

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2023

Vít Šimeček



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

DISKRÉTNÍ SLEPIANOVA ORTOGONÁLNÍ BÁZE A JEJÍ VYUŽITÍ PRO ZÍSKÁNÍ SPEKTRA SIGNÁLŮ

APPLICATION OF SLEPIAN SEQUENCES FOR SPECTRAL ANALYSIS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Vít Šimeček

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Ondřej

Mihálik

BRNO 2023

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Vít Šimeček

ID: 230187

Ročník: 3

Akademický rok: 2022/23

NÁZEV TÉMATU:

Diskrétní Slepianova ortogonální báze a její využití pro získání spektra signálů

POKyny PRO VYPRACOVÁNÍ:

Cílem bakalářské práce je nastudovat vlastnosti diskrétní Slepianovy ortogonální báze a aplikovat je při tvorbě souboru funkcí a skriptů, které by umožňovaly analýzu vybraných signálů. Předpokládá se implementace v prostředí MATLAB.

1. Seznamte se s pojmy „signály s omezeným spektrem“ a „Hilbertův prostor l^2 “.
2. Seznamte se s funkcí „dps“ v prostředí MATLAB a s její pomocí generujte ortonormální Slepianovu bázi.
3. Prostudujte některé vlastnosti Slepianovy ortogonální báze, např. prověření ortogonalitu, spektrum, definiční konvoluční vztah... Na zvolených příkladech tyto vlastnosti demonstřujte.
4. Pro zvolený signál spočítejte jeho Slepianovu transformaci. Získané spektrum porovnejte s klasickým Fourierovým spektrem.
5. Provéřte možnost využití Slepianovy báze pro extrapolaci frekvenčně omezeného signálu.
6. Zhodnoťte dosažené výsledky a doplňte je grafickými ukázkami.

DOPORUČENÁ LITERATURA:

- [1] HARDY, Godfrey Harold a Werner ROGOSINSKI. Fourierovy řady. Přeložil Alois KUFNER. Praha: Nakladatelství technické literatury, 1971.
- [2] MIHÁLIK, O.; MICHALÍK, D.; JURA, P.; FIEDLER, P. Spectral Super-Resolution and Band-Limited Extrapolation Using Slepian Series. In 16th IFAC Conference on Programmable Devices and Embedded Systems – PDeS 2019. IFAC-PapersOnLine (ELSEVIER). 52. IFAC, 2019. s. 388-393. ISSN: 2405-8963.

Termín zadání: 6.2.2023

Termín odevzdání: 22.5.2023

Vedoucí práce: Ing. Ondřej Mihálik

doc. Ing. Václav Jirsík, CSc.
předseda rady studijního programu

UPOZORNĚNÍ:

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

ABSTRAKT

Práce se týká zpracování diskrétních signálů pomocí Slepianovy transformace, což je varianta získání spektra signálu. Signál je rozložen do Slepianovy ortogonální báze, která je v Matlabu generována funkcí "DPSS". Dále je v práci toto spektrum v práci srovnáno s klasickým Fourierovým spektrem, kosinovým spektrem. Je provedena komprese dat a srovnání bází. Pomocí Slepianovy báze je extrapolován jednoduchý signál.

KLÍČOVÁ SLOVA

Hilbertův prostor ℓ_2 , Zpracování diskrétních signálů, funkce DPSS v Matlabu, Slepianova transformace, frekvenčně omezený signál, spektrum, Fourierova transformace, kosinová transformace, extrapolace signálu, komprese dat, aproximace signálu

ABSTRACT

This thesis is focused on the Digital signal processing using the Slepian transform, which is a option of gaining a signal spectrum. The signal is transformed to the Slepian orthogonal base, which can be generated in Matlab by the function "DPSS". Furthermore, this spectrum is in the thesis compared with classic Fourier spectrum and cosine spectrum. Further, a data compression is computed and bases are compared to each other. By the Slepian base, a simple signal is extrapolated.

KEYWORDS

Hilbert space ℓ_2 , Discrete signal processing, DPSS function in Matlab, Slepian transform, band-limited signal, spectrum, Fourier transform, cosine transform, signal extrapolation, data compression, signal approximation

ŠIMEČEK, Vít. *Diskrétní Slepianova ortogonální báze a její využití pro získání spektra signálů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2023, 54 s. Bakalářská práce. Vedoucí práce: Ing. Ondrej Mihálik

Prohlášení autora o původnosti díla

Jméno a příjmení autora: Vít Šimeček
VUT ID autora: 230187
Typ práce: Bakalářská práce
Akademický rok: 2022/23
Téma závěrečné práce: Diskrétní Slepianova ortogonální báze a její využití pro získání spektra signálů

