné signály nemohou být popsány matematickými funkcemi, protože jejich budoucí hodnoty nejsou známy. Matematické nástroje pro reprezentaci a analýzu náhodných signálů se proto liší od nástrojů používaných pro deterministické signály. Přesněji řečeno, náhodné signály jsou studovány pomocí pojmů a technik z teorie pravděpodobnosti a statistiky.

Reprezentace náhodného procesu v transformační (frekvenční) doméně není přímočará. Přímé použití Fourierových transformací na náhodný proces nedává velký smysl, protože v takovém případě je tvar vlny $x(n)$ z definice neznámý. Kromě toho některé z náhodných procesů (např. náhodný stacionární signál) představují nekonečnou energii, což vyžaduje určité specifické matematické zpracování. To však neznamená, že neexistují žádné transformační nástroje pro analýzu náhodných signálů. Analýza náhodných signálů může využívat transformační nástroje mezi časovou a frekvenční oblastí. Frekvenční reprezentace autokorelační funkce se používá k popisu vztahu vstup – výstup u lineárního systému, kde je zpracování náhodných signálů velmi užitečné. [1]

1.5 Výkonová spektrální hustota

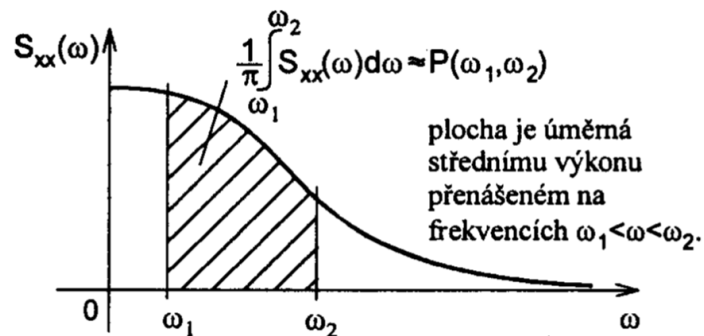
Tzv. funkce výkonové spektrální hustoty (PSD) $S_{xx}(\omega)$ je definována jako Fourierova transformace autokorelační funkce daného náhodného procesu $\{X\}$. Pro proces WSS (wide-sense stationary) platí [1]

$$S_{xx}(\omega) = \int_{-\infty}^{\infty} R_{xx}(\tau) e^{-j\omega\tau} d\tau, \quad (5)$$

pak

$$R_{xx}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_{xx}(\omega) e^{j\omega\tau} d\omega. \quad (6)$$

Výkonová spektrální hustota se používá k popisu náhodného procesu – udává rozložení energie na jednotlivé frekvence harmonických složek obsažených v realizaci náhodného procesu.



Obr. 3. Výkonová spektrální hustota.

Rovnice (5) a (6) jsou společně označovány jako Wiener-Khinchinův teorém pro signály s diskrétním časem. Náhodný proces se nazývá wide-sense stationary (WSS), pokud jeho střední hodnota a autokorelační funkce mají následující vlastnosti: [1]

$$E\{X(n)\} = c, \quad \forall n, \quad (7)$$

$$R_{xx}(n, n + \tau) = R_{xx}(\tau), \quad \forall n, \tau. \quad (8)$$

Z rovnice (7) plyne, že střední hodnota všech náhodných proměnných $X(n)$ je během celého procesu konstantní. Vztah v rovnici (8) poskytuje informaci o tom, že autokorelační funkce procesu WSS závisí pouze na časovém intervalu mezi dvěma náhodnými proměnnými, nikoli na jejich absolutních časech. Celkově tyto dva vztahy ukazují, že statistiky

prvního a druhého řádu náhodného procesu se v průběhu času nemění, což ukazuje na (wide-sense) stacionární povahu procesu z hlediska statistiky. Pokud jsou však statistiky všech řádů časově neměnné, je tento proces označován jako striktně stacionární (strict-sense). Procesy SSS jsou tedy také WSS, zatímco opak nemusí být nutně pravdivý. Protože je velmi obtížné ověřit vlastnost neměnnosti pro všechny řády, v praxi se často pracuje s charakterizací WSS, která vyžaduje pouze neměnnost prvního a druhého řádu. Proces WSS lze dobře charakterizovat také ve frekvenční oblasti.

Odhad PSD se používá v mnoha případech, jako např. u radaru, sonaru nebo modelování zvuku atd. Obecně je problém odhadu nejprve řešen odhadem autokorelační funkce z dostupných dat a poté Fourierovou transformací. Tato metoda je založena na Wiener-Khinchinově teorému. Existují různé algoritmy pro odhad spektra. Každý z nich má odlišné charakteristiky, pokud jde o výpočetní složitost, přesnost, frekvenční rozlišení a další statistické aspekty. Všechny metody lze rozdělit do dvou tříd: parametrické a neparametrické. Obecně parametrické metody vychází z toho, že signál lze popsat pomocí parametrického modelu. V našem případě budeme dále uvažovat parametrický autoregresní model. Nejprve jsou odhadnuty parametry tohoto modelu (slouží k modelování signálu) a pak se provede odhad spektra. Neparametrické metody odhadují výkonové spektrum přímo z navzorkovaného signálu, odhad je založen na konečném počtu vzorků signálu.

1.6 Okénkování

Pomocí DFT lze zpracovat N hodnot vzorků číslicového signálu a získat N hodnot vzorků spektrální funkce. Je-li originální posloupnost delší než N , pak se dá říct, že výběrem N vzorků pro DFT dochází k násobení původní delší posloupnosti obdélníkovým oknem, čímž ve spektru vzniká konvoluce mezi spektrem originální posloupnosti a spektrem okna. Důsledkem je zkreslení, tzv. prosakování ve spektru (spectral leakage). Násobení obdélníkovým oknem není z hlediska frekvenčních vlastností vhodné, proto byla navržena jiná okna (váhovací funkce), které mají lepší spektrální vlastnosti a jejich násobením s původní posloupností (případně také zvýšením počtu N) je možné dosáhnout potlačení prosakování ve spektru.

Následuje přehled nejvýznamnějších nebo v této práci použitých oken.

Obdélníkové okno

$$w[n] = \begin{cases} 1 & n \in \langle 0, N-1 \rangle, \\ 0 & \text{pro ostatní } n. \end{cases} \quad (9)$$

Bartlettovo (trojúhelníkové) okno

$$w[n] = \begin{cases} \frac{2n}{N-1} & n \in \langle 0, \frac{N-1}{2} \rangle, \\ 2 - \frac{2n}{N-1} & n \in \langle \frac{N-1}{2}, N-1 \rangle. \end{cases} \quad (10)$$

Hannovo okno

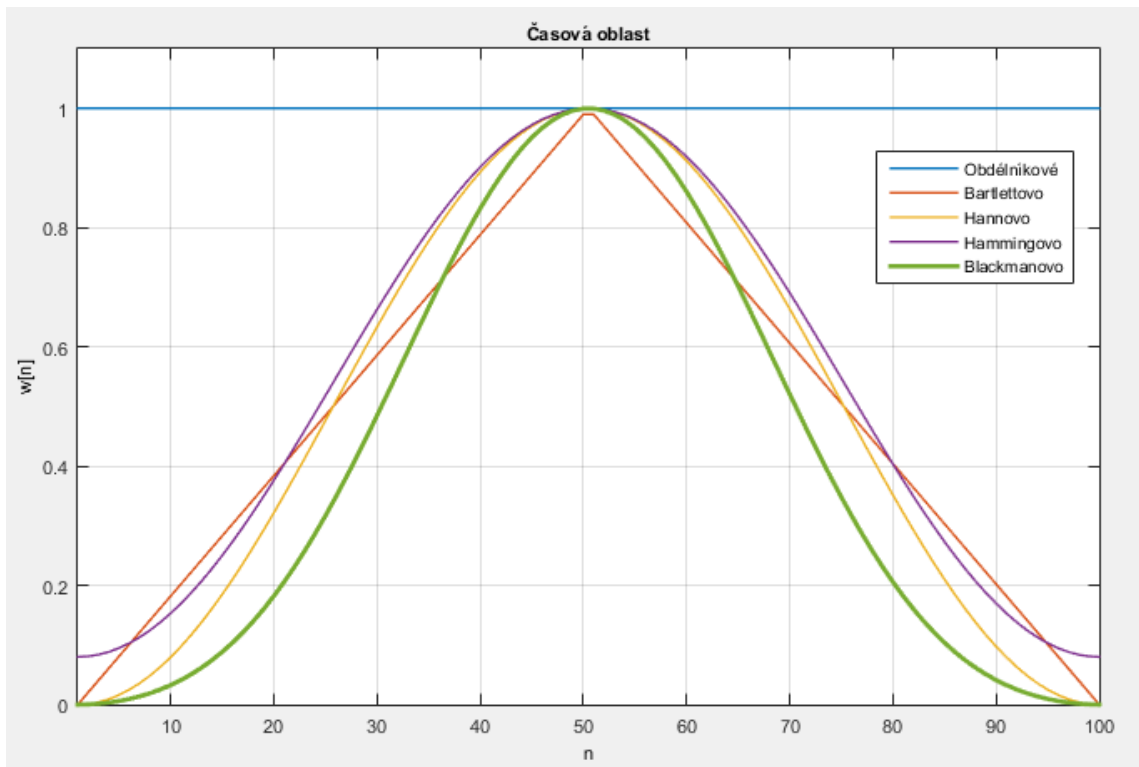
$$w[n] = 0,5 \left[1 - \cos \frac{2\pi n}{N-1} \right] \quad n \in \langle 0, N-1 \rangle. \quad (11)$$

Hammingovo okno

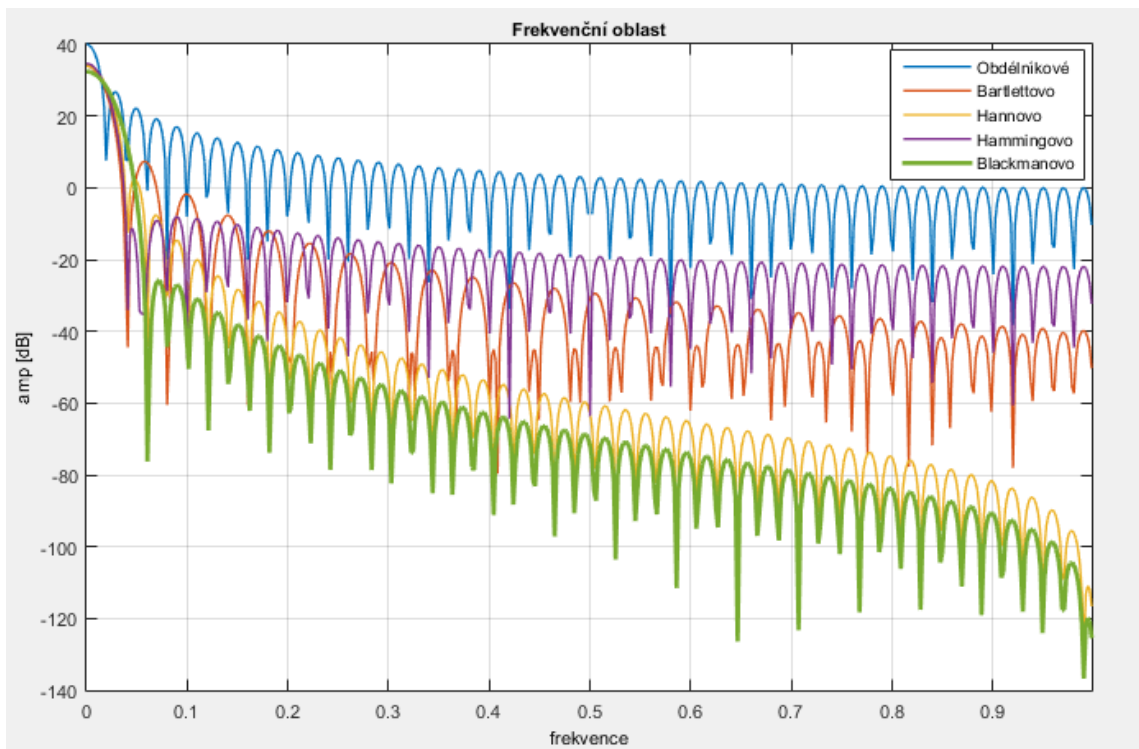
$$w[n] = 0,54 - 0,46 \cos \frac{2\pi n}{N-1} \quad n \in \langle 0, N-1 \rangle. \quad (12)$$

Blackmanovo okno

$$w[n] = 0,42 - 0,5 \cos \frac{2\pi n}{N-1} + 0,08 \cos \frac{4\pi n}{N-1} \quad n \in \langle 0, N-1 \rangle. \quad (13)$$



Obr. 4. Zobrazení oken v časové oblasti.



Obr. 5. Zobrazení oken ve frekvenční oblasti.

2 METODY PRO ODHAD SPEKTRA - TEORIE

2.1 Spektrální analýza

Tato kapitola se zabývá problémem odhadu PSD daného diskrétního signálu s diskrétním časem $x(n)$. Potřeba odhadu výkonové spektrální hustoty vyvstává v celé řadě aplikací, včetně měření korelací a výkonových spektrálních hustot pro návrh optimálních filtrů, detekce úzkopásmových signálů v širokopásmovém šumu, odhad parametrů systémů LTI (lineární časově invariantní) pomocí použití excitace bílého šumu a extrakce informací o fyzickém systému sledováním nepřítomnosti nebo přítomnosti energie ve specifických frekvenčních pásmech. [3] Obecně je problém často vyřešen nejprve odhadem autokorelační funkce spojené s daty, které máme tzv. po ruce, následován Fourierovou transformací pro získání požadovaného spektrálního popisu procesu. [1] Fourierova řada a Fourierova transformace nám umožňují zobrazit deterministické časové funkce jako vážený součet nebo integrál sinusových funkcí. U časové funkce, která se mění pomalu, mají vyšší váhy nízkofrekvenční sinusové složky. U časové funkce, která se mění rychle, mají vyšší váhy složky s vyššími frekvencemi. Rychlost, s jakou se deterministická časová funkce mění, tedy souvisí s váhovou funkcí Fourierovy řady nebo transformace. Tato váhová funkce se nazývá „spektrum“ časové funkce. [2]

Existuje několik algoritmů pro získání spektrálního odhadu. Každý z nich má odlišné vlastnosti s ohledem na výpočetní složitost, přesnost, rozlišení kmitočtu nebo jiné statistické aspekty. Všechny algoritmy lze klasifikovat jako neparametrické nebo parametrické metody. Neparametrické metody nepředpokládají žádnou konkrétní strukturu za dostupnými daty, zatímco parametrická schémata jsou založena na předpokladu, že proces sleduje určitý vzorec charakterizovaný specifickou sadou parametrů vztahujících se k danému modelu. Obecně platí, že parametrické přístupy bývají jednodušší a přesnější, ale záleží na a priori informacích týkajících se daného problému. [1]

2.2 Neparametrické metody pro odhad spektra

Neparametrický odhad spektra se opírá o Wiener-Khinchinův teorém, která definuje vztah mezi autokorelační funkcí procesu a jeho Fourierovou transformací, která odpovídá PSD procesu. Tato část vysvětluje, jak odhadnout PSD nebo autokorelační sekvenci ergodického procesu WSS (wide-sense stationary) $\{X\}$ z omezeného počtu naměřených dat. Takový odhad je ve skutečnosti náhodný proces $\{P\}$, odlišný od původního náhodného procesu

$\{X\}$. V praxi, kdykoli je proveden odhad, je vypočítána realizace tohoto náhodného procesu $\{P\}$ pomocí dostupné realizace náhodného procesu $\{X\}$. Ke zdůraznění jsou použita malá písmena $x(n)$ k reprezentaci vzorků procesu $\{X\}$. [1]

2.2.1 Metoda periodogramu

Nechť $x(0), x(1), \dots, x(L-1)$ jsou L pozorování diskrétního, nulově středního, širokopásmového stacionárního procesu. Odhad periodogramu pro $S_{xx}(\omega)$ je definován jako: [2]

$$\hat{S}_{xx}(\omega) = \frac{1}{L} |\hat{x}_L(\omega)|^2,$$

kde $\hat{x}_L(\omega)$ se získá jako Fourierova transformace sekvence pozorování:

$$\hat{x}_L(\omega) = \sum_{n=0}^{L-1} x(n) e^{-j\omega n}.$$

To je ekvivalentní použití obdélníkového okna na signál $x(n)$ pro časový interval $0 \leq n \leq (L-1)$, umocnění absolutní hodnoty Fourierovy transformace zkrácené sekvence a normalizace výsledku prvkem L , pro získání míry hustoty spektrální energie. Na druhé straně Fourierova transformace $S_{xx}(\omega)$ autokorelační funkce $R_{xx}(\tau)$ širokospektrálního stacionárního procesu (WSS) $x(n)$ se nazývá výkonová spektrální hustota (PSD) procesu. [3]

Je třeba odhadnout funkci $S_{xx}(\omega)$ z konečné sady pozorování $\{x(n), 0 \leq n \leq L-1\}$ jedné realizace. Přirozený odhad PSD lze získat nahrazením skutečné $R_{xx}(\tau)$ odhadem $\hat{R}_{xx}(\tau)$, pro všechna dostupná zpoždění. Výsledkem je následující odhad [3]

$$\hat{S}_{xx}(\omega) = \sum_{\tau=-(L-1)}^{L-1} \hat{R}_{xx}(\tau) e^{-j\omega\tau}, \quad (14)$$

kde odhad autokorelační funkce je

$$\hat{R}_{xx}(\tau) = \frac{1}{L} \sum_{n=0}^{L-1-|\tau|} x(n)x(n+|\tau|). \quad (15)$$

Protože $\hat{R}_{xx}(\tau)$ v nedostupných zpožděních je libovolně nastavena na nulu, tento odhad bude dávat velké chyby v případě, kdy se $R_{xx}(\tau)$ výrazně liší od nuly pro $|\tau| > L$ [3].

Očekávalo by se, že při zvyšování počtu vzorků L odhad periodogramu bude konvergovat k $S_{xx}(\omega)$. K tomu však nedojde. Místo toho bylo zjištěno, že prudce kolísá okolo skutečné spektrální hustoty a že tento náhodný rozptyl neklesá s rostoucím L .

2.2.2 Metoda Bartlett

Základním výsledkem v teorii pravděpodobnosti je to, že průměr sekvence nezávislých realizací náhodných proměnných se přibližuje skutečnému průměru s pravděpodobností jedna. Je získán odhad pro $S_{xx}(\omega)$ který směřuje k nule s počtem pozorování $K = L/k$ tak, že se vezme průměr k nezávislých periodogramů na vzorcích velikosti K : [2]

$$\langle \hat{S}_{xx}(\omega) \rangle_K = \frac{1}{k} \sum_{i=1}^k \hat{S}_{xx,i}(\omega).$$

Sada vzorků o velikosti L je rozdělena na k bloků a pro každý blok je vypočítán samostatný periodogram. Tzv. vyhlazený (smoothed) odhad je pak průměr z k periodogramů. Tato metoda se nazývá Bartlett's smoothing procedure. Zde je nutno připomenout, že obecně nejsou výsledné periodogramy nezávislé, protože základní bloky nejsou nezávislé. Tento vyhlazovací postup je tedy třeba chápat jako aproximaci výpočtu a průměrování nezávislých periodogramů. [2] Tento přístup však snižuje množství dat použitých v každém odhadu a snižuje související spektrální rozlišení. V tomto smyslu pak metoda průměrovaného periodogramu vymění snížení rozptylu na úkor nižšího rozlišení ve výsledném odhadu PSD. [1]

2.2.3 Metoda Blackman-Tukey

Protože pro každou hodnotu ω představuje odhadce $S_{xx}(\omega)$ náhodnou proměnnou, může být nepravidelné chování periodogramu vysvětleno pomocí jeho průměru, rozptylu a kovariance. [3]

Pokud se vezmou v úvahu rovnice (14) a (15), pak pro střední hodnotu odhadu platí: [3]

$$E[S_{xx}(\omega)] = \sum_{\tau=-(L-1)}^{L-1} E(\hat{R}_{xx}(\tau)) e^{-j\omega\tau} = \sum_{\tau=-(L-1)}^{L-1} \left(1 - \frac{|\tau|}{L}\right) R_{xx}(\tau) e^{-j\omega\tau}.$$

Protože $E[S_{xx}(\omega)] \neq S_{xx}(\omega)$, periodogram dává zkreslený odhad $S_{xx}(\omega)$. Nicméně, pro každé $|\tau|$ však prvek $(1 - |\tau|/L) \rightarrow 1$ stejně jako $N \rightarrow \infty$. Výsledkem je

$$\lim_{L \rightarrow \infty} E[S_{xx}(\omega)] = S_{xx}(\omega),$$

to znamená, že periodogram je asymptoticky nezkreslený odhadce $S_{xx}(\omega)$.