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora*

* Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Ondreji Mihálikovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Úvod	11
1 Základní pojmy a definice	12
1.1 Signály s omezeným spektrem	12
1.2 Hilbertův prostor ℓ_2	13
2 Slepianovy sekvence – „DPSS“	15
2.1 Definiční konvoluční vztah DPSS	15
2.2 Spektrum spojité Slepianovy transformace	16
2.3 Funkce „DPSS“ v prostředí Matlab	16
2.4 Prověření ortogonality Slepianovy báze	17
3 Slepianova transformace	21
3.1 Definice DFT	21
3.2 Slepianova transformace zvoleného signálu	22
3.3 Inverzní Slepianova transformace	23
3.4 Rychlá Fourierova transformace FFT	23
3.5 Rozdíly ve spektrech	27
3.6 Srovnání Slepianovy transformace s FFT	27
4 Využití „DPSS“ při extrapolaci signálu	33
4.1 Definice extrapolace	33
4.2 Použitý algoritmus pro extrapolaci	33
4.3 Nedodržení parametrů	37
5 Využití „DPSS“ při kompresi signálu ze simulátoru řízení vozidla	41
5.1 Diskrétní kosinová transformace	41
5.2 Simulátor řízení vozidla	42
5.3 Kompresi naměřených dat	43
Závěr	50
Literatura	51
Seznam symbolů a zkratk	53
A Obsah elektronické přílohy	54

Seznam obrázků

2.1	Slepianova ortonormální báze, prvních 5 sekvencí.	18
2.2	Důkaz ortonormality Slepianovy ortonormální báze.	20
2.3	Chyba ortonormality Slepianovy báze. Jedná se o výpočet rozdílu matic $\mathbf{V} - \mathbf{I}$	20
3.1	Navzorkovaná část signálu $f(t)$. Jedná se o 127 vzorků.	23
3.2	Výsledný signál z inverzní Slepianovy transformace. Každá iterace algoritmu přidává jednu složku spektra vynásobenou příslušným koeficientem a odpovídá příslušnému řádu aproximace. Zobrazeno prvních cca 50 aproximací signálu.	24
3.3	Výsledný signál z inverzní Slepianovy transformace, IFFT a původní signál. U obou transformací byly použity úplné báze/všechny složky spektra.	24
3.4	Výsledný signál z inverzní FFT transformace. Každá iterace algoritmu přidává jednu složku spektra – roste řád aproximace. Zobrazeno prvních 42 aproximací signálu.	25
3.5	Výsledkem Slepianovy transformace signálu $f(t)$ je jeho spektrum. . .	26
3.6	FFT spektrum amplitudové (vertikální osa vlevo) a fázové (osa vpravo). 26	
3.7	Průběhy odchylky iterační zpětné transformace a původního signálu $f(t)$	29
3.8	Řád aproximace, kdy pro FFT a Slepianovu bázi maximální odchylka splnila podmínku (vysvětleno na začátku sekce 3.6).	30
3.9	Řád aproximace, kdy pro FFT a Slepianovu bázi průměrná odchylka splnila podmínku (vysvětleno na začátku sekce 3.6).	31
3.10	Průměrný řád aproximace, kdy pro FFT a Slepianovu bázi průměrná odchylka splnila podmínku (vysvětleno na začátku sekce 3.6).	31
3.11	Histogram řádu aproximace, kdy pro FFT a Slepianovu bázi maximální odchylka splnila podmínku (vysvětleno na začátku sekce 3.6). . .	32
3.12	Histogram řádu aproximace, kdy pro FFT a Slepianovu bázi průměrná odchylka splnila podmínku (vysvětleno na začátku sekce 3.6). . .	32
4.1	Prodloužená Slepianova báze na 401 vzorků symetricky kolem původních 127 vzorků. ψ_0 až ψ_4	34
4.2	Prodloužená Slepianova báze na 401 vzorků symetricky kolem původních 127 vzorků. ψ_{10} až ψ_{14}	34
4.3	Ukázka hodnot λ při parametru $NW = 6,4$. Je patrné, že některé hodnoty nebyly vypočteny (v Matlabu jsou nulové).	35

4.4	Extrapolace signálu $f(t)$, $NW = 6, 4$. Svislými čarami je naznačen interval měřeného periodického signálu. Extrapolace je velmi přesná pro cca dalších 16 vzorků (samples).	37
4.5	Extrapolace signálu $f(t)$, $NW = 2, 5$. Extrapolovaný signál dosahuje hodnot o několik řádů vyšších.	38
4.6	Extrapolace signálu $f(t)$, $NW = 2, 5$, detail. Výsledek nelze považovat za rekonstrukci signálu $f(t)$	38
4.7	Extrapolace signálu $f(t)$, $NW = 30$. V porovnání s obr. 4.4 však extrapolace není tak kvalitní.	39
4.8	Závislost délky extrapolace při definované kvadratické odchylce na parametru NW . Tečkovaně naznačena používaná hodnota $NW = 6, 4$	40
5.1	Realizace simulátoru řízení vozidla, zdroj obrázku: [15].	43
5.2	První signál je poloha vozidla na silnici vzhledem k její šířce.	44
5.3	Druhý signál symbolizuje natočení volantu během řízení, matematicky vzato se jedná o derivaci signálu na obr. 5.2.	44
5.4	Výřez signálu, kde bylo použito necelé spektrum, a sice pouze 1/4 spektra.	47
5.5	Výřez signálu, kde bylo použito necelé spektrum, a sice pouze 1/6 spektra. Detail ukazuje vyhlazení průběhu (odstranění šumu díky odstranění vyšších frekvenčních složek).	47
5.6	Výřez signálu, kde bylo použito necelé spektrum, a sice pouze 1/6 spektra. Detail ukazuje vyhlazení průběhu (odstranění šumu díky odstranění vyšších frekvenčních složek).	48
5.7	Závislost průměrné globální kvadratické chyby na řádu aproximace signálu.	48
5.8	Průměrný řád aproximace, kdy pro FFT, CT a ST průměrná odchylka splnila podmínku (odpovídá srovnání v sekci 3.6).	49
5.9	Histogram průměrného řádu aproximace, kdy pro FFT, CT a ST průměrná odchylka splnila podmínku (odpovídá srovnání v sekci 3.6) Fourierova transformace opět využívá symetrie spektra, tedy pouze 1/2 spektrálních koeficientů a její nejvyšší řád aproximace je 64.	49

Seznam výpisů

2.1	Generování Slepianovy ortogonální báze v Matlabu.	18
3.1	Výpočet koeficientů Slepianova spektra.	22
3.2	Maticové násobení v Matlabu.	23
3.3	Výpočet odchylky v jednotlivých aproximacích signálu.	28
3.4	Generování 100 harmonických signálů.	28
4.1	Určení nejvyšší nenulové λ	35
4.2	Výpočet prodloužených ψ	36
4.3	Inverzní Slepianova transformace prodloužených ψ	36

Úvod

Ke zpracování signálů se obvykle používá Fourierova transformace v jedné z jejích variant, například rychlá Fourierova transformace (FFT) se obvykle používá při zpracování signálů digitálními zařízeními. Tato práce se bude zabývat jednou z méně obvyklých alternativ, což je Slepianova transformace. Týkat se bude zpracování diskrétních signálů, což je důležitější oblast oproti zpracování spojitých signálů, neboť drtivá většina aplikací, které zpracování vyžadují, je digitálních.

V první kapitole budou definovány důležité pojmy z oblasti zpracování signálů a odpovídajícího matematického aparátu.

Dále ve druhé kapitole bude definována Slepianova transformace a Slepianova ortogonální báze, která bude posléze generována pomocí funkce „DPSS“ v Matlabu. Budou prověřeny některé její vlastnosti definované v první a na začátku druhé kapitoly, např. ortogonalita.

Třetí kapitola se zabývá srovnáním Slepianovy a Rychlé Fourierovy transformace, a sice budou zde porovnána spektra, a také bude signál zpětně rekonstruován. Zpětná rekonstrukce signálu proběhne i po jednotlivých řádech aproximace, kdy budou postupně k rekonstruovanému signálu přidávány složky spektra, čímž bude možné porovnat, která z možností rychleji konverguje k původnímu signálu.

Ve čtvrté kapitole bude popsána možnost extrapolace signálu mimo interval pozorování za pomoci Slepianovy báze včetně ukázky podmínek, které musí být splněny, aby byla extrapolace úspěšná.

Na závěr bude rozvinuta třetí kapitola, a sice na datech ze simulátoru řízení vozidla bude provedena komprese spektra – Slepianova, Fourierova a kosinového, jelikož kosinová transformace je jeden z nástrojů využívaných při kompresi – jednotlivé transformace budou srovnány co se týče možnosti komprese spektra signálů a kvality zpětné rekonstrukce signálu z komprimovaného spektra.

1 Základní pojmy a definice

1.1 Signály s omezeným spektrem

Práce se bude týkat výhradně diskretních signálů s omezeným spektrem, proto budou na začátku vysvětleny některé důležité pojmy a souvislosti, které se signálů s omezeným spektrem týkají.