Pokud je definováno trojúhelníkové nebo Bartlettovo okno (10) s nulovou fází a je použit teorém kmitočtové konvoluce, pak

$$E[S_{xx}(\omega)] = \sum_{\tau=-(L-1)}^{L-1} w_B(\tau) R_{xx}(\tau) e^{-j\omega\tau} = \frac{1}{2\pi} \int_{-\pi}^{\pi} S(\theta) W_B(\omega - \theta) d\theta.$$

Očekávaná hodnota periodogramu se tedy získá konvolucí skutečné PSD $S_{xx}(\omega)$ s Fourierovou transformací $W_B(\omega)$ Bartlettova okna. Aby $E[S_{xx}(\omega)]$ byl co nejbližší k $S_{xx}(\omega)$, $W_B(\omega)$ by měl být blízkou aproximací k jednotkovému impulzu $\delta(\omega)$. Hlavní lalok $W_B(\omega)$ má přibližnou šířku pásma 3 dB při $2\pi/N$ rad. Proto $W_B(\omega)$ je poměrně špatná aproximace $\delta(\omega)$ pro malé hodnoty L ; nicméně $W_B(\omega)$ se blíží k periodickému sledu impulsů, když L se blíží nekonečnu. Průměrný periodogram $E[S_{xx}(\omega)]$ se přibližuje asymptoticky, když $L \rightarrow \infty$, skutečné PSD $S_{xx}(\omega)$. [3]

K pochopení toho, jak Bartlettovo okno vstupuje do výpočtu $E[S_{xx}(\omega)]$, se definuje modifikovaný periodogram

$$\hat{S}_{xx}(\omega) = \frac{1}{L} \left| \sum_{n=0}^{L-1} w(n)x(n)e^{-j\omega n} \right|^2, \quad (16)$$

kde $w(n)$ je datové okno s dobou trvání L . Účelem použití datového okna je snížení prosakování ve spektru způsobeného silnými úzkopásmovými součástmi snížením úrovně postranních laloků. Pokud je signál v okně definován

$$v(n) = w(n)x(n),$$

lze rovnici (16) vyjádřit jako

$$\hat{S}_{vv}(\omega) = \sum_{\tau=-(L-1)}^{L-1} \hat{R}_{vv}(\tau) e^{-j\omega\tau},$$

kde

$$\hat{R}_{vv}(\tau) = \frac{1}{L} \sum_{n=0}^{L-\tau-1} w(n+\tau)w(n)x(n+\tau)x(n),$$

což je zobecnění rovnice (14) na libovolná okna.

2.2.4 Metoda nezkresleného odhadu

Je možné se vyhnout zkreslení v autokorelaci periodogramu použitím $(L - |\tau|)$ namísto L ve jmenovateli rovnice (15), čímž se získá nový odhad autokorelace [1]

$$\hat{R}_{xx}(\tau) = \frac{1}{L - |\tau|} \sum_{n=0}^{L-1-|\tau|} x(n)x(n + |\tau|).$$

2.2.5 Metoda Blackman–Tukey 2

Základ této metody je podobný jako metoda Blackman – Tukey, ale v této metodě namísto vynásobení vzorků oknem a poté získáním autokorelační funkce, se nejprve získá autokorelační funkce vzorků signálu a až poté se výsledná funkce vynásobí oknem.

2.2.6 Metoda Welch

Metoda Welch rozšiřuje Bartlettův přístup překrýváním segmentů a okénkováním každého segmentu. Účelem použití datového okna je snížení spektrálního průsaku způsobeného silnými úzkopásmovými součástmi snížením úrovně postranních laloků. Překrývající se segmenty přinášejí další redukci rozptylu kvůli zvýšenému počtu průměrovaných periodogramů. M (možná překrývající se) segmenty jsou definovány pomocí [3]

$$x_m(n) = x(n + (m - 1)D), \quad 0 \leq n \leq L - 1,$$

kde D je offsetová vzdálenost. Pokud $D < L$, pak se segmenty překrývají; a pro $D = L$ segmenty sousedí. Upravené periodogramy jsou dány

$$\hat{S}_{mm}(\omega) = \frac{1}{L} \left| \sum_{n=0}^{L-1} w(n)x_m(n)e^{-j\omega n} \right|^2, \quad (17)$$

kde $w(n)$ je datové okno s dobou trvání L , které nemusí být symetrické. Welchův odhad je dán průměrem M modifikovaných periodogramů

$$\hat{S}_{xx}(\omega) = \frac{1}{M} \sum_{n=0}^{L-1} \tilde{S}_{mm}(\omega). \quad (18)$$

Welch ukázal, že tvar okna může snížit průsak, ale nemá vliv na rozptyl. Překrývání segmentů o padesát procent však snižuje rozptyl asi dvakrát a to v důsledku zdvojnásobení počtu segmentů. Další překrývání nevede k další redukci rozptylu, protože datové segmenty jsou stále méně nezávislé. Je zřejmé, že nepřekrývající se segmenty mohou být zcela nekorelované (lineárně nezávislé) pouze pro procesy s bílým šumem. Datové segmenty však lze považovat za přibližně nekorelované, pokud nemají ostré spektrální vrcholy nebo pokud jejich autokorelační sekvence rychle klesá. [3]

2.2.7 Metoda minimálního rozptylu

Dosud prezentované metody neparametrického spektrálního odhadu jsou založeny na periodogramu popsaném v rovnici (14). Tato rovnice naznačuje, že na signál $x(n)$ umístěný v časovém intervalu $0 \leq n \leq (L - 1)$, se aplikuje obdélníkové okno, poté se vypočítá jeho Fourierova transformace a následně se provede konečný výpočet. [1]

Přístup s minimálním rozptylem odhaduje spektrum při frekvenci ω_c filtrováním signálu úzkým pásmovým filtrem vystředěným na ω_c a odhadováním výkonu na výstupu z filtru. Filtr by měl být optimalizován tak, aby měl minimální energii mimo přeložené pásmo. [1]

Podle zdroje [1] vypadá metoda MV následovně

$$\hat{S}_{xx}(\omega) \approx \frac{L}{e^{*T}(e^{j\omega})R_X^{-1}e(e^{j\omega})'}$$

kde v

$$e(e^{j\omega}) = [1 e^{-j\omega_c} e^{-j2\omega_c} \dots e^{-j(L-1)\omega_c}]^T,$$

$$x(n) = [x(n) x(n-1) \dots x(n-L+1)]^T,$$

$$R_X = E[x(n)x^{*T}(n)].$$

Jak je známo z výše uvedených rovnic, pro odhad výkonové spektrální hustoty je zapotřebí inverzní matice R_X^{-1} . Ale protože matice R_X^{-1} je velká s ohledem na počet vektorů x , vyžaduje její inverzní výpočet velké množství výpočetních prostředků. Proto se používá následující metoda, ve které se nemusí počítat inverzní matice

$$\hat{S}_{xx}(\omega) = \frac{L}{\sum_{l=0}^{L-1} |e^{*T}(e^{j\omega})q_l|^2 / \lambda_l}, \quad (19)$$

kde R_X^{-1}

$$R_X^{-1} = \sum_{l=0}^{L-1} \frac{q_l q_l^{*T}}{\lambda_l}.$$

2.2.8 Metoda diskrétní Fourierovy transformace

Teorie potřebná k pochopení metody DFT již byla probrána v předchozích částech této práce a to především v sekci (1.3).

2.3 Parametrické metody pro odhad spektra

Spektrální odhady založené na periodogramu provádějí primární odhad autokorelační funkce pomocí Wiener-Khinchinova teorému. V takovém případě je odhad autokorelace omezen na interval zpoždění vymezený množstvím dostupných dat. Mimo tento interval se odhad stává nulovým, což vede k zkreslenému odhadu PSD.

Pomocí parametrického spektrálního odhadu se nejprve modeluje náhodný proces a výsledný odhad PSD se získá jako výkonové spektrum tohoto modelu. [1]

Tato část popisuje klasický přístup, běžně označovaný jako lineární predikce (LP), k odhadu parametrů AR (autoregresního) modelu pro konkrétní datový soubor.

2.3.1 Metoda Burg

Zde se vyjde ze znalosti souboru diskrétních dat $\{x(0) x(1) \dots\}$ z konkrétního procesu. Tato data mohou pocházet z měření ceny akcií, některé populace bakterií nebo například řečového signálu. Myšlenkou problému LP je odhadnout hodnotu $x(n)$ jako lineární kombinaci N minulých vzorků procesu; to je:

$$\hat{x}(n) = \hat{a}_1 x(n-1) + \hat{a}_2 x(n-2) + \dots + \hat{a}_N x(n-N) = \sum_{i=1}^N \hat{a}_i x(n-i),$$

kde \hat{a}_i pro $i = 1, 2, \dots, N$, jsou tzv. LP koeficienty a N je pořadí modelu LP. Pro odhad \hat{a}_i se použije Burgova metoda.

Chyba odhadu mezi $x(n)$ a optimálním odhadem LP je také známá jako chyba dopředného odhadu $N - \text{tého}$ řádu a je vyjádřena

$$x_{f,[N]}(n) = x(n) - \sum_{i=1}^N \hat{a}_{i,[N]}^* x(n-i).$$

Pokud je model LP naplněn verzí procesu $\{X\}$ v převráceném čase, budou budoucí vzorky zkombinovány a vytvoří odhad $x(n-N)$. Protože $R_X(\tau)$ je sudá funkce, odpovídající Wiener-Hopfova rovnice a následně optimální LP model zůstávají stejné. V takovém případě se chyba mezi minulým vzorkem $x(n-N)$ a jeho odhadem nazývá jako zpětná chyba LP a označuje se jako [1]

$$x_{b,[N]}(n) = x(n-N) - \sum_{i=1}^N \hat{a}_{i,[N]}^* x(n-N+i).$$

Burgova metoda určuje koeficienty odrazu k_i , Po úpravě se získá

$$k_i = \frac{2 \sum_{n=i}^{L-1} x_{b,[i-1]}(n-1) x_{f,[i-1]}(n)}{\sum_{n=i}^{L-1} (x_{b,[i-1]}^2(n-1) + x_{f,[i-1]}^2(n))}. \quad (20)$$

Tento odhad se používá při Levinson-Durbin rekurzích následujícím způsobem

$$\hat{a}_{j,[i]}^* = \begin{cases} \hat{a}_{j,[i-1]}^* - k_i \hat{a}_{i-j,[i-1]}^* & j = 1, 2, \dots, (i-1) \\ k_i & j = i \end{cases}, \quad (21)$$

k určení odpovídajícího LP modelu a požadovaného odhadu PSD. K simulaci této metody je třeba upravit odhady vztahů $\hat{a}_{j,[i]}^*$ a k_i do vektorové formy a použít následující algoritmus

1. Inicializace.

$$i = 0,$$

$$x_{f_0} = x_{b_0} = [x(1), \quad x(2), \dots, x(L)].$$

2. Pro $i = 1:N$.
3. Odhadne se $\hat{a}_{j,[i]}^*$ s využitím rovnice (20).
4. Odstraní se poslední prvek x_{f_i} a první prvek x_{b_i}

$$xf_i = xf_i(1:L - i),$$

$$xb_i = xb_i(i + 1:L).$$

5. Vypočítají se koeficienty odrazu

$$k_i = -\frac{2 xb_i^H xf_i}{xb_i^H xb_i + xf_i^H xf_i}.$$

6. Aktualizují se chyby predikce

$$xf_{i+1} = xf_i + k_i xb_i,$$

$$xb_{i+1} = xb_i + k_i xf_i.$$

7. Zpět na krok 2.

II. PRAKTICKÁ ČÁST

3 METODY PRO ODHAD SPEKTRA - ALGORITMY

V této části jsou v kompaktní formě zobrazeny a popsány algoritmy, které slouží pro výpočet odhadu PSD v prostředí MATLAB. Zároveň jsou zobrazeny průběhy odhadu PSD pro každou metodu.

3.1 Metoda periodogramu

Existuje několik způsobů, jak vypočítat periodogram. Zde jsou vypočítány rovnice (14) a (15) potřebné pro získání periodogramu, tj. odhad výkonové spektrální hustoty pomocí autokorelační funkce. K tomu je nejprve třeba získat autokorelační funkci (15), a protože autokorelační funkce je sudá – symetrie kolem vertikální osy ($\hat{R}_{xx}(\tau = 0)$) (tato vlastnost je z důvodu $|\tau|$ v rovnici (15)), pokud je vypočtena jedna strana, je druhá získána symetricky. Z tohoto důvodu je vypočtena v MATLAB kódu pouze pravá strana autokorelační funkce a následně je získána levá strana pomocí symetrie.

Je tedy potřeba smyčka `for`, která posune vzorek $x(n + |\tau|)$ pro $\tau = 1:L$ a poté vypočítá autokorelační funkci pro tuto hodnotu τ (ve skutečnosti by τ měla být $0:L-1$, ale v programu MATLAB začínají proměnné od 1). To se provede pomocí následujícího kódu:

```
for i = 1:L % výpočet pravé strany AF
    % posun signálu doleva (složení signálu z původního signálu a nul)
    xtau = [x(i:end) zeros(1,i-1)];
    % součin původního a posunutého (zpožděného) signálu
    rx = xtau .* x;

    Rx(i) = sum(rx);
end
```

Kód je možné psát bez použití příkazu `sum`:

```
for i = 1:L % výpočet pravé strany AF
    % posun signálu doleva (složení signálu z původního signálu a nul)
    xtau = [x(i:end) zeros(1,i-1)];
    % součin původního a posunutého (zpožděného) signálu
    rx = xtau .* x;
    s = 0;
```

```

for j = 1:L % suma rx
    s = s + rx(j);
end
Rx(i) = s;
end

```

Nyní je třeba zrcadlit pravou stranu autokorelační funkce, aby se získala levá strana funkce a pak se spojí obě strany a vydělí pomocí L . To se děje pomocí následujících příkazů:

```

for i = 1:L-1
    Rx1(i) = Rx(L-i+1); % získání levé strany AF ze strany pravé
end
Rx = [Rx1 Rx] ./L; % vytvoření AF z pravé a levé strany a dělení pomocí L

```

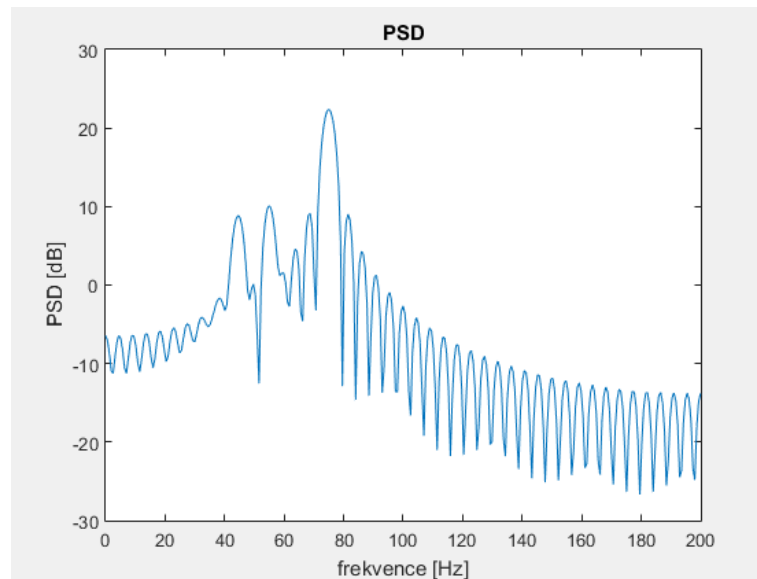
Byla získána autokorelační funkce a nyní je možné se dostat k periodogramu pomocí Fourierovy transformace (14). K získání Fourierovy transformace lze použít vestavěnou MATLAB funkci `fft`, ale místo toho se využije následující kód, který dělá totéž.

```

for i = 1:length(w) % smyčka pro frekvenci
    psd = 0;
    for j = 1:2*L-1 % smyčka pro sumu
        psd = psd + Rx(j) * exp(-1i*w(i)*nu(j));
    end
    PSD(i) = abs(psd)/pi;
End

```

Tento kód v rovnici (14) pro každé ω , ($\omega = 0:0.01:\pi$), vynásobí autokorelační funkci s $e^{-j\omega\tau}$ a sečte výsledné vzorky. Vnější `for` smyčka se používá pro výpočet hodnot $\hat{S}_{xx}(\omega)$ v daném ω a vnitřní `for` smyčka pro součet vzorků. Zde je nutno připomenout, že pomocí tohoto kódu se získá pravá strana Fourierovy transformace ($\omega = 0:0.01:\pi$), protože Fourierova transformace je také symetrická kolem vertikální osy, takže získání jedné strany je dostatečné pro zjištění všech frekvenčních složek.



Obr. 6. Průběh PSD s využitím metody periodogramu.

3.2 Metoda Bartlett

Pro simulaci této metody je třeba jednoduše rozdělit vzorek signálu x do nb bloků a vypočítat autokorelační funkci pro každý z těchto bloků a poté získat Fourierovu transformaci, následně zprůměrovat odhady výkonové spektrální hustoty každého bloku. Nakonec se odhadne výkonové spektrum signálu x . To se provede přidáním smyčky `for` pro bloky signálu následovně:

```
for j = 1:nb % rozdělení signálu na bloky
    X = x((j-1)*Lb+1:j*Lb);
    ...
    Výpočet autokorelační funkce bloku x
    ...
    Rxb(:,j) % autokorelační funkce bloku j
end
```

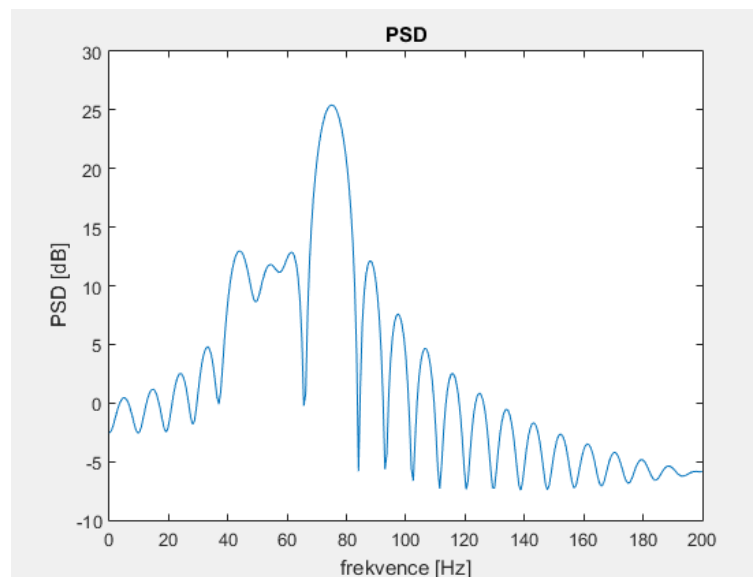
Pak je nutné vypočítat Fourierovu transformaci každého bloku. Protože je autokorelační funkce bloků zabudována do sloupců proměnné `Rxb`, je možné postupovat jako dříve a ukládat získané Fourierovy transformace do sloupců matice `PSD`.

```

for i = 1:length(w) % smyčka pro frekvenci
    psd = zeros(1,4);
    for j = 1:2*Lb-1 % smyčka pro sumu
        psd = psd + Rxb(j,:) * exp(-1i*w(i)*nu(j));
    end
    % i-tý psd vzorek každého bloku uložený ve sloupcích PSD
    PSD(i,:) = abs(psd)/pi;
end
end

```

Nyní se pro každou frekvenční složku ω vypočítá průměrný odhad PSD bloků a je považován za odhad PSD signálu $x(n)$.



Obr. 7. Průběh PSD s využitím metody Bartlett.

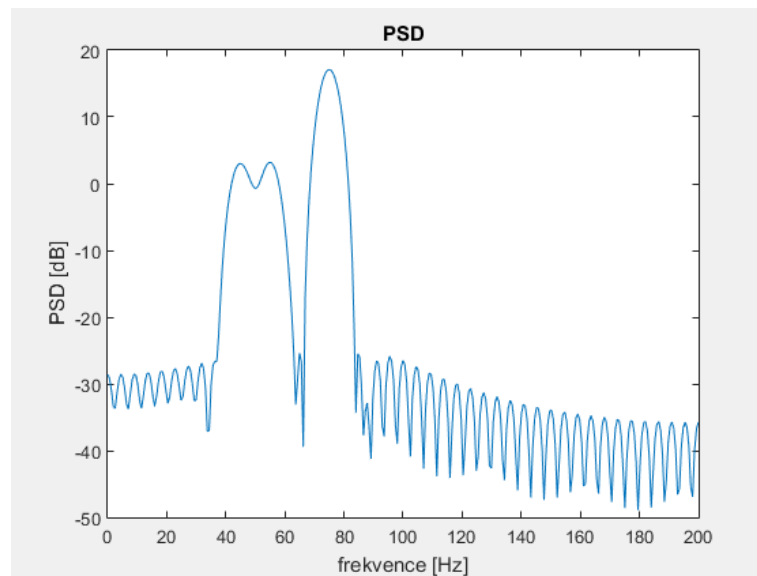
3.3 Metoda Blackman–Tukey

Ve dříve zmíněných vztazích bylo zjištěno, že na vzorky signálu je možno použít okno a poté aplikovat na periodogram pro odhad spektrální hustoty. Takže v simulaci provedené v tomto kódu se vybere Hammingovo okno, je jím vynásoben signál a kód pokračuje podobně, jako tomu bylo u periodogramu. Vytvoření okna a jeho použití na signál se provádí pomocí následujících příkazů:

```

win = 0.54 - 0.46*cos(2*pi*(0:L-1)/L); % Definice Hammingova okna
x = x .* win; % aplikace Hammingova okna na simulovaný signál - okénko-
vání

```



Obr. 8. Průběh PSD s využitím metody Blackman-Tukey.

3.4 Metoda nezkresleného odhadu

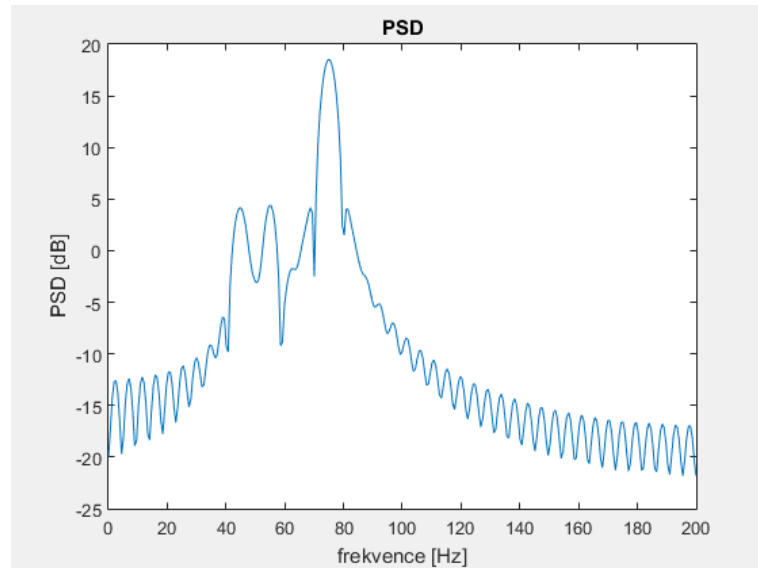
K provedení této metody se použije v simulaci namísto příkazu

```
Rx = [Rx1 Rx] ./L; % vytvoření AF z pravé a levé strany a dělení pomocí L
```

který byl použit v metodě periodogramu, následující příkaz:

```
% vytvoření AF z pravé a levé strany a dělení pomocí L-abs(nu)
```

```
Rx = [Rx1 Rx] ./(L-abs(tau));
```

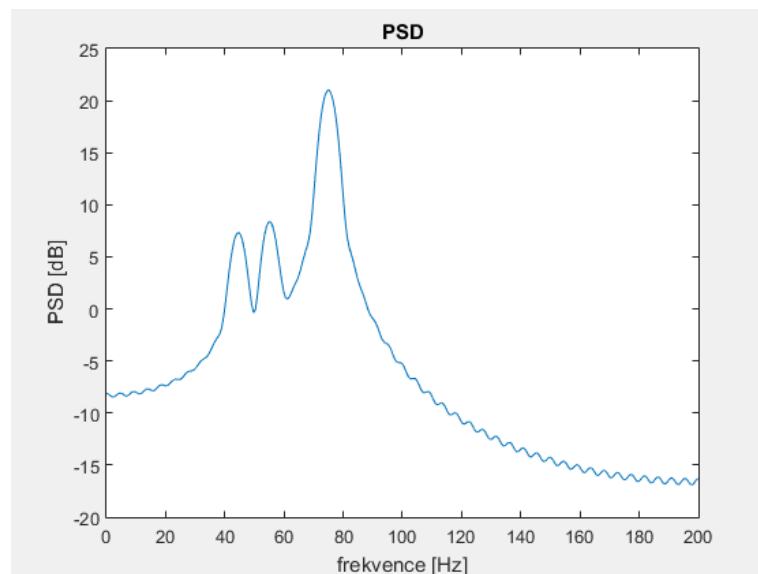


Obr. 9. Průběh PSD s využitím metody nezkresleného odhadu.

3.5 Metoda Blackman–Tukey 2

Pro simulaci je třeba získat autokorelační funkci podobně jako je tomu v metodě Blackman – Tukey, ale Hammingovo se neaplikuje na signál, ale až na autokorelační funkci a poté je použita Fourierova transformace. Provede se příkazem:

```
Rx = Rx .* win; % násobení AF Hammingovým oknem
```



Obr. 10. Průběh PSD s využitím metody Blackman-Tukey 2.

3.6 Metoda Welch

Podle dříve uvedeného vysvětlení Welchova metoda nejprve rozdělí vzorek signálu x na k segmentů. Každý segment se z padesáti procent překrývá se sousedními segmenty. Vybrané okno je Hannovo.

Simulace metody probíhá následovně. Nejprve se vybere délka okna a poté se vytvoří Hannovo okno následujícími příkazy.

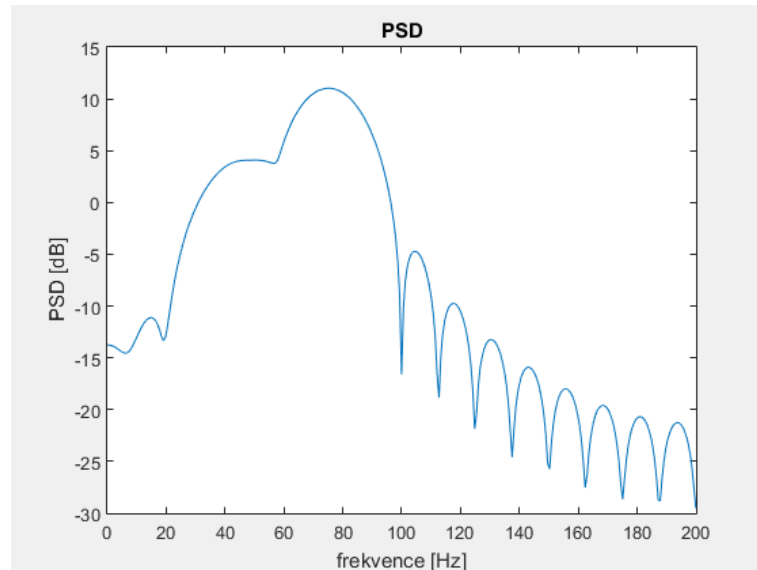
```
N = 32; % Délka Hannova okna
win = 0.5*(1-cos(2*pi*(0:N)/N)); % Definice Hannova okna
```

Dále se získá počet segmentů, je definována indexová proměnná, která se bude pohybovat po vzorcích. Indexová proměnná má počáteční hodnotu 1:32 a mění svou hodnotu ve smyčce `for`. Pokud tedy $N = 32$, indexová proměnná bere v prvním kroku 32 prvních vzorků, vynásobí s oknem a v dalším kroku oddělí vzorky 17:48 a opět vynásobí s oknem. A tak dále až do posledních vzorků (vzorky 49:80).

```
for i = 1:k
    % postupné násobení vzorků x s Hannovým oknem
    xw = win.*x(index);
    index = index + N/2; % pokračování do dalšího okna (50% překrytí)
    % rovnice (17), jde o výpočet periodogramu xw
    PSD = PSD + XW/k; % rovnice (18)
end
```

K výpočtu sumy v rovnici (17) je vnitřní smyčka zapsána následovně:

```
for j = 1:length(w) % smyčka pro frekvenci
    Xw = 0;
    for n = 1:N % smyčka pro sumu
        Xw = Xw + xw(n) * exp(-1i*w(j)*(n-1));
    end
    XW(j) = abs(Xw)/pi;
end
```



Obr. 11. Průběh PSD s využitím metody Welch.

3.7 Metoda minimálního rozptylu

K simulování metody minimálního rozptylu je třeba nejprve určit funkci R_X , takže pro výpočet prvního řádku je třeba posunout vektor vzorku $x(n)$ z $0:L-1$ a vynásobit jej se vzorkem $x(n)$. K posunu vzorků se musí vytvořit následující vektor

```
% sestavení vektoru z nul a vektoru wx a zpětná transpozice
wx_pad = [zeros(L-1,1);wy;zeros(L-1,1)].';
```

Nový vektor obsahuje nuly délky L na začátku a na konci, a vzorky x uprostřed. V každé iteraci smyčky `for` se odřízne vzorek L vektoru `wx_pad` a výsledkem je matice s $0:L-1$ posuny vektoru x .

```
for i = 1:L
    % vytvoření matice XU z vektoru wx_pad (matice s posuny vektoru x)
    XU(:,i) = wx_pad(L-i+1:3*L-i-1);
end
```

Pokud je XU násoben sám sebou, bude výsledkem matice R_X

```
% vytvoření matice Rwx násobením matic XU s XU'
Rwx = XU'*XU / (L-1);
```

Nyní se pomocí příkazu `eig` získají vlastní vektory (eigenvectors) a vlastní hodnoty (eigenvalues) matice R_x a nahradí se rovnicí (19). Pro výpočet rovnice (19) se zapíše tři smyčky `for`. Vnitřní `for` smyčka níže spočítá $e^{*T}(e^{j\omega})q_l$

```
for k = 1:L
    % výpočet e(e^jw)*q_l z rovnice (19)
    u = u + eigvec(k,l) * exp(1i*(k-1)*w(j));
end
```

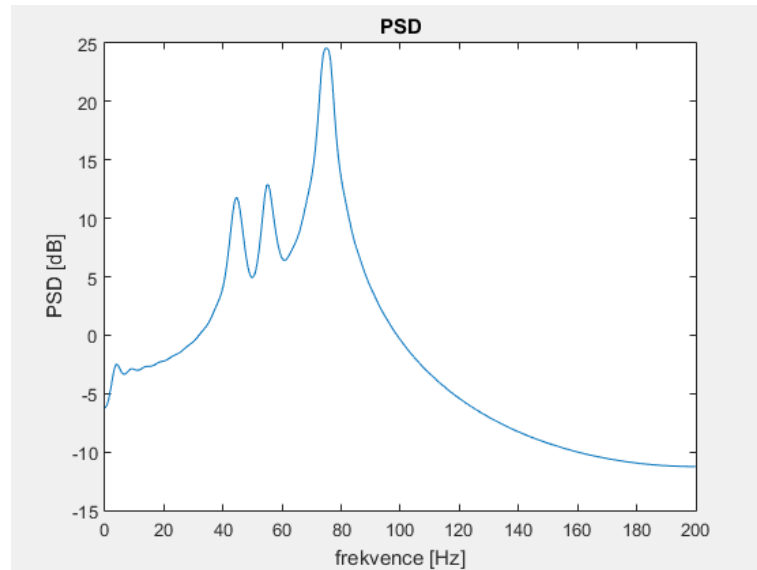
A prostřední `for` smyčka vypočítá jmenovatel rovnice (19).

```
for l = 1:L
    u = 0;
    for k = 1:L
        % výpočet e(e^jw)*q_l z rovnice (19)
        u = u + eigvec(k,l) * exp(1i*(k-1)*w(j));
    end
    % výpočet jmenovatele z rovnice (19): |e(e^jw)*q_l|^2/lambda_l
    U = abs(u).^2 ./ (eigval(l,l) + eps);
    PSD(j) = PSD(j) + U;
end
```

Tento postup se opakuje pro různé frekvence vnější `for` smyčkou.

```
% rovnice (19) pro různé frekvence
for j = 1: numel(w)

end
```



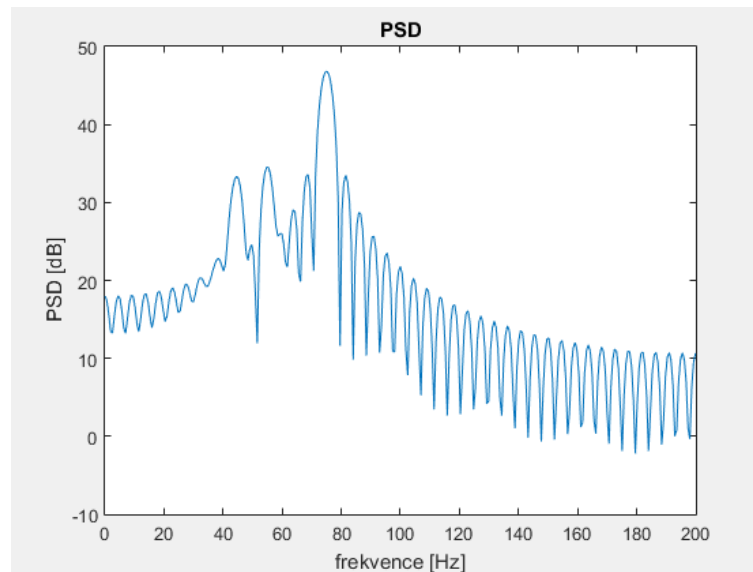
Obr. 12. Průběh PSD s využitím metody minimálního rozptylu.

3.8 Metoda Diskrétní Fourierovy transformace

K simulaci metody DFT podle rovnice (1) je potřeba pro každou frekvenci (vnější smyčka) získat sumu vzorků a $e^{j\frac{2\pi}{N}kn}$ (vnitřní smyčka). Část $\frac{2\pi}{N}k$ z výrazu $e^{j\frac{2\pi}{N}kn}$ tvoří ω v MATLAB kódu. L je počet pozorování diskrétního procesu, ω představuje frekvenční složku.

```
tau = 0:L-1; % vytvoření pole posunutí (zpoždění) o velikosti L-1
w = -0:0.01:pi; % vytvoření pole frekvencí, pro které se bude počítat PSD

% výpočet DFT
for i = 1:length(w) % smyčka pro frekvenci
    psd = 0;
    for j = 1:L-1 % smyčka pro sumu
        psd = psd + x(j) * exp(-1i*w(i)*tau(j));
    end
    DFT(i) = psd;
end
```

Obr. 13. Průběh PSD s využitím metody DFT.

3.9 Metoda Burg

Simulace prováděné v programu MATLAB v každém kroku algoritmu jsou následující.

Krok 1 a 2 pro $i = 0$ a výpočet k_1 :

```
xf = y(1:end-1);    % x_f[0]
xb = y(2:end);      % x_b[0]
k(1) = -2*(xf * xb') / (xb*xb' + xf*xf'); % rovnice (20)
```

krok 3:

```
% rovnice (21)
if i==j
    Aest(j,i) = k(i);
else
    Aest(j,i) = Aest(j,i-1) + k(i)*Aest(i-j,i-1);
end
```

krok 4:

```
xf = xf(1:end-i);  % x_f[i] (n)
xb = xb(i+1:end);  % x_b[i] (n)
```

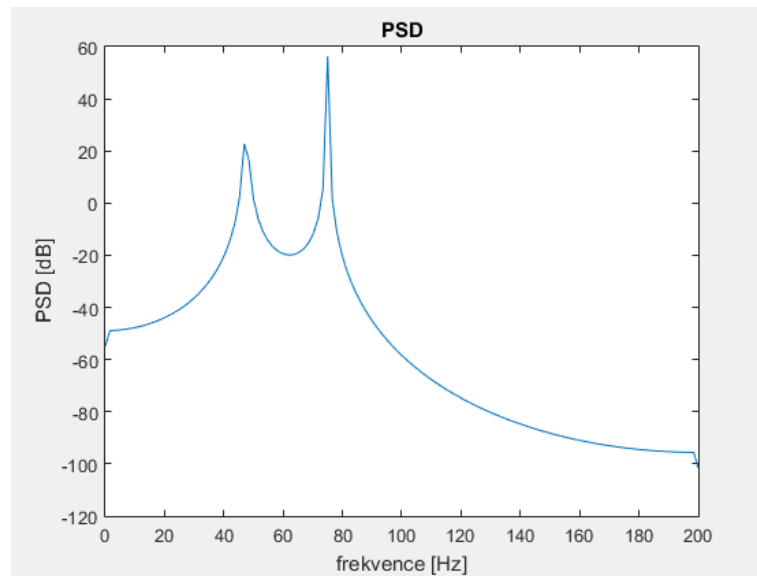
krok 5:

```
k(i+1) = -2*(xf * xb')/(xb*xb' + xf*xf'); % rovnice (20)
```

krok 6:

```
xf = xf + k(i) * xb;
```

```
xb = xb + k(i) * xf;
```



Obr. 14. Průběh PSD s využitím metody Burg.

4 METODY PRO ODHAD SPEKTRA - VYHODNOCENÍ

4.1 Individuální vyhodnocení metod

V této sekci budou vyhodnoceny všechny výše popsané metody sloužící k odhadu PSD a to především s ohledem na jejich zkreslení a rozptyl. Simulovaný signál je pro všechny metody stejný:

$$x(t) = \sin(2\pi t f_1) + \sin(2\pi t f_2) + 5\sin(2\pi t f_3), \quad (22)$$

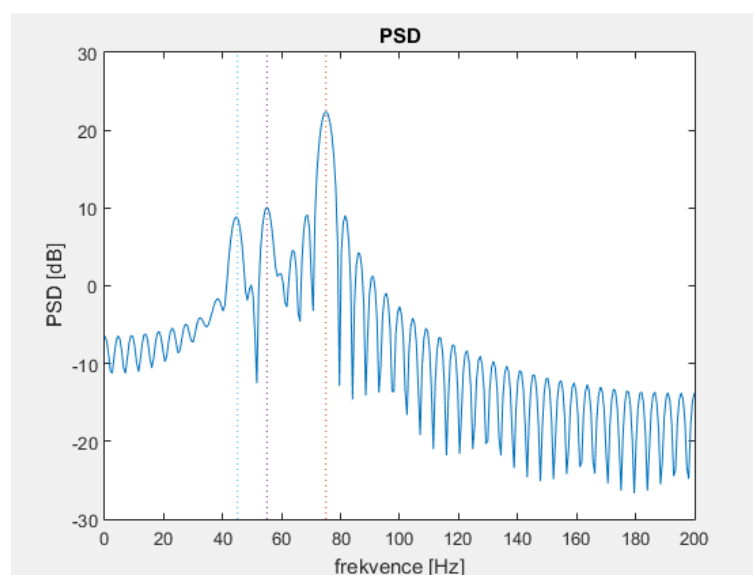
kde t je diskrétní čas. Definice časové osy je následující

$$t = 0: \frac{1}{f_s} : 10 \times T_1,$$

$f_s = 400$ Hz je vzorkovací frekvence a $T_1 = \frac{1}{f_1}$ je perioda harmonického signálu s nejnižší frekvencí. Jednotlivé frekvence jsou $f_1 = 45$ Hz, $f_2 = 55$ Hz, $f_3 = 75$ Hz.

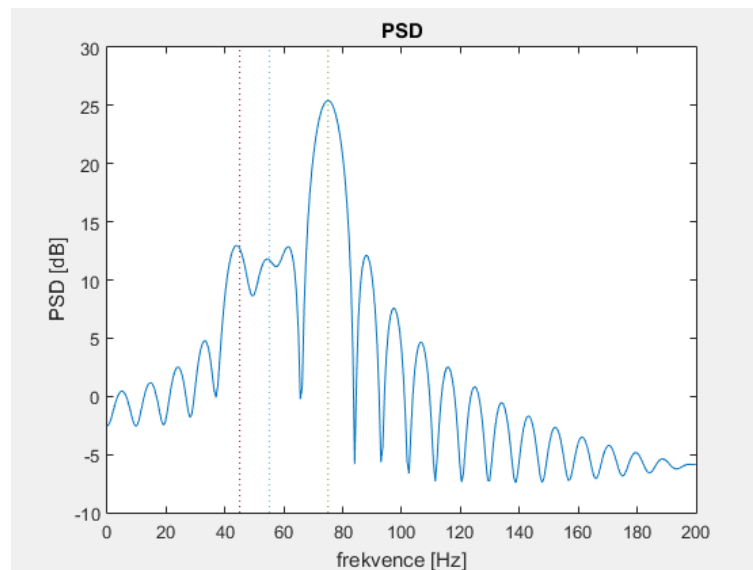
Na následujících obrázcích jsou uvedeny výsledky simulací. Pro lepší přehlednost jsou na všech obrázcích vyznačeny tečkovanou čarou frekvence, na kterých mají být pozorovány všechny frekvenční vrcholy, tzn. na frekvencích původních složek signálu.

Na obrázku (**Obr. 15**) je patrné, že odhad získaný **metodou periodogramu** vykazuje jen mírné zkreslení, ale především výrazný rozptyl. Pokud se vezme v úvahu odhad frekvenčních vrcholů (anglicky „peak“), ten je poměrně přesný.



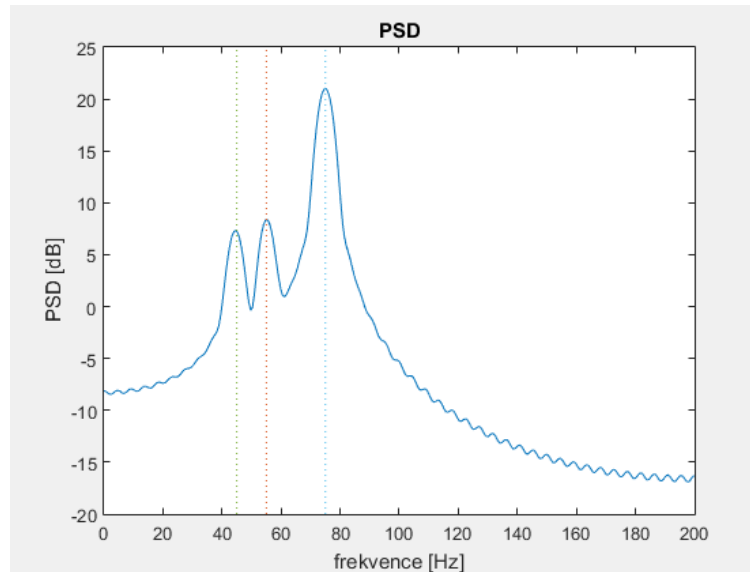
Obr. 15. Vyhodnocení metody periodogramu.

Metoda Bartlett (průměrného periodogramu), kde se zprůměrují čtyři bloky o velikosti 44 vzorků, je znázorněna na obrázku (**Obr. 16**). Zde je názorně vidět výsledné snížení rozptylu a zároveň ztráta spektrálního rozlišení, která vedla k tomu, že není pozorován frekvenční vrchol při 55 Hz. Průměrný periodogram lze chápat jako asymptoticky nezkreslený. V tomto případě by periodogram při dané vzorkovací frekvenci potřeboval větší počet vzorků, aby mohly být pozorovány všechny frekvenční vrcholy.



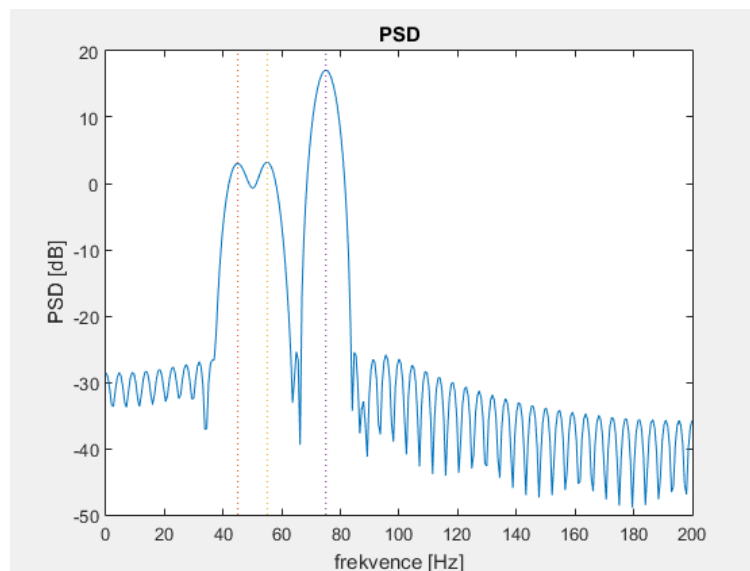
Obr. 16. Vyhodnocení metody Bartlett.

Jak je vidět na obrázku (**Obr. 17**), pomocí **metody Blackman-Tukey 2** lze dosáhnout kompromisu mezi zkreslením a rozptylem tak, že dojde ke zpřesnění odhadu PSD. Nicméně zlepšení kvality jde na úkor výpočetní složitosti. Protože autokorelační funkce musí být nejprve vynásobena oknem (zde se jedná o Hammingovo okno), dojde tak ke zvýšení počtu kroků násobení. Přesto se dá říct, že výsledek této metody je nad očekávání dobrý, protože jak rozptyl, tak zkreslení jsou velmi nízké.



Obr. 17. Vyhodnocení metody Blackman-Tukey 2.

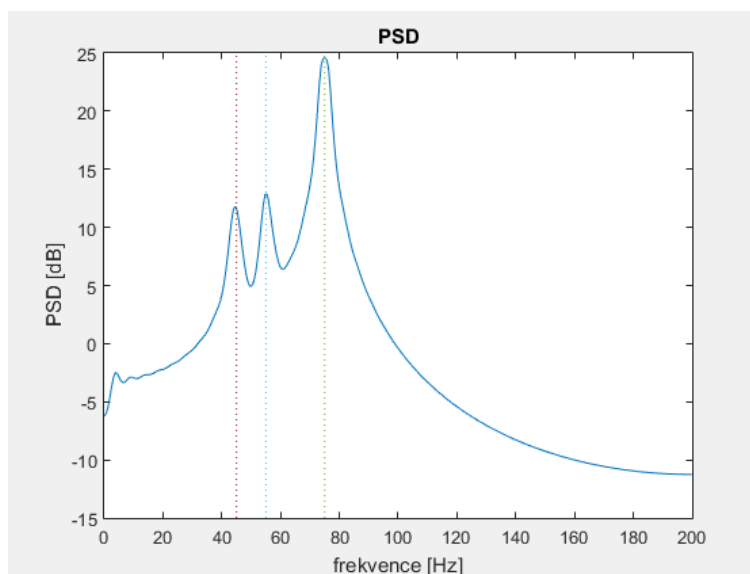
Na obrázku (**Obr. 18**) je zobrazen odhad PSD pomocí **metody Blackman-Tukey**. Průběh PSD je poměrně hladký. Je ale patrné, že došlo téměř ke splynutí frekvenčních vrcholů frekvencí 45 a 55 Hz. Možnost splynutí vrcholů frekvencí, které jsou blízko u sebe je slabou stránkou této metody. Na základě simulace lze tedy konstatovat, že tato metoda vyžaduje větší frekvenční rozlišení.



Obr. 18. Vyhodnocení metody Blackman-Tukey.

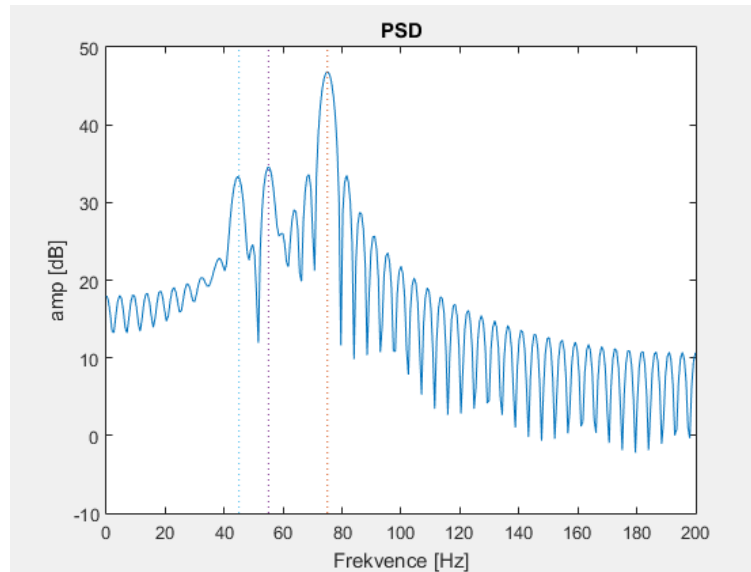
Na obrázku (**Obr. 19**) je odhad PSD získaný použitím **metody minimálního rozptylu**. Z obrázku je zřejmé, že křivka je hladká, čehož je dosaženo zvýšením výpočetní složitosti.

Zvýšení výpočetní složitosti je dáno tím, že je nutné spočítat vlastní čísla (eigenvalues) a vlastní vektory (eigenvectors) korelační matice.



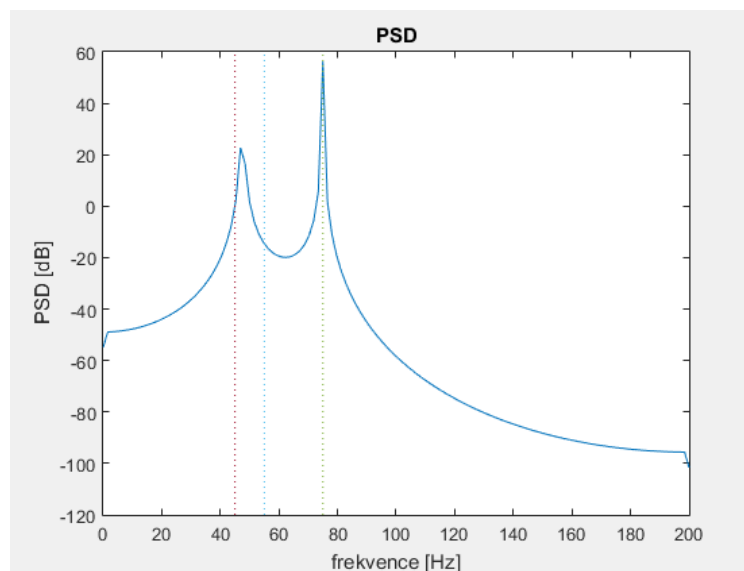
Obr. 19. Vyhodnocení metody minimálního rozptylu.

Fourierova řada a Fourierova transformace nám umožňují zobrazit deterministické časové funkce jako vážený součet nebo integrál sinusových funkcí. Proto je užitečné použít DFT k získání frekvenčního spektra deterministického signálu. Průběh amplitudového spektra jako výsledek DFT signálu je znázorněn na obrázku (Obr. 20). Jak je vidět na obrázku, sinusový signál vykazuje tři vrcholy při třech frekvencích. Při porovnání tohoto průběhu s průběhem získaným metodou periodogramu je možno pozorovat značnou podobnost těchto dvou průběhů. Tato podobnost je dána tím, že signál je deterministický. U deterministického signálu se periodogram signálu a jeho DFT liší velmi nepatrně, a to konkrétně v amplitudě frekvenčního vrcholu. U náhodného signálu by se průběh periodogramu a DFT lišil značně.



Obr. 20. Vyhodnocení metody DFT.

Na obrázku (Obr. 21) je znázorněn odhad výkonového spektra sinusového signálu pomocí **metody Burg**. Jak je vidět, ze tří frekvenčních vrcholů vykazuje sinusový signál pouze vrcholy dva a i ty jsou nepřesné. Je třeba poznamenat, že tato metoda je parametrická a vhodná pro signály, které lze modelovat jako AR model.

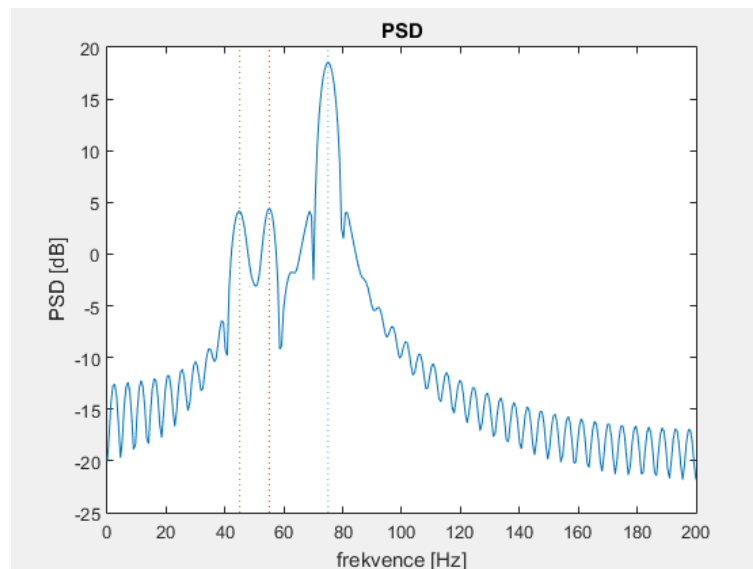


Obr. 21. Vyhodnocení metody Burg.

Použitím $(L - |\tau|)$ namísto L ve jmenovateli výpočtu autokorelační funkce se lze vyhnout zkreslení v autokorelaci periodogramu, čímž se získá nový autokorelační odhad

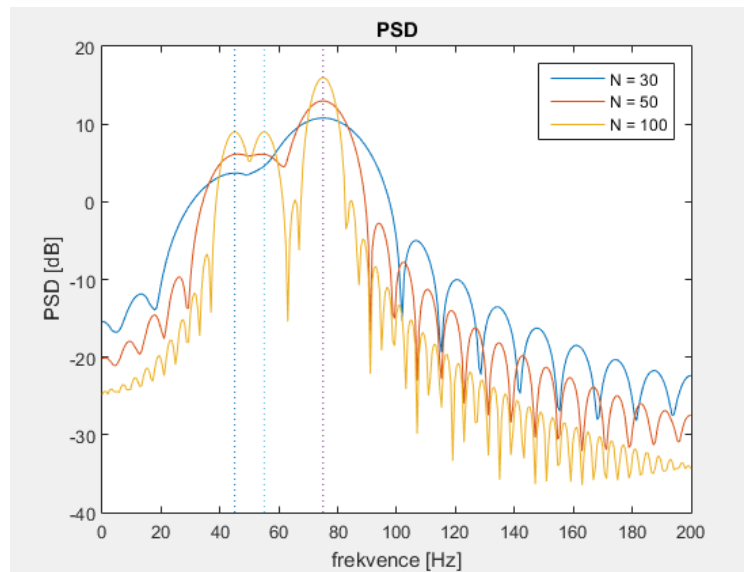
$$\hat{R}_{xx}(\tau) = \frac{1}{L - |\tau|} \sum_{n=0}^{L-1-|\tau|} x(n)x(n + |\tau|).$$

Tato změna může způsobit negativní hodnoty ve výsledné funkci PSD. To lze odstranit zavedením okénkové funkce $w(n)$ délky $(2K + 1)$ při $K < L$. Ponecháním $K \ll L$ odstraníme zašuměné vzorky autokorelační funkce při výpočtu PSD, čímž se zmenší rozptyl ve výsledném odhadu na úkor mírného nárůstu zkreslení, což plyne ze skutečnosti, že bylo použito méně dat. Tyto změny ve srovnání s metodou periodogramu způsobily, že jsou zřetelně vidět všechny tři frekvenční vrcholy. Odhad PSD sinusového signálu pomocí **metody nezkresleného odhadu** je vidět na obrázku (**Obr. 22**). Průběh grafu je poměrně hladký a zároveň je vidět, že i zkreslení je nízké.



Obr. 22. Vyhodnocení metody nezkresleného odhadu.

Pro vyhodnocení **metody Welch** byly vybrány tři různé hodnoty pro ty části rozsahů frekvencí, které při výpočtu zasahují do sousedních segmentů. Délku segmentu označme jako N ; Postupně bylo voleno $N = 30$, $N = 50$ a $N = 100$. Výsledky znázorněné na obrázku (**Obr. 23**) jasně ukazují kompromis mezi zkreslením a rozptylem. Jak rozptyl klesá, zkreslení se zvyšuje. Pokud se vezme v úvahu tvar křivky, tak zvýšení rozptylu znamená zúžení laloků okolo určité frekvence (protože v časové oblasti se graf autokorelační funkce rozšíří). Je třeba zdůraznit, že výběr délky segmentu N je poměrně náročný úkol, který vyžaduje praktické zkušenosti i dobrou znalost teorie.



Obr. 23. Vyhodnocení metody Welch.

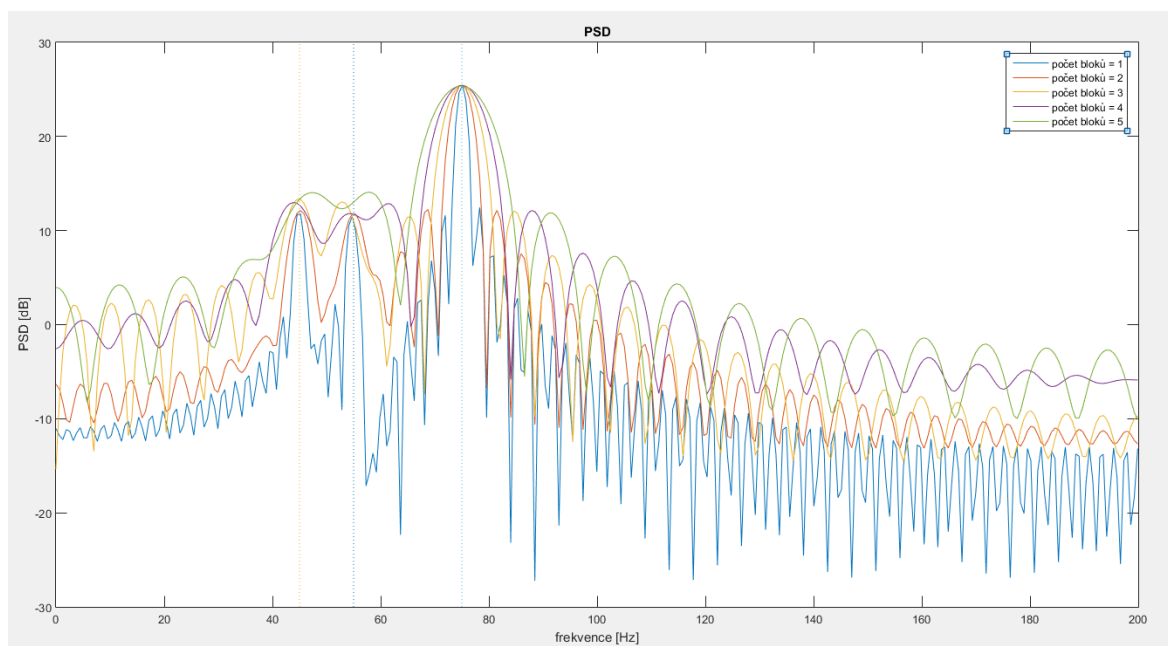
4.2 Vzájemné srovnání metod

Zde bude provedeno srovnání různých metod odhadu PSD. Srovnání je provedeno tak, že jedna metoda je zvolena jako tzv. referenční a vzhledem k ní jsou porovnány další metody podle různých kritérií. Zde je vybrána metoda periodogramu, ze které vychází celá řada dalších metod diskutovaných v této práci. Metoda periodogramu je nejznámější metodou pro výpočet výkonového spektra a je v literatuře často citovaná a je často aplikovaná. Důvodem je její poměrně malá výpočetní náročnost a ve většině případů dobrá přesnost. Jsou brána v úvahu následující kritéria: zkreslení, rozptyl a výpočetní složitost.

Jak již bylo řečeno, odhad PSD pomocí **metody periodogramu** má značné zkreslení, ale pokud je počet dostupných vzorků velmi vysoký, dojde k výraznému snížení tohoto zkreslení. To znamená, že frekvenční vrcholy se vyskytují přesně na frekvenci, na které jsou očekávány. Jeho výpočetní složitost s využitím časování pomocí MATLAB funkce tic-toc je 0,0211 sekund.

Další metodou je **metoda Bartlett** (průměrný periodogram). V této metodě se rozdělí dostupná data do bloků a poté se odhadne PSD pro každý blok zvlášť a následně se výsledky zprůměrují. Z obrázku (**Obr. 24**) níže je vidět, že čím větší je počet bloků, tím vyšší je nežádoucí zkreslení a rozptyl odhadu a tím nižší je přesnost celého odhadu. Takže pokud jsou data rozdělena do pěti bloků, odhad frekvenčního vrcholu je velmi zkreslený. Funkce tic-toc (0,1101 sekund) také ukazuje, že doba zpracování této metody je vyšší než

je tomu u metody periodogramu, zatímco kvalita odhadu není lepší než u metody periodogramu (měření času je zvoleno pro 4 bloky).



Obr. 24. Vyhodnocení metody Bartlett s více bloky.

Další metodou je **Blackman-Tukey**. Tato metoda nejprve vynásobí data oknem (v simulaci je použito Hammingovo okno) a poté pracuje s daty procházejícími oknem jako metoda periodogramu. Z obrázku (**Obr. 18**) je zřejmé, že má mírně vyhlazenější odhad než metoda periodogramu. Lze tedy říci, že pokud se bere v potaz rozptyl, dává tato metoda lepší výsledky než periodogram. Výpočetní složitost se mírně zvýší, protože průměrný výsledek funkce tic-toc je 0,0248 sekund.

Pokud se v metodě Blackman-Tukey vloží $(L - |\tau|)$ do jmenovatele vztahu pro výpočet odhadu, bude tato metoda nezkreslená, ale její rozptyl se zvýší. Tato metoda se nazývá **metodou nezkresleného odhadu**. Jak je vidět z obrázku (**Obr. 22**), tento graf je méně zkreslený a má hladší průběh než průběh získaný z odhadu pomocí metody periodogramu. Průměrný čas získaný z funkce tic-toc je podobný jako u metody Blackman-Tukey 0,0247 sekund.

Dalším způsobem jak odhadnout PSD, je **metoda Blackman-Tukey 2**. Tato metoda nejprve vypočítá autokorelační funkci a poté ji vynásobí oknem. Tato metoda ve srovnání se dvěma předchozími metodami, které nejprve vynásobily vzorky oknem a poté získaly autokorelační funkci, má velmi podobnou výpočetní složitost, kdy průměrný čas pro běh metody je 0,0245 sekund. Výsledek metody Blackman-Tukey 2 je však mnohem lepší než u obou předchozích metod, protože jak je zřejmé z obrázku (**Obr. 17**), má nízký rozptyl i zkreslení a frekvenční vrcholy jsou zřetelně vidět u odpovídající frekvence. Tato metoda také poskytuje lepší výsledek ve srovnání s metodou periodogramu.

Welchova metoda spočívá v rozdělení vzorků na segmenty a jejich vynásobení oknem o délce menší, než je celkový počet vzorků. Překrývání těchto segmentů je obvykle padesát procent, ale tato metoda vyžaduje hodně zkušeností, aby bylo možné najít správnou délku a překrytí dat, což je její slabou stránkou. Na obrázku (**Obr. 23**) je vidět, že pro správný odhad musí být počet vstupních vzorků mnohem větší, než počet vzorků potřebných pro metodu periodogramu. Konkrétně počet vzorků použitých v metodě periodogramu byl $L = 177$, ale v obrázku (**Obr. 23**) se počet vzorků zvýšil desetkrát ($N =$ délka segmentu). Konečně, s odpovídající délkou okna je možné dosáhnout požadovaného tvaru. Výpočetní složitost metody Welch je v průměru 0,0259 sekund.

Další metodou, která patří mezi neparametrické, je **metoda minimálního rozptylu**. Tato metoda, jak její název napovídá, má nejmenší rozptyl a její křivka má velmi hladký průběh (**Obr. 19**) ve srovnání s tvarem získaným metodou periodogramu a frekvenční vrcholy jsou odhadnuty naprosto správně. Cenou za toto zlepšení je výpočetní složitost. Průměrná doba potřebná pro běh této metody je 1,042 sekund, což je přibližně padesátkrát víc než u metody periodogramu.

Pokud je pro deterministický signál použita **metoda DFT (Obr. 20)**, její průběh je podobný jako u metody periodogramu. U metody DFT je brána v úvahu amplituda a u metody periodogramu čtverec amplitudy. Průměrná doba pro běh metody DFT je menší než u periodogramu a její hodnota je 0,02 sekund.

Poslední je **metoda Burg**, která je parametrická, což znamená, že jsou k dispozici informace o signálu, například řád AR modelu. Tato metoda byla použita k odhadu PSD náhodného signálu, ale jak je patrné z obrázku (**Obr. 21**), frekvenční vrcholy nejsou dobře definovány. Přitom v metodě periodogramu byly zřetelné. Výpočetní složitost metody Burg je 0,0022 sekund. Což je menší hodnota než u metody periodogramu, ale toto snížení také vedlo ke ztrátě kvality.

4.3 Srovnání výpočetní složitosti a přesnosti

Jak již bylo řečeno, výpočetní složitost byla vyhodnocena pomocí MATLAB funkce `tic toc`, která měří dobu provedení operací. Výsledné hodnoty byly získány průměrem z deseti měření za srovnatelných podmínek (použití stejného HW a SW, HW nebyl připojen k síti internet) pro každou metodu a jsou zobrazeny vzestupně v tabulce (Tab. 1) níže.

Tab. 1. Čas běhu metod pomocí funkce tic toc

Název metody	Funkce tic toc / sekunda
Burg	0,0022
DFT	0,0200
periodogram	0,0211
Blackman-Tukey 2	0,0245
nezkreslený odhad	0,0247
Blackman-Tukey	0,0248
Welch	0,0259
Bartlett	0,1101
minimální rozptyl	1,0420

Další možností jak porovnat metody mezi sebou, je využít kritérium přesnosti. Protože všechny metody použitelné pro zjištění PSD jsou pouhým odhadem, je nutné zvolit jednu metodu jako referenční a s touto metodou pak zbylé porovnávat. Na základě teoretických předpokladů a vyhodnocení simulací výše byla jako nejvhodnější metoda vybrána metoda minimálního rozptylu. Přesnost je pak statisticky vyhodnocena jako odchylka od PSD pro

jednotlivé frekvence (suma čtverců odchylek). Hodnoty jsou uvedeny v tabulce (Tab. 2) níže.

Tab. 2. Suma čtverců odchylek.

Název metody	Suma čtverců odchylek / tisíce
DFT	54,90
periodogram	82,57
Blackman-Tukey 2	122,69
nezkreslený odhad	243,17
Blackman-Tukey	281,44
Bartlett	296,03
Welch	314,58
Burg	429,51

4.4 Vyhodnocení metod - závěr

V této části bylo zkoumáno devět různých algoritmů pro odhad PSD. Byly získány výsledky simulací. Výsledky ukazují, že s ohledem na kritéria zkreslení a rozptylu by nejlepší metodou byla metoda minimálního rozptylu, ale tato metoda má nejvyšší výpočetní složitost. To znamená, že pokud nemáme omezení, pokud jde o výpočetní složitost, nejlepším způsobem odhadu PSD je metoda minimálního rozptylu. Na druhou stranu metoda Burg má zdaleka nejnižší výpočetní složitost, ale zaostává v kritériu přesnosti. Velmi nízké zkreslení a rozptyl vykazuje také metoda Blackman-Tukey 2 a na rozdíl od metody minimálního rozptylu má výrazně nižší výpočetní složitost. Takže představuje vhodnou alternativu s ohledem na zmiňovaná kritéria.

Je nutné poznamenat, že i jiné metody mají své výhody. Například metoda periodogramu, i když není tak přesná jako metoda minimálního rozptylu, pokud jde o zkreslení a rozptyl, tak poskytuje přijatelnou přesnost s nízkou výpočetní složitostí. Pokud je počet vstupních vzorků příliš velký, lze použít metodu Bartlett (průměrný periodogram).

Metody, které používají okénkování pro data nebo pro autokorelační funkci (Blackman-Tukey a Blackman-Tukey 2), poskytují dost odlišné výsledky. Metoda Blackman-Tukey nedává tak dobrý výsledek jak metoda periodogramu ve smyslu zkreslení, má přibližně stejnou výpočetní složitost a horší přesnost (suma čtverců odchylek). To znamená, že nemá přesnost metody minimálního rozptylu. Na druhou stranu např. metoda Blackman-Tukey 2 příliš nezaostává v přesnosti za metodou periodogramu, nárůst výpočetní složitost není velký a její výsledek s ohledem na kritéria rozptylu a zkreslení je srovnatelný s metodou minimálního rozptylu.

Obecně lze říct, že z metod založených na periodogramu (využívajících autokorelační funkci pro odhad PSD), mezi které patří právě metody periodogramu, Bartlett, nezkresleného odhadu, Blackman-Tukey a Blackman-Tukey 2, dává nejlepší výsledek metoda Blackman-Tukey 2.

Je třeba poznamenat, že u velké části metod existuje kompromis mezi přesností a výpočetní složitostí.

Pokud se bere v úvahu frekvenční rozlišení, tak se studované metody příliš neliší, a pokud se má zvolit vhodná metoda, je důležitější zvážit další kritéria. Jedině u metod Bartlett a Blackman-Tukey se dá říct, že pro získání odpovídajících výsledků potřebují vyšší frekvenční rozlišení.

Výsledky však není možné zobecnit tak, aby bylo možné vybrat jednu metodu, která by byla vhodná pro všechny možné situace.

5 NÁVRH VZOROVÉHO PROTOKOLU

5.1 Zadání

1. Definujte diskrétní signál s třemi harmonickými složkami. Frekvence složek budou $f_1 = 45$ Hz, $f_2 = \left(\frac{a_1}{2} + 55\right)$ Hz, $f_3 = (a_2 + 75)$ Hz. Koeficient a_1 volte jako měsíc narození a koeficient a_2 jako den narození. Výsledné frekvence zaokrouhlete na celá čísla. Tedy například datum 7. října znamená, že jednotlivé frekvence budou ve tvaru:

$$f_1 = 45 \text{ Hz}, f_2 = \left(\frac{10}{2} + 55\right) \text{ Hz}, f_3 = (7 + 75) \text{ Hz},$$

$$f_1 = 45 \text{ Hz}, f_2 = 60 \text{ Hz}, f_3 = 82 \text{ Hz}.$$

2. Vzorkovací frekvenci zvolte jako čtyřnásobek nejvyšší ze tří zadaných frekvencí a zaokrouhlete na desítky. Tedy pro výše zmíněný příklad bude vzorkovací frekvence následující:

$$f_s = 4 * 82 = 328 \doteq 330 \text{ Hz}.$$

3. V prostředí MATLAB vytvořte programy pro výpočet výkonového spektra simulovaného diskrétního signálu dle algoritmů metody periodogramu a metody Blackman-Tukey (vyučující může zadat jiné metody).
4. Zobrazte výkonová spektra získaná jednotlivými metodami.
5. Výsledky získané jednotlivými metodami porovnejte navzájem. Berte v úvahu následující kritéria: zkreslení, rozptyl a výpočetní složitost.

5.2 Metoda periodogramu

Periodogram se získá jako odhad výkonové spektrální hustoty (PSD) pomocí autokorelační funkce (použití Fourierovy transformace na autokorelační funkci). Výsledkem je následující odhad:

$$\hat{S}_{xx}(\omega) = \sum_{\tau=-(L-1)}^{L-1} \hat{R}_{xx}(\tau) e^{-j\omega\tau}, \quad (23)$$

kde odhad autokorelační funkce je

$$\hat{R}_{xx}(\tau) = \frac{1}{L} \sum_{n=0}^{L-1-|\tau|} x(n)x(n+|\tau|). \quad (24)$$

L jsou pozorování diskrétního procesu (počet vzorků), τ představuje zpoždění.

5.3 Metoda Blackman-Tukey

Odhad PSD pomocí metody Blackman-Tukey se získá podobně jako u metody periodogramu. Nejprve se však na vzorky signálu aplikuje Hammingovo okno, až poté se získá autokorelační funkce a na ni aplikuje Fourierova transformace. Signál v okně je definován:

$$v(n) = w(n)y(n),$$

pak odhad lze vyjádřit:

$$\hat{S}_{vv}(\omega) = \sum_{\tau=-(L-1)}^{L-1} \hat{R}_{vv}(\tau) e^{-j\omega\tau}, \quad (25)$$

kde odhad autokorelační funkce je

$$\hat{R}_{vv}(\tau) = \frac{1}{L} \sum_{n=0}^{L-|\tau|-1} w(n+\tau)w(n)x(n+\tau)x(n). \quad (26)$$

Pro Hammingovo okno platí:

$$w[n] = 0,54 - 0,46 \cos \frac{2\pi n}{N-1} \quad n \in \langle 0, N-1 \rangle. \quad (27)$$

6 VYPRACOVÁNÍ VZOROVÉHO PROTOKOLU

6.1 Metoda periodogramu

6.1.1 Definice signálu

Simulovaný signál je ve tvaru:

$$x(t) = \sin(2\pi t f_1) + \sin(2\pi t f_2) + 5\sin(2\pi t f_3),$$

kde t je diskrétní čas.

Kód v programu Matlab pro definici signálu:

```
signal1 = sin(2*pi*f1*t); % definice první složky simulovaného signálu  
signal2 = sin(2*pi*f2*t); % definice druhé složky simulovaného signálu  
signal3 = 5*sin(2*pi*f3*t); % definice třetí složky simulovaného signálu
```

Výsledný simulovaný signál je součtem tří harmonických složek:

```
x = signal1 + signal2 + signal3; % výsledný simulovaný signál
```

Definice časové osy je následující:

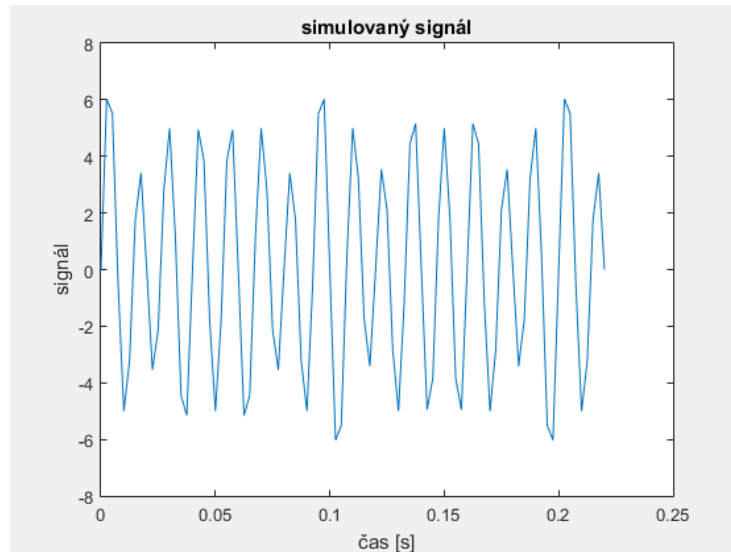
$$t = 0:\frac{1}{f_s}:10 \times T_1.$$

Frekvence složek jsou: $f_1 = 45$ Hz, $f_2 = 55$ Hz, $f_3 = 75$ Hz.

$f_s = 400$ Hz je vzorkovací frekvence a $T_1 = \frac{1}{f_1}$ je perioda harmonického signálu s nejnižší frekvencí.

Kód pro vykreslení simulovaného signálu:

```
figure(1);  
plot(t,x)  
title('signal'), xlabel('čas [s]'), ylabel('signál')
```



Obr. 25. Průběh simulovaného signálu.

6.1.2 Algoritmus výpočtu

Nejprve se pomocí rovnice (24) vypočítá autokorelační funkce, a protože autokorelační funkce je sudá – symetrie kolem vertikální osy, pokud je vypočtena jedna strana, je druhá získána na základě symetrie. Z tohoto důvodu je vypočtena v MATLAB kódu pouze pravá strana autokorelační funkce a následně je získána levá strana pomocí symetrie.

Je tedy potřeba smyčka `for`, která posune vzorek $x(n + |\tau|)$ pro $\tau = 1:L$ a poté vypočítá autokorelační funkci pro tuto hodnotu τ (ve skutečnosti by τ měla být $0:L - 1$, ale v programu MATLAB začínají proměnné od 1). To se provede pomocí následujícího kódu:

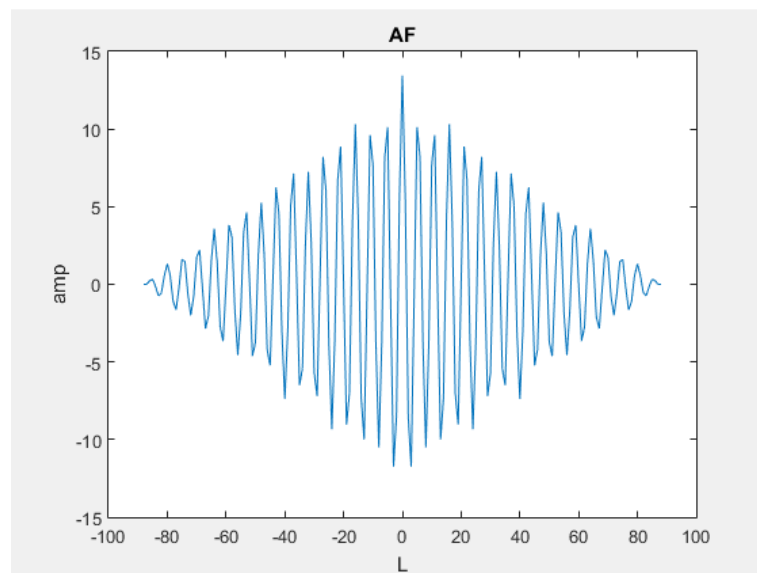
```
for i = 1:L % výpočet pravé strany AF
    % posun signálu doleva (složení signálu z původního signálu a nul)
    xtau = [x(i:end) zeros(1,i-1)];
    % součin původního a posunutého (zpožděného) signálu
    rx = xtau .* x;
    s = 0;
    for j = 1:L % suma rx
        s = s + rx(j);
    end
    Rx(i) = s;
end
```

Nyní je třeba zrcadlit pravou stranu autokorelační funkce, aby se získala levá strana funkce a pak se spojí obě strany a vydělí pomocí L . To se děje pomocí následujících příkazů:

```
for i = 1:L-1
    Rx1(i) = Rx(L-i+1); % získání levé strany AF ze strany pravé
end
Rx = [Rx1 Rx] ./L; % vytvoření AF z pravé a levé strany a dělení pomocí L
```

Kód pro vykreslení autokorelační funkce:

```
% vykreslení autokorelační funkce
figure(2);
plot(tau,Rx)
title('AF'), xlabel('L'), ylabel('amp')
```



Obr. 26. Průběh autokorelační funkce.

Byla získána autokorelační funkce a nyní je možné se dostat k periodogramu pomocí Fourierovy transformace (23). K získání Fourierovy transformace lze použít vestavěnou MATLAB funkci `fft`, ale místo toho se využije následující kód, který dělá totéž.

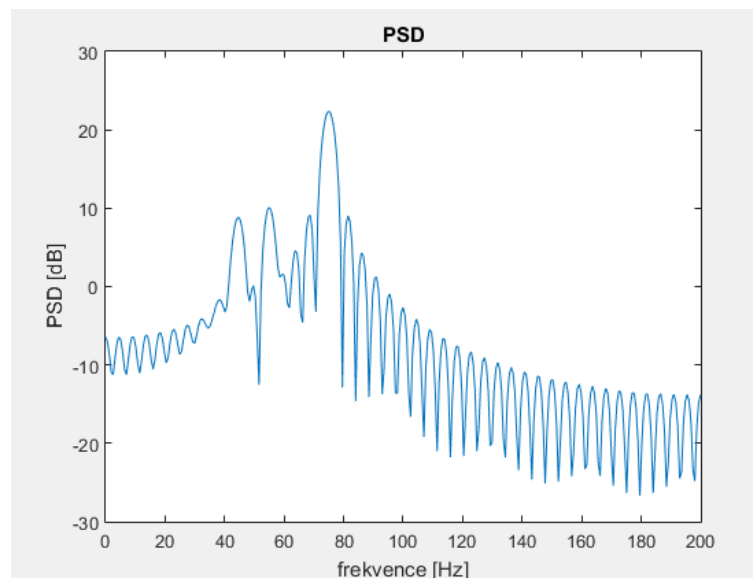
```
for i = 1:length(w) % smyčka pro frekvenci
    psd = 0;
    for j = 1:2*L-1 % smyčka pro sumu
        psd = psd + Rx(j) * exp(-1i*w(i)*nu(j));
    end
end
```

```
PSD(i) = abs(psd)/pi;  
End
```

Tento kód v rovnici (23), pro každé ω , ($\omega = 0:0.01:\pi$), vynásobí autokorelační funkci $e^{-j\omega v}$ a sečte výsledné vzorky. Vnější `for` smyčka se používá pro výpočet hodnot $\hat{S}_{xx}(\omega)$ v daném ω a vnitřní `for` smyčka pro součet vzorků. Zde je nutno připomenout, že pomocí tohoto kódu se získá pravá strana Fourierovy transformace ($\omega = 0:0.01:\pi$), protože Fourierova transformace je také symetrická kolem vertikální osy, takže získání jedné strany je dostatečné pro zjištění všech frekvenčních složek.

Kód pro vykreslení PSD:

```
% vykreslení PSD  
figure(3);  
plot(w,10*log10(PSD))  
title('PSD'), xlabel('frekvence [Hz]'), ylabel('PSD [dB]')
```



Obr. 27. Průběh PSD periodogramu.

6.2 Metoda Blackman-Tukey

6.2.1 Definice signálu

Simulovaný signál je ve tvaru:

$$x(t) = \sin(2\pi t f_1) + \sin(2\pi t f_2) + 5\sin(2\pi t f_3),$$

kde t je diskrétní čas.

Kód v programu Matlab pro definici signálu:

```
signal1 = sin(2*pi*f1*t); % definice první složky simulovaného signálu  
signal2 = sin(2*pi*f2*t); % definice druhé složky simulovaného signálu  
signal3 = 5*sin(2*pi*f3*t); % definice třetí složky simulovaného signálu
```

Výsledný simulovaný signál je součtem tří harmonických složek:

```
x = signal1 + signal2 + signal3; % výsledný simulovaný signál
```

Definice časové osy je následující:

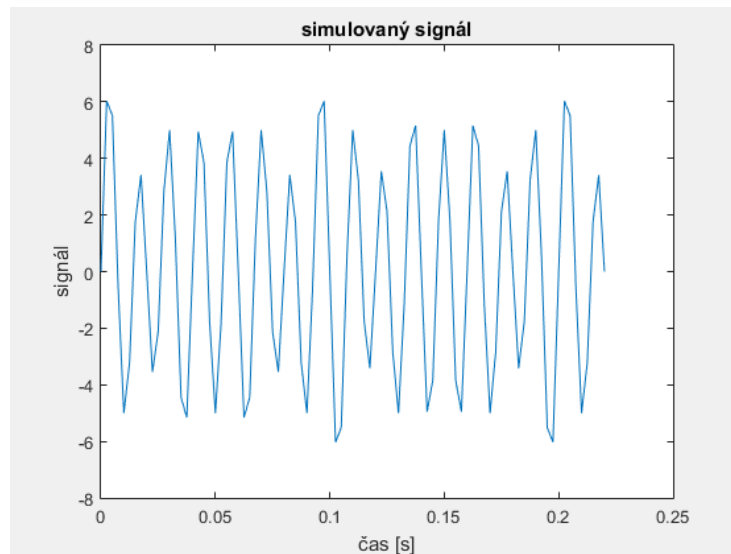
$$t = 0:\frac{1}{f_s}:10 \times T_1.$$

Frekvence složek jsou: $f_1 = 45$ Hz, $f_2 = 55$ Hz, $f_3 = 75$ Hz.

$f_s = 400$ Hz je vzorkovací frekvence a $T_1 = \frac{1}{f_1}$ je perioda harmonického signálu s nejnižší frekvencí.

Kód pro vykreslení simulovaného signálu:

```
figure(1);  
plot(t,x)  
title('signal'), xlabel('čas [s]'), ylabel('signál')
```



Obr. 28. Průběh simulovaného signálu.

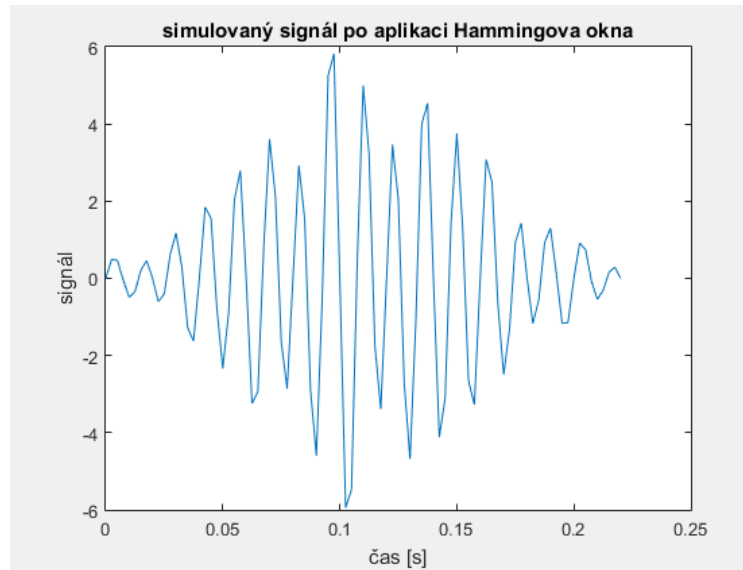
6.2.2 Algoritmus výpočtu

Ve výše zmíněných vztazích bylo zjištěno, že na vzorky signálu je možno použít okno a poté aplikovat na periodogram pro odhad spektrální hustoty. Takže v simulaci provedené v tomto algoritmu se nejprve vybere Hammingovo okno a je jím vynásoben signál. Vytvoření okna a jeho použití na signál se provádí pomocí následujících příkazů:

```
win = 0.54 - 0.46*cos(2*pi*(0:L-1)/L); % Definice Hammingova okna
% aplikace Hammingova okna na simulovaný signál - okénkování
x = x .* win;
```

Kód pro vykreslení simulovaného signálu po aplikaci Hammingova okna:

```
% vykreslení simulovaného signálu po aplikaci Hammingova okna
figure(2);
plot(t,x)
title('simulovaný signál po aplikaci Hammingova okna'), xlabel('čas
[s]'), ylabel('signál')
```



Obr. 29. Průběh simulovaného signálu po aplikaci Hammingova okna.

Nyní se pomocí rovnice (24) vypočítá autokorelační funkce, a protože autokorelační funkce má vlastnost symetrie kolem nulového bodu, pokud je vypočtena jedna strana, je druhá získána symetricky. Z tohoto důvodu je vypočtena v MATLAB kódu pouze pravá strana autokorelační funkce a následně je získána levá strana pomocí symetrie.

Je tedy potřeba smyčka `for`, která posune vzorek $x(n + |\tau|)$ pro $\tau = 1:L$ a poté vypočítá autokorelační funkci pro tuto hodnotu τ . To se provede pomocí následujícího kódu:

```
for i = 1:L % výpočet pravé strany AF
    % posun signálu doleva (složení signálu z původního signálu a nul)
    xtau = [x(i:end) zeros(1,i-1)];
    % součin původního a posunutého (zpožděného) signálu
    rx = xtau .* x;
    s = 0;
    for j = 1:L % suma rx
        s = s + rx(j);
    end
    Rx(i) = s;
end
```

Nyní je třeba zrcadlit pravou stranu autokorelační funkce, aby se získala levá strana funkce a pak se spojí obě strany a vydělí pomocí L . To se děje pomocí následujících příkazů:

```
for i = 1:L-1
```

```

    Rx1(i) = Rx(L-i+1); % získání levé strany AF ze strany pravé
end
Rx = [Rx1 Rx] ./L; % vytvoření AF z pravé a levé strany a dělení pomocí L

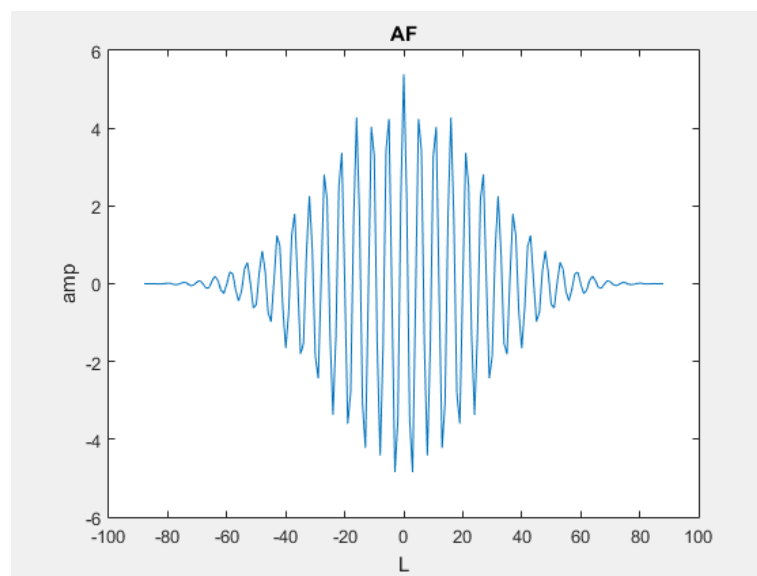
```

Kód pro vykreslení autokorelační funkce:

```

% vykreslení autokorelační funkce
figure(2);
plot(tau,Rx)
title('AF'), xlabel('L'), ylabel('amp')

```



Obr. 30. Průběh autokorelační funkce.

Byla získána autokorelační funkce a nyní je možné vypočítat PSD pomocí Fourierovy transformace (23). K získání Fourierovy transformace lze použít vestavěnou MATLAB funkci `fft`, ale místo toho se využije následující kód, který dělá totéž.

```

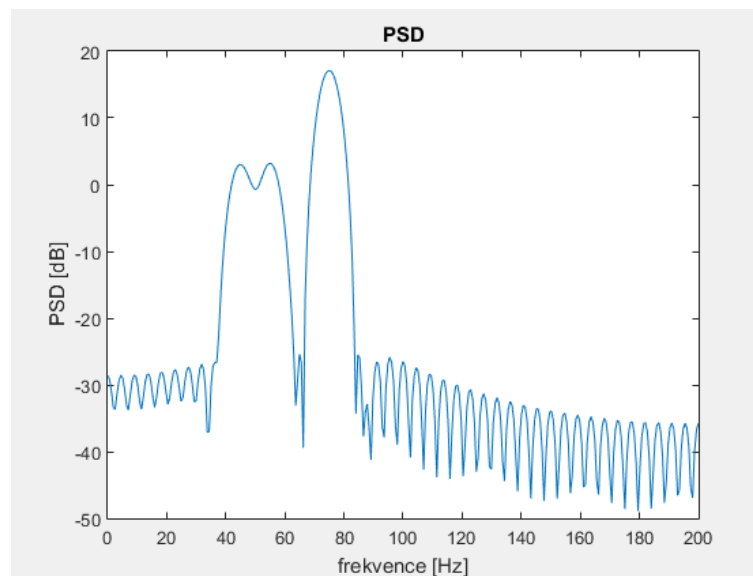
for i = 1:length(w) % smyčka pro frekvenci
    psd = 0;
    for j = 1:2*L-1 % smyčka pro sumu
        psd = psd + Rx(j) * exp(-1i*w(i)*nu(j));
    end
    PSD(i) = abs(psd)/pi;
End

```


Tento kód v rovnici (23), pro každé ω , ($\omega = 0:0.01:\pi$), vynásobí autokorelační funkci $e^{-j\omega v}$ a sečte výsledné vzorky. Vnější `for` smyčka se používá pro výpočet hodnot $\hat{S}_{xx}(\omega)$ v daném ω a vnitřní `for` smyčka pro součet vzorků. Zde je nutno připomenout, že pomocí tohoto kódu se získá Fourierova transformace pro kladné hodnoty frekvencí ($\omega = 0:0.01:\pi$), protože Fourierova transformace je také symetrická kolem vertikální osy, takže získání jedné strany je dostatečné pro zjištění všech frekvenčních složek.

Kód pro vykreslení PSD:

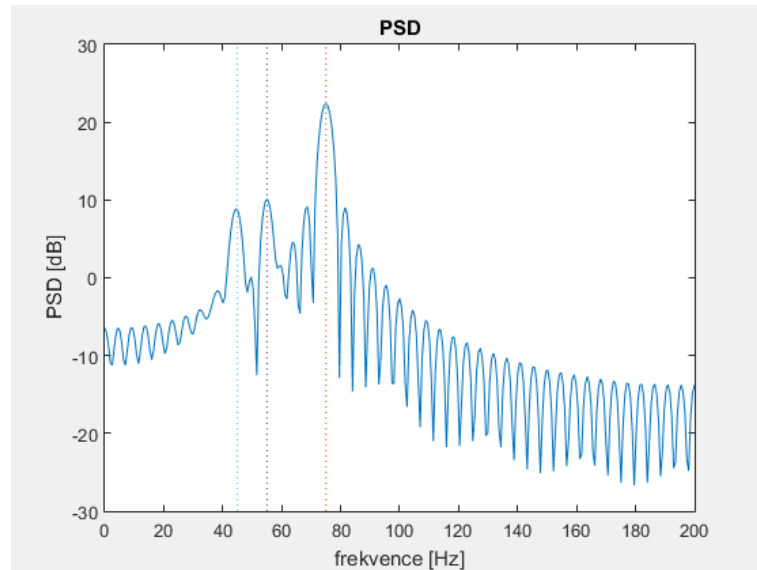
```
% vykreslení PSD
figure(3);
plot(w,10*log10(PSD))
title('PSD'), xlabel('frekvence [Hz]'), ylabel('PSD [dB]')
```



Obr. 31. Průběh PSD Blackman-Tukey.

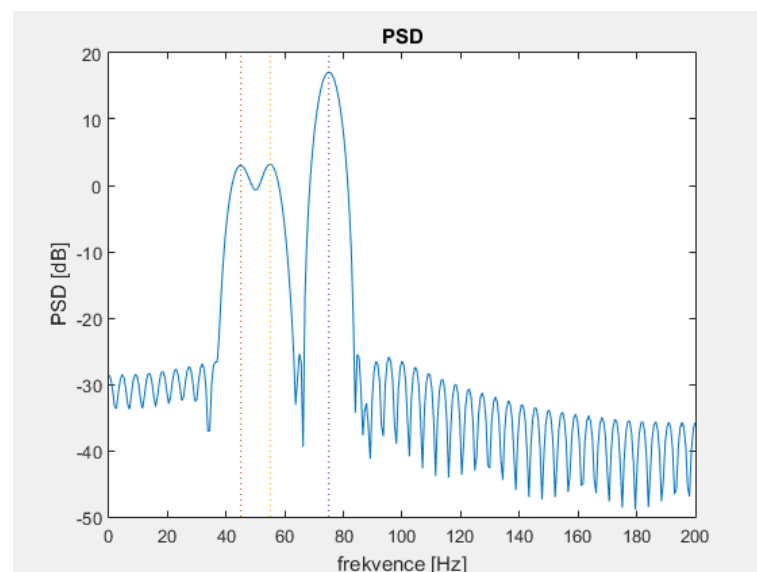
6.3 Vyhodnocení metod periodogramu a Blackman-Tukey

Zde budou vyhodnoceny obě výše popsané metody sloužící k odhadu PSD. Jsou brána v úvahu následující kritéria: zkreslení, rozptyl a výpočetní složitost.



Obr. 32. Vyhodnocení metody periodogramu.

Jak již bylo řečeno, odhad PSD pomocí metody periodogramu může mít značné zkreslení, ale pokud je počet dostupných vzorků velmi vysoký, dojde k výraznému snížení tohoto zkreslení. To znamená, že frekvenční vrcholy se vyskytují přesně na frekvenci, na které jsou očekávány. Na obrázku (Obr. 32) je patrné, že odhad získaný touto metodou vykazuje jen mírné zkreslení, ale především výrazný rozptyl. Jeho výpočetní složitost s využitím časování pomocí MATLAB funkce tic-toc je 0,0211 sekund.



Obr. 33. Vyhodnocení metody Blackman-Tukey.

Na obrázku (Obr. 33) je zobrazen odhad PSD pomocí metody Blackman-Tukey. Tato metoda nejprve vynásobí data oknem (v simulaci je použito Hammingovo okno) a poté pracuje s daty procházejícími oknem jako metoda periodogramu. Z obrázku (Obr. 33) je zřejmé, že průběh PSD je poměrně hladký. Lze tedy říci, že pokud se bere v potaz rozptyl, dává tato metoda mírně lepší výsledky než periodogram. Je ale patrné, že došlo téměř ke splynutí frekvenčních vrcholů frekvencí 45 a 55 Hz. Možnost splynutí vrcholů frekvencí, které jsou blízko u sebe je slabou stránkou této metody. Na základě simulace lze tedy konstatovat, že tato metoda vyžaduje větší frekvenční rozlišení. Výpočetní složitost se mírně zvýší, protože průměrný výsledek funkce tic-toc je 0,0248 sekund.

6.4 Závěr

Pro obě metody byly vypracovány algoritmy v prostředí MATLAB, provedeny simulace a získány odhady PSD. Odhad získaný metodou periodogramu vykazuje o něco lepší výsledky s ohledem na kritéria zkreslení nebo výpočetní složitosti. Naopak odhad pomocí metody Blackman-Tukey se jeví lépe, pokud se bere v úvahu rozptyl.

6.5 Zdrojový kód

Kód programu sa nachází v příloze č. I a III.

ZÁVĚR

Diplomová práce se zabývá problematikou algoritmů a metod vhodných pro výpočet odhadu výkonových spektrálních hustot náhodných signálů. Odhady výkonových spekter jsou důležitou částí spektrální analýzy náhodných signálů, lze pomocí nich určit dominantní složky signálu.

Součástí teoretické části je rešerše metod, které se používají k odhadu výkonových spekter. Byly zpracovány jak metody parametrické, tak neparametrické. Byly představeny neparametrické metody založené na periodogramu a to konkrétně metoda periodogramu, metoda Bartlett, metoda Blackman-Tukey, metoda nezkresleného odhadu a metoda Blackman-Tukey 2. Následovala metoda Welch, metoda minimálního rozptylu a metoda DFT. Z parametrických metod byla zpracována metoda Burg.

V praktické části byly získané teoretické poznatky zpracovány do algoritmů, které byly realizovány v prostředí MATLAB. Implementované algoritmy byly simulačně testovány na signálu tvořeném třemi harmonickými složkami. Dvě frekvence byly voleny v malé vzájemné vzdálenosti na frekvenční ose, aby bylo možno vyhodnotit zkreslení výsledného výkonového spektra. Byly získány výsledné průběhy PSD a všechny metody byly vyhodnoceny z hlediska výpočetní náročnosti, přesnosti, rozlišení spektra, zkreslení a rozptylu. Nakonec byla navržena vhodná úloha pro předmět Identifikace a modelování náhodných signálů a byl vypracován vzorový protokol. S ohledem na náročnost algoritmu a význam metod byla zvolena metoda periodogramu a jako doplňková metoda Blackman-Tukey.

Neexistuje metoda, která by předčila všechny ostatní metody ve všech sledovaných parametrech. Například metoda Burg byla nejlepší z hlediska výpočetní náročnosti, ale výrazně zaostala v přesnosti. Metoda minimálního rozptylu, jak už její název napovídá, se představila jako dobrý odhadce PSD, vykazovala malý rozptyl, zkreslení a byla tedy vybrána jako referenční metoda pro kritérium přesnosti. Na druhou stranu vykazovala největší výpočetní složitost. Solidní výsledek měla metoda periodogramu v oblastech přesnosti, výpočetní složitosti, méně však již v rozptylu. Celkově se dá říct, že z metod založených na periodogramu, se nejlépe jevila metoda Blackman-Tukey 2, která příliš nezaostala za metodou minimálního rozptylu v parametru rozptylu, zkreslení a přesnosti a zároveň měla solidní výpočetní složitost. Metoda DFT vykazovala velmi podobné výsledky jako metoda periodogramu. Poměrně obtížnou na hodnocení se ukázala metoda Welch, kde velmi záleží na výběru délky segmentu N , což není snadný úkol, který vyžaduje značné zkušenosti.

Závěrem se dá říct, že výsledky není možné zobecnit tak, aby bylo možné vybrat jednu metodu, která by byla vhodná pro všechny možné situace.

SEZNAM POUŽITÉ LITERATURY

- [1] DINIZ, Paulo Sergio Ramirez, Eduardo A. B. DA SILVA a Sergio L. NETTO. Digital signal processing: system analysis and design. Second edition. New York: Cambridge University Press, 2010, xxi, 889 s. ISBN 9780521887755.
- [2] LEON-GARCIA, Alberto a Alberto LEON-GARCIA. Probability, statistics, and random processes for electrical engineering. 3rd ed. Upper Saddle River, NJ: Pearson/Prentice Hall, c2008. ISBN 0131471228.
- [3] MANOLAKIS, Dimitris G. a Vinay K. INGLE. Applied digital signal processing: theory and practice. New York: Cambridge University Press, 2011. ISBN 0521110025.
- [4] OPPENHEIM, Alan V., Ronald W. SCHAFER a John R. BUCK. Discrete-time signal processing. 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1999. ISBN 0-13-754920-2.
- [5] OPPENHEIM, Alan V., Alan S. WILLSKY a Syed Hamid NAWAB. Signals & systems. 2nd ed. Upper Saddle River: Prentice Hall, c1997, xxx, 957 s. Prentice-Hall signal processing series. ISBN 0138147574.
- [6] ZAPLATÍLEK, Karel a Bohuslav DOŇAR. MATLAB: začínáme se signály. Praha: BEN - technická literatura, 2006. 271s., ISBN 80-7300-200-0.
- [7] LI, Tan. Digital Signal Processing, Fundamentals and Applications. Elsevier, 2008, 816 s. ISBN 978-0-12-374090-8.
- [8] CANDY, James. Model Based Signal Processing. John Wiley & Sons, 2006, 677 s., ISBN 978-0-471-23632-0.
- [9] KUBALČÍK, Marek. *Modelování a identifikace náhodných procesů* [online]. Zlín, 2017 [cit. 2020-05-16]. Dostupné z: <http://vyuka.fai.utb.cz/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

PSD	Power Spectral Density.
FT	Fourier transform.
DFT	Discrete Fourier transform.
FFT	Fast Fourier transform.
AF	Autokorelační funkce.
\hat{R}	Odhad autokorelační funkce.
\hat{S}	Odhad PSD.
\sum	Suma.
\int	Integrál.
lim	Limita.
\in	Prvek množiny.
\forall	Pro každé.
\approx	Přibližně rovno.
\neq	Nerovná se.

SEZNAM OBRÁZKŮ

<i>Obr. 1. Amplituda FT.</i>	13
<i>Obr. 2. Fáze FT.</i>	14
<i>Obr. 3. Výkonová spektrální hustota.</i>	17
<i>Obr. 4. Zobrazení oken v časové oblasti.</i>	20
<i>Obr. 5. Zobrazení oken ve frekvenční oblasti.</i>	20
<i>Obr. 6. Průběh PSD s využitím metody periodogramu.</i>	33
<i>Obr. 7. Průběh PSD s využitím metody Bartlett.</i>	34
<i>Obr. 8. Průběh PSD s využitím metody Blackman-Tukey.</i>	35
<i>Obr. 9. Průběh PSD s využitím metody nezkresleného odhadu.</i>	36
<i>Obr. 10. Průběh PSD s využitím metody Blackman-Tukey 2.</i>	36
<i>Obr. 11. Průběh PSD s využitím metody Welch.</i>	38
<i>Obr. 12. Průběh PSD s využitím metody minimálního rozptylu.</i>	40
<i>Obr. 13. Průběh PSD s využitím metody DFT.</i>	41
<i>Obr. 14. Průběh PSD s využitím metody Burg.</i>	42
<i>Obr. 15. Vyhodnocení metody periodogramu.</i>	43
<i>Obr. 16. Vyhodnocení metody Bartlett.</i>	44
<i>Obr. 17. Vyhodnocení metody Blackman-Tukey 2.</i>	45
<i>Obr. 18. Vyhodnocení metody Blackman-Tukey.</i>	45
<i>Obr. 19. Vyhodnocení metody minimálního rozptylu.</i>	46
<i>Obr. 20. Vyhodnocení metody DFT.</i>	47
<i>Obr. 21. Vyhodnocení metody Burg.</i>	47
<i>Obr. 22. Vyhodnocení metody nezkresleného odhadu.</i>	48
<i>Obr. 23. Vyhodnocení metody Welch.</i>	49
<i>Obr. 24. Vyhodnocení metody Bartlett s více bloky.</i>	50
<i>Obr. 25. Průběh simulovaného signálu.</i>	58
<i>Obr. 26. Průběh autokorelační funkce.</i>	59
<i>Obr. 27. Průběh PSD periodogramu.</i>	60
<i>Obr. 28. Průběh simulovaného signálu.</i>	62
<i>Obr. 29. Průběh simulovaného signálu po aplikaci Hammingova okna.</i>	63
<i>Obr. 30. Průběh autokorelační funkce.</i>	64
<i>Obr. 31. Průběh PSD Blackman-Tukey.</i>	65
<i>Obr. 32. Vyhodnocení metody periodogramu.</i>	66

Obr. 33. Vyhodnocení metody Blackman-Tukey. 66

SEZNAM TABULEK

<i>Tab. 1. Čas běhu metod pomocí funkce tic toc</i>	<i>52</i>
<i>Tab. 2. Suma čtverců odchylek</i>	<i>52</i>

SEZNAM PŘÍLOH

- P I Algoritmus metody periodogram.
- P II Algoritmus metody Bartlett.
- P III Algoritmus metody Blackman-Tukey.
- P IV Algoritmus metody nezkresleného odhadu.
- P V Algoritmus metody Blackman-Tukey 2.
- P VI Algoritmus metody minimálního rozptylu.
- P VII Algoritmus metody Welch.
- P VIII Algoritmus metody DFT.
- P IX Algoritmus metody Burg.

PŘÍLOHA Č I: ALGORITMUS METODY PERIODOGRAM

```
clc, clear, close all

% Odhad výkonové spektrální hustoty (Power Spectral Density - PSD) s vyu-
žitím metody periodogramu
%% generování signálu

fs = 400; % vzorkovací frekvence
f1 = 45; % frekvence první složky simulovaného signálu
f2 = 55; % frekvence druhé složky simulovaného signálu
f3 = 75; % frekvence třetí složky simulovaného signálu
T1 = 1/f1;
T2 = 1/f2;
T3 = 1/f3;
t = 0:1/fs:10*T1; % definice časové osy
L = numel(t); % počet prvků pole "t" = počet prvků simulovaného diskreti-
zovaného signálu
signal1 = sin(2*pi*f1*t); % definice první složky simulovaného signálu
signal2 = sin(2*pi*f2*t); % definice druhé složky simulovaného signálu
signal3 = 5*sin(2*pi*f3*t); % definice třetí složky simulovaného signálu

x = signal1 + signal2 + signal3; % výsledný simulovaný signál

% vykreslení simulovaného signálu
figure(1);
plot(t,x)
title('simulovaný signál'), xlabel('čas [s]'), ylabel('signál')

%% autokorelační funkce AF
tic % start funkce "tic toc" pro měření doby provedení operací

Rx = zeros(1, L); % vytvoření a naplnění pole Rx nulami
for i = 1:L % výpočet pravé strany AF
    % posun signálu doleva (složení signálu z původního signálu a nul)
    xnu = [x(i:end) zeros(1,i-1)];
    % součin původního a posunutého (zpožděného) signálu
    rx = xnu .* x;
    s = 0;
    for j = 1:L % suma ry
```

```

        s = s + rx(j);
    end
    Rx(i) = s;
end

Rx1 = zeros(1, L-1); % vytvoření a naplnění pole Rx1 nulami
for i = 1:L-1
    Rx1(i) = Rx(L-i+1); % získání levé strany AF ze strany pravé
end

Rx = [Rx1 Rx] ./L; % vytvoření AF z pravé a levé strany a dělení pomocí L

tau = -(L-1):L-1; % vytvoření pole posunutí (zpoždění) o velikosti 2*(L-1)+1
w = 0:0.01:pi; % vytvoření pole frekvencí, pro které se bude počítat PSD

% vykreslení autokorelační funkce
figure(2);
plot(tau,Rx)
title('AF'), xlabel('L'), ylabel('amp')

%% PSD
% rovnice (14)
PSD = zeros(1, length(w)); % vytvoření a naplnění pole PSD nulami
for i = 1:length(w) % smyčka pro frekvenci
    psd = 0;
    for j = 1:2*L-1 % smyčka pro sumu
        psd = psd + Rx(j) * exp(-1i*w(i)*tau(j));
    end
    PSD(i) = abs(psd)/pi;
end
toc % konec funkce "tic toc" pro měření doby provedení operací

w = w./pi/2*fs; % převod jednotek z rad/sec na Hz

% vykreslení PSD
figure(3);
plot(w,10*log10(PSD))
title('PSD'), xlabel('frekvence [Hz]'), ylabel('PSD [dB]')
```

PRÍLOHA Č II: ALGORITMUS METODY BARTLETT

```
clc, clear, close all

% Odhad výkonové spektrální hustoty (Power Spectral Density - PSD) s
% využitím metody Bartlett - průměrný Periodogram
%% generování signálu

fs = 400; % vzorkovací frekvence
f1 = 45; % frekvence první složky simulovaného signálu
f2 = 55; % frekvence druhé složky simulovaného signálu
f3 = 75; % frekvence třetí složky simulovaného signálu
T1 = 1/f1;
T2 = 1/f2;
T3 = 1/f3;
t = 0:1/fs:20*T1; % definice časové osy
L = numel(t); % počet prvků pole "t" = počet prvků simulovaného diskreti-
zovaného signálu
signal1 = sin(2*pi*f1*t); % definice první složky simulovaného signálu
signal2 = sin(2*pi*f2*t); % definice druhé složky simulovaného signálu
signal3 = 5*sin(2*pi*f3*t); % definice třetí složky simulovaného signálu

x = signal1 + signal2 + signal3; % výsledný simulovaný signál

% vykreslení simulovaného signálu
figure(1);
plot(t,x)
title('simulovaný signál'), xlabel('čas [s]'), ylabel('signál')

%% autokorelační funkce AF
tic % start funkce "tic toc" pro měření doby provedení operací

nb = 4; % 4 bloky délky L/nb
Lb = fix(L/nb); % délka bloku (zaokrouhlená na celá čísla dolů)
tau = -(Lb-1):Lb-1; % vytvoření pole posunutí (zpoždění) o velikosti
2*(Lb-1)+1

% výpočet AF pro každý blok
Rxb = zeros(length(tau), 4); % vytvoření a naplnění pole Rxb nulami
for j = 1:nb % rozdělení signálu na bloky
```

```

X = x((j-1)*Lb+1:j*Lb);

Rx = zeros(1, length(Lb)); % vytvoření a naplnění pole Rx nulami
for k = 1:Lb % výpočet pravé strany AF daného bloku
    % posun signálu doleva (složení signálu z původního signálu a
nul)
    Xtau = [X(k:end) zeros(1,k-1)];
    % součin původního a posunutého (zpožděného) signálu
    rx = Xtau .* X;
    s = 0;
    for l = 1:Lb % suma ry
        s = s + rx(l);
    end
    Rx(k) = s;
end

Rx1 = zeros(1, length(Lb)-1); % vytvoření a naplnění pole Rx1 nulami
for i = 1:Lb-1 % smyčka pro každý blok
    Rx1(i) = Rx(Lb-i+1); % získání levé strany AF ze strany pravé
end
% vytvoření AF z pravé a levé strany a dělení pomocí Lb
Rxb(:,j) = [Rx1 Rx] ./Lb;

end

w = -0:0.01:pi; % vytvoření pole frekvencí, pro které se bude počítat PSD

% vykreslení autokorelační funkce pro každý blok
figure(2);
plot(tau,Rxb)
title('AF'), xlabel('L'), ylabel('amp')

%% PSD

% rovnice (14)
PSD = zeros(length(w), nb); % vytvoření a naplnění pole PSD nulami
for i = 1:length(w) % smyčka pro frekvenci
    psd = zeros(1,4);
    for j = 1:2*Lb-1 % smyčka pro sumu
        psd = psd + Rxb(j,:) * exp(-1i*w(i)*tau(j));
    end
end

```

```

    % i-tý psd vzorek každého bloku uložený ve sloupcích PSD
    PSD(i,:) = abs(psd)/pi;
end

PSDa = zeros(length(w), 1); % vytvoření a naplnění pole PSDa nulami
for i = 1:length(w) % průměr PSD z bloků
    avg = 0;
    for j = 1:nb
        avg = avg + PSD(i,j);
    end
    PSDa(i) = avg; % výsledný zprůměrovaný periodogram
end
toc % konec funkce "tic toc" pro měření doby provedení operací

w = w./pi/2*fs; % převod jednotek z rad/sec na Hz

% vykreslení PSD
figure(3);
plot(w,10*log10(PSDa))
title('PSD'), xlabel('frekvence [Hz]'), ylabel('PSD [dB]')
```


PRÍLOHA Č III: ALGORITMUS METODY BLACKMAN-TUKEY

```
clc, clear, close all

% Odhad výkonové spektrální hustoty (Power Spectral Density - PSD) s
% využitím metody Blackman-Tukey
%% generování signálu

fs = 400; % vzorkovací frekvence
f1 = 45; % frekvence první složky simulovaného signálu
f2 = 55; % frekvence druhé složky simulovaného signálu
f3 = 75; % frekvence třetí složky simulovaného signálu
T1 = 1/f1;
T2 = 1/f2;
T3 = 1/f3;
t = 0:1/fs:10*T1; % definice časové osy
L = numel(t); % počet prvků pole "t" = počet prvků simulovaného diskreti-
zovaného signálu
signal1 = sin(2*pi*f1*t); % definice první složky simulovaného signálu
signal2 = sin(2*pi*f2*t); % definice druhé složky simulovaného signálu
signal3 = 5*sin(2*pi*f3*t); % definice třetí složky simulovaného signálu

x = signal1 + signal2 + signal3; % výsledný simulovaný signál

% vykreslení simulovaného signálu
figure(1);
plot(t,x)
title('simulovaný signál'), xlabel('čas [s]'), ylabel('signál')

%% autokorelační funkce AF
tic % start funkce "tic toc" pro měření doby provedení operací

win = 0.54 - 0.46*cos(2*pi*(0:L-1)/L); % Definice Hammingova okna
x = x .* win; % aplikace Hammingova okna na simulovaný signál - okénko-
vání

% vykreslení simulovaného signálu po aplikaci Hammingova okna
figure(2);
plot(t,x)
```

```

title('simulovaný signál po aplikaci Hammingova okna'), xlabel('čas
[s]'), ylabel('signál')

Rx = zeros(1, L); % vytvoření a naplnění pole Rx nulami
for i = 1:L % výpočet pravé strany AF
    % posun signálu doleva (složení signálu z původního signálu a nul)
    xtau = [x(i:end) zeros(1,i-1)];
    % součin původního a posunutého (zpožděného) signálu
    rx = xtau .* x;
    s = 0;
    for j = 1:L % suma ry
        s = s + rx(j);
    end
    Rx(i) = s;
end

Rx1 = zeros(1, L-1); % vytvoření a naplnění pole Rx1 nulami
for i = 1:L-1
    Rx1(i) = Rx(L-i+1); % získání levé strany AF ze strany pravé
end

Rx = [Rx1 Rx] ./L; % vytvoření AF z pravé a levé strany a dělení pomocí L

tau = -(L-1):L-1; % vytvoření pole posunutí (zpoždění) o velikosti 2*(L-
1)+1
w = 0:0.01:pi; % vytvoření pole frekvencí, pro které se bude počítat PSD

% vykreslení autokorelační funkce
figure(3);
plot(tau,Rx)
title('AF'), xlabel('L'), ylabel('amp')

%% PSD

% rovnice (14)
PSD = zeros(1, length(w)); % vytvoření a naplnění pole PSD nulami
for i = 1:length(w) % smyčka pro frekvenci
    psd = 0;
    for j = 1:2*L-1 % smyčka pro sumu
        psd = psd + Rx(j) * exp(-1i*w(i)*tau(j));
    end
end

```

```
end
    PSD(i) = abs(psd)/pi;
end
toc % konec funkce "tic toc" pro měření doby provedení operací

w = w./pi/2*fs; % převod jednotek z rad/sec na Hz

% vykreslení PSD
figure(4);
plot(w,10*log10(PSD))
title('PSD'), xlabel('frekvence [Hz]'), ylabel('PSD [dB]')
```

PRÍLOHA Č IV: ALGORITMUS METODY NEZKRESLENÉHO ODHADU

```
clc, clear, close all

% Odhad výkonové spektrální hustoty (Power Spectral Density - PSD) s
% využitím metody nezkresleného odhadu
%% generování signálu

fs = 400; % vzorkovací frekvence
f1 = 45; % frekvence první složky simulovaného signálu
f2 = 55; % frekvence druhé složky simulovaného signálu
f3 = 75; % frekvence třetí složky simulovaného signálu
T1 = 1/f1;
T2 = 1/f2;
T3 = 1/f3;
t = 0:1/fs:10*T1; % definice časové osy
L = numel(t); % počet prvků pole "t" = počet prvků simulovaného diskreti-
zovaného signálu
signal1 = sin(2*pi*f1*t); % definice první složky simulovaného signálu
signal2 = sin(2*pi*f2*t); % definice druhé složky simulovaného signálu
signal3 = 5*sin(2*pi*f3*t); % definice třetí složky simulovaného signálu

x = signal1 + signal2 + signal3; % výsledný simulovaný signál

% vykreslení simulovaného signálu
figure(1);
plot(t,x)
title('simulovaný signál'), xlabel('čas [s]'), ylabel('signál')

%% autokorelační funkce AF
tic % start funkce "tic toc" pro měření doby provedení operací

tau = -(L-1):L-1; % vytvoření pole posunutí (zpoždění) o velikosti 2*(L-
1)+1
win = 0.54 - 0.46*cos(2*pi*(0:L-1)/L); % Definice Hammingova okna
x = x .* win; % aplikace Hammingova okna na simulovaný signál - okénko-
vání

% vykreslení simulovaného signálu po aplikaci Hammingova okna
```

```

figure(2);
plot(t,x)
title('simulovaný signál po aplikaci Hammingova okna'), xlabel('čas
[s]'), ylabel('signál')

Rx = zeros(1, L); % vytvoření a naplnění pole Rx nulami
for i = 1:L
    % posun signálu doleva (složení signálu z původního signálu a nul)
    xtau = [x(i:end) zeros(1,i-1)];
    % součin původního a posunutého (zpožděného) signálu
    rx = xtau .* x;
    s = 0;
    for j = 1:L % suma ry
        s = s + rx(j);
    end
    Rx(i) = s;
end

Rx1 = zeros(1, L-1); % vytvoření a naplnění pole Rx1 nulami
for i = 1:L-1
    Rx1(i) = Rx(L-i+1); % získání levé strany AF ze strany pravé
end
% vytvoření AF z pravé a levé strany a dělení pomocí L-abs(nu)
Rx = [Rx1 Rx] ./ (L-abs(tau));

w = -0:0.01:pi; % vytvoření pole frekvencí, pro které se bude počítat PSD

% vykreslení autokorelační funkce
figure(3);
plot(tau,Rx)
title('AF'), xlabel('L'), ylabel('amp')

%% PSD

% rovnice (14)
PSD = zeros(1, length(w)); % vytvoření a naplnění pole PSD nulami
for i = 1:length(w) % smyčka pro frekvenci
    psd = 0;
    for j = 1:2*L-1 % smyčka pro sumu
        psd = psd + Rx(j) * exp(-li*w(i)*tau(j));
    end
end

```

```
end
    PSD(i) = abs(psd)/pi;
end
toc % konec funkce "tic toc" pro měření doby provedení operací

w = w./pi/2*fs; % převod jednotek z rad/sec na Hz

% vykreslení PSD
figure(4);
plot(w,10*log10(PSD))
title('PSD'), xlabel('frekvence [Hz]'), ylabel('PSD [dB]')
```

PRÍLOHA Č V: ALGORITMUS METODY BLACKMAN-TUKEY 2

```
clc, clear, close all

% Odhad výkonové spektrální hustoty (Power Spectral Density - PSD) s
% využitím metody Blackman-Tukey 2
%% generování signálu

fs = 400; % vzorkovací frekvence
f1 = 45; % frekvence první složky simulovaného signálu
f2 = 55; % frekvence druhé složky simulovaného signálu
f3 = 75; % frekvence třetí složky simulovaného signálu
T1 = 1/f1;
T2 = 1/f2;
T3 = 1/f3;
t = 0:1/fs:10*T1; % definice časové osy
L = numel(t); % počet prvků pole "t" = počet prvků simulovaného diskreti-
zovaného signálu
signal1 = sin(2*pi*f1*t); % definice první složky simulovaného signálu
signal2 = sin(2*pi*f2*t); % definice druhé složky simulovaného signálu
signal3 = 5*sin(2*pi*f3*t); % definice třetí složky simulovaného signálu

x = signal1 + signal2 + signal3; % výsledný simulovaný signál

% vykreslení simulovaného signálu
figure(1);
plot(t,x)
title('simulovaný signál'), xlabel('čas [s]'), ylabel('signál')

%% autokorelační funkce AF
tic % start funkce "tic toc" pro měření doby provedení operací

tau = -(L-1):L-1; % vytvoření pole posunutí (zpoždění) o velikosti 2*(L-
1)+1
win = 0.54 - 0.46*cos(2*pi*(0:2*L-2)/2/L); % Definice Hammingova okna

Rx = zeros(1, L); % vytvoření a naplnění pole Rx nulami
for i = 1:L % výpočet pravé strany AF
    % posun signálu doleva (složení signálu z původního signálu a nul)
    xtau = [x(i:end) zeros(1,i-1)];
```

```

% součin původního a posunutého (zpožděného) signálu
rx = xtau .* x;
s = 0;
for j = 1:L % suma ry
    s = s + rx(j);
end
Rx(i) = s;
end

Rx1 = zeros(1, L-1); % vytvoření a naplnění pole Rx1 nulami
for i = 1:L-1
    Rx1(i) = Rx(L-i+1); % získání levé strany AF ze strany pravé
end
Rx = [Rx1 Rx] ./L; % vytvoření AF z pravé a levé strany a dělení pomocí L
Rx = Rx .* win; % násobení AF Hammingovým oknem

w = -0:0.01:pi; % vytvoření pole frekvencí, pro které se bude počítat PSD

% vykreslení autokorelační funkce
figure(2);
plot(tau,Rx)
title('AF'), xlabel('L'), ylabel('amp')

%% PSD

% rovnice (14)
PSD = zeros(1, length(w)); % vytvoření a naplnění pole PSD nulami
for i = 1:length(w) % smyčka pro frekvenci
    psd = 0;
    for j = 1:2*L-1 % smyčka pro sumu
        psd = psd + Rx(j) * exp(-1i*w(i)*tau(j));
    end
    PSD(i) = abs(psd)/pi;
end
toc % konec funkce "tic toc" pro měření doby provedení operací

w = w./pi/2*fs; % převod jednotek z rad/sec na Hz

% vykreslení PSD

```



```
figure(3);  
plot(w,10*log10(PSD))  
title('PSD'), xlabel('frekvence [Hz]'), ylabel('PSD [dB]')
```

PRÍLOHA Č VI: ALGORITMUS METODY MINIMÁLNÍHO ROZPTYLU

```
clc, clear, close all

% Odhad výkonové spektrální hustoty (Power Spectral Density - PSD) s
% využitím metody minimálního rozptylu
%% generování signálu

fs = 400; % vzorkovací frekvence
f1 = 45; % frekvence první složky simulovaného signálu
f2 = 55; % frekvence druhé složky simulovaného signálu
f3 = 75; % frekvence třetí složky simulovaného signálu
T1 = 1/f1;
T2 = 1/f2;
T3 = 1/f3;
t = 0:1/fs:10*T1; % definice časové osy
L = numel(t); % počet prvků pole "t" = počet prvků simulovaného diskreti-
zovaného signálu
signal1 = sin(2*pi*f1*t); % definice první složky simulovaného signálu
signal2 = sin(2*pi*f2*t); % definice druhé složky simulovaného signálu
signal3 = 5*sin(2*pi*f3*t); % definice třetí složky simulovaného signálu

x = signal1 + signal2 + signal3; % výsledný simulovaný signál

% vykreslení simulovaného signálu
figure(1);
plot(t,x)
title('simulovaný signál'), xlabel('čas [s]'), ylabel('signál')

%% metoda minimálního rozptylu
tic % start funkce "tic toc" pro měření doby provedení operací

avg = 0;
for j = 1:L
    avg = avg + x(j)/L; % výpočet průměrné hodnoty signálu
end
% transpozice vektoru signálu a odečtení průměrné hodnoty signálu od něj
wx = x.' - avg;
```

```

% sestavení vektoru z nul a vektoru wx a zpětná transpozice
wx_pad = [zeros(L-1,1);wx;zeros(L-1,1)].';

XU = zeros(2*L-1, L); % vytvoření a naplnění pole XU nulami
for i = 1:L
    % vytvoření matice XU z vektoru wx_pad (matice s posuny vektoru x)
    XU(:,i) = wx_pad(L-i+1:3*L-i-1);
end

% vytvoření matice Rwx násobením matic XU s XU'
Rwx = XU'*XU / (L-1);

% získání vlastních vektorů (eigenvectors) a vlastních hodnot (eigenvalues)
% matice Rwy s využitím příkazu eig
[eigvec, eigval] = eig(Rwx);

w = 0:0.01:pi; % vytvoření pole frekvencí, pro které se bude počítat PSD
U = zeros(L,1);
PSD = zeros(numel(w),1); % vytvoření a naplnění pole PSD nulami

% výpočet rovnice (19) pro různé frekvence
for j = 1:numel(w)
    for l = 1:L
        u = 0;
        for k = 1:L
            % výpočet  $e(e^{jw}) * q_l$  z rovnice (19)
            u = u + eigvec(k,l) * exp(1i*(k-1)*w(j));
        end
        % výpočet jmenovatele z rovnice (19):  $|e(e^{jw}) * q_l|^2 / \lambda_l$ 
        U = abs(u).^2 ./ (eigval(l,l) + eps);
        PSD(j) = PSD(j) + U;
    end
end

PSD = L ./ PSD;
toc % konec funkce "tic toc" pro měření doby provedení operací

w = w./pi/2*fs; % převod jednotek z rad/sec na Hz

% vykreslení PSD
figure(2);

```

```
plot(w,10*log10(PSD))  
title('PSD'), xlabel('frekvence [Hz]'), ylabel('PSD [dB]')
```

PRÍLOHA Č VII: ALGORITMUS METODY WELCH

```
clc, clear, close all

% Odhad výkonové spektrální hustoty (Power Spectral Density - PSD) s
% využitím metody Welch
%% generování signálu

fs = 400; % vzorkovací frekvence
f1 = 45; % frekvence první složky simulovaného signálu
f2 = 55; % frekvence druhé složky simulovaného signálu
f3 = 75; % frekvence třetí složky simulovaného signálu
T1 = 1/f1;
T2 = 1/f2;
T3 = 1/f3;
t = 0:1/fs:20*T1; % definice časové osy
L = numel(t); % počet prvků pole "t" = počet prvků simulovaného diskreti-
zovaného signálu
signal1 = sin(2*pi*f1*t); % definice první složky simulovaného signálu
signal2 = sin(2*pi*f2*t); % definice druhé složky simulovaného signálu
signal3 = 5*sin(2*pi*f3*t); % definice třetí složky simulovaného signálu

x = signal1 + signal2 + signal3; % výsledný simulovaný signál

% vykreslení simulovaného signálu
figure(1);
plot(t,x)
title('simulovaný signál'), xlabel('čas [s]'), ylabel('signál')

%% metoda Welch
tic % start funkce "tic toc" pro měření doby provedení operací

N = 30; % délka Hannova okna
win = 0.5*(1-cos(2*pi*(0:N-1)/N)); % definice Hannova okna
k = fix(length(x)/(N/2))-1; % počet segmentů k
% definice indexové proměnné, která se bude pohybovat po vzorcích
index = 1:N;

w = 0:0.01:pi; % vytvoření pole frekvencí, pro které se bude počítat PSD
PSD = zeros(1, numel(w)); % vytvoření a naplnění pole PSD nulami
```

```

XW = zeros(1, numel(w)); % vytvoření a naplnění pole XW nulami

for i = 1:k
    % postupné násobení vzorků x s Hannovým oknem
    xw = win.*x(index);
    index = index + N/2; % pokračování do dalšího okna (50% překrytí)
    % rovnice (17), jde o výpočet periodogramu xw
    for j = 1:length(w) % smyčka pro frekvenci
        Xw = 0;
        for n = 1:N % smyčka pro sumu
            Xw = Xw + xw(n) * exp(-1i*w(j)*(n-1));
        end
        XW(j) = abs(Xw)/pi;
    end

    PSD = PSD + XW/k; % rovnice (18)
end

% výpočet PSD pomocí vestavěné funkce pwelch
[PSDwelch, F] = pwelch(x, win, 9);
toc % konec funkce "tic toc" pro měření doby provedení operací

w = w./pi/2*fs; % převod jednotek z rad/sec na Hz

% vykreslení PSD
figure(2),
plot(w, 10*log10(abs(PSD)))
title('PSD'), xlabel('frekvence [Hz]'), ylabel('PSD [dB]')

```

PRÍLOHA Č VIII: ALGORITMUS METODY DFT

```
clc, clear, close all

% Odhad výkonové spektrální hustoty (Power Spectral Density - PSD) s
% využitím metody DFT
%% generování signálu

fs = 400; % vzorkovací frekvence
f1 = 45; % frekvence první složky simulovaného signálu
f2 = 55; % frekvence druhé složky simulovaného signálu
f3 = 75; % frekvence třetí složky simulovaného signálu
T1 = 1/f1;
T2 = 1/f2;
T3 = 1/f3;
t = 0:1/fs:10*T1; % definice časové osy
L = numel(t); % počet prvků pole "t" = počet prvků simulovaného diskreti-
zovaného signálu
signal1 = sin(2*pi*f1*t); % definice první složky simulovaného signálu
signal2 = sin(2*pi*f2*t); % definice druhé složky simulovaného signálu
signal3 = 5*sin(2*pi*f3*t); % definice třetí složky simulovaného signálu

x = signal1 + signal2 + signal3; % výsledný simulovaný signál

% vykreslení simulovaného signálu
figure(1);
plot(t,x)
title('simulovaný signál'), xlabel('čas [s]'), ylabel('signál')

%% metoda DFT
tic % start funkce "tic toc" pro měření doby provedení operací

tau = 0:L-1; % vytvoření pole posunutí (zpoždění) o velikosti L-1
w = -0:0.01:pi; % vytvoření pole frekvencí, pro které se bude počítat PSD

DFT = zeros(numel(w),1); % vytvoření a naplnění pole DFT nulami

% výpočet DFT
for i = 1:length(w) % smyčka pro frekvenci
    psd = 0;
```

```
for j = 1:L-1 % smyčka pro sumu
    psd = psd + x(j) * exp(-1i*w(i)*tau(j));
end
DFT(i) = psd;
end
toc % konec funkce "tic toc" pro měření doby provedení operací

w = w./pi/2*fs; % převod jednotek z rad/sec na Hz

% vykreslení DFT
figure(2),
plot(w, 20*log10(abs(DFT)))
title('PSD'), xlabel('frekvence [Hz]'), ylabel('PSD [dB]')
```


PRÍLOHA Č IX: ALGORITMUS METODY BURG

```
clc, clear, close all

% Odhad výkonové spektrální hustoty (Power Spectral Density - PSD) s
% využitím metody Burg
%% generování signálu

fs = 400; % vzorkovací frekvence
f1 = 45; % frekvence první složky simulovaného signálu
f2 = 55; % frekvence druhé složky simulovaného signálu
f3 = 75; % frekvence třetí složky simulovaného signálu
T1 = 1/f1;
T2 = 1/f2;
T3 = 1/f3;
t = 0:1/fs:10*T1; % definice časové osy
L = numel(t); % počet prvků pole "t" = počet prvků simulovaného diskreti-
zovaného signálu
signal1 = sin(2*pi*f1*t); % definice první složky simulovaného signálu
signal2 = sin(2*pi*f2*t); % definice druhé složky simulovaného signálu
signal3 = 5*sin(2*pi*f3*t); % definice třetí složky simulovaného signálu

x = signal1 + signal2 + signal3; % výsledný simulovaný signál

% vykreslení simulovaného signálu
figure(1);
plot(t,x)
title('simulovaný signál'), xlabel('t'), ylabel('signál')

%% metoda Burg
tic % start funkce "tic toc" pro měření doby provedení operací

N = 4; % pořadí modelu lineární predikce (LP)

xf = x(1:end-1); % x_f[0]
xb = x(2:end); % x_b[0]
k = zeros(1, numel(N)); % vytvoření a naplnění pole k nulami
k(1) = -2*(xf * xb')/(xb*xb' + xf*xf'); % rovnice (20)

for i = 1:N
```

```

Aest = zeros(4); % vytvoření a naplnění pole Aest nulami
for j = 1:i
    % rovnice (21)
    if i==j
        Aest(j,i) = k(i);
    else
        Aest(j,i) = Aest(j,i-1) + k(i)*Aest(i-j,i-1);
    end
end
xf = xf(1:end-i); % x_f[i] (n)
xb = xb(i+1:end); % x_b[i] (n)

xf = xf + k(i) * xb;
xb = xb + k(i) * xf;
k(i+1) = -2*(xf * xb')/(xb*xb' + xf*xf'); % rovnice (20)
end

w = 0:0.01:pi; % vytvoření pole frekvencí, pro které se bude počítat PSD
PSD = zeros(numel(w),1); % vytvoření a naplnění pole PSD nulami
% přenosová funkce
% tato část se rovná výstupu z funkce freqz()
for j = 1:numel(w)
    Hjw = 1;
    for i = 1:N
        Hjw = Hjw + Aest(i,N)*exp(-li*w(j)*i);
    end
    PSD(j) = 1/Hjw;
end

% výpočet PSD pomocí vestavěné funkce pburg
[PSDburg ,fburg] = pburg(x,4);
toc % konec funkce "tic toc" pro měření doby provedení operací

fburg = fburg./pi/2*fs; % převod jednotek z rad/sec na Hz

% vykreslení PSD
figure(2)
plot(fburg, 20*log10(abs(PSDburg)))
title('PSD'), xlabel('frekvence [Hz]'), ylabel('PSD [dB]')

```