Definice 1.1.1 *Jako signál s omezeným spektrem (angl. Bandlimited signal) je definován signál v časové doméně $f(t)$, jehož frekvenční spektrum $F(\omega)$ je pro kmitočty vyšší než Ω nulové. Kmitočet Ω je pak nazýván bandlimit signálu. To je popsáno také v (1.1), [1].*

$$|\omega| > \Omega \Rightarrow F(\omega) = 0 \quad (1.1)$$

Diskretní signály se často získávají pomocí vzorkování spojitých signálů. Následující definice proto uvádí Vzorkovací teorém (též Shannon–Nyquist, Shannon–Kotelnik teorém):

Definice 1.1.2 *Máme-li spojitý signál $f(t)$, který je zároveň frekvenčně omezený nejvyšší úhlovou frekvencí Ω [rad/s], vzorkovací frekvence ω_s musí být větší než dvojnásobek Ω , což je také zapsáno v (1.2), [1].*

$$\omega_s > 2 \cdot \Omega \quad (1.2)$$

Definice 1.1.3 *Normalizovaná šířka pásma (normalized bandwidth) W je definována pomocí normalizované frekvence z následujícího vztahu:*

$$\Omega = 2\pi f_{max} = 2\pi f_s W = 2\pi W/T_S \quad (1.3)$$

Z čehož po úpravě vyplývá:

$$W = \frac{\Omega}{2\pi f_s} = \frac{\Omega T_S}{2\pi} \quad (1.4)$$

Bude-li normalizovaná ω_S rovna jedné, pak může W nabývat hodnoty nejvýše 1/2 (viz 1.1.2).

Je-li vzorkovací teorém splněn, lze opětovně rekonstruovat signál $f(t)$ ze vzorkované sekvence $f(nT_s)$ se vzorkovací periodou T_s . Pokud splněn není, dojde k tzv. *aliasing efektu* a k překrytí frekvenčních spekter, čímž se ztratí informace o překrývajících se složkách spektra a znemožní se rekonstrukce signálu.

1.2 Hilbertův prostor ℓ_2

V literatuře se setkáváme s pojmem Hilbertova prostoru L_2 pro spojité funkce. Jeho obdobou je prostor ℓ_2 diskrétních funkcí:

Definice 1.2.1 Označme jako $\ell_2(a, b)$ množinu všech měřitelných diskrétních komplexních funkcí, pro které je suma (1.5) konečná [2].

$$\sum_{k=a}^b |f(k)|^2 \quad (1.5)$$

Dále definujme skalární součin funkcí náležících do $\ell_2(a, b)$ jako (1.6).

Definice 1.2.2 Skalární součin v prostoru $\ell_2(a, b)$ je roven (1.6), kde pruh nad $g(k)$ značí komplexně sdružené hodnoty. [2].

$$(f, g) = \sum_{k=a}^b f(k) \overline{g(k)} \quad (1.6)$$

Věta 1.2.3 Pro skalární součin v prostoru $\ell_2(a, b)$ platí Schwarzova nerovnost:

$$|(f, g)| \leq \|f\| \cdot \|g\| \quad (1.7)$$

Ještě je třeba znát normu neboli velikost vektoru v prostoru ℓ_2 , obecně $\|x\|_{\ell_2}$.

Definice 1.2.4 Norma funkce $f(k)$ v prostoru $\ell_2(a, b)$ je rovna:

$$\|f(k)\|_{\ell_2(a, b)} = \sqrt{\sum_{k=a}^b |f(k)|^2} \quad (1.8)$$

Příčemž nyní budou snáze vysvětleny pojmy ortogonalita a ortonormalita.

Definice 1.2.5 Dvě různé diskrétní funkce $f(k)$ a $g(k)$ jsou ortogonální na intervalu (a, b) , platí-li:

$$(f, g) = \sum_{k=a}^b f(k) \overline{g(k)} = 0 \quad (1.9)$$

Což znamená, že jejich skalární součin (1.6) je roven 0.

Definice 1.2.6 Nechť platí, že norma funkce je rovna 1:

$$\|e(k)\|^2 = 1 \quad (1.10)$$

Poté může být funkce $e(k)$ nazývána normalizovanou funkcí. Z funkce $f(k)$ lze pak vytvořit normalizovanou $e(k)$ následovně:

$$e(k) = f(k) \cdot \|f(k)\|^{-1} \quad (1.11)$$

Definice 1.2.7 *Vzájemně ortogonální sekvence e_1, e_2, \dots, e_N tvoří ortogonální bázi (jsou-li normalizované, pak se jedná o ortonormální bázi) v ℓ_2 , platí-li alespoň jedna z následujících podmínek:*

1. *Pro každou sekvenci $f \in \ell_2$ platí:*

$$\|f(k)\|^2 = \sum_{n=1}^{\infty} \frac{|(f, e_n)|^2}{\|e_n\|^2} \quad (1.12)$$

2. *Pro každou sekvenci $f \in \ell_2$ platí:*

$$\|f(k)\|^2 = \sum_{n=1}^{\infty} \frac{|(f, e_n)|}{\|e_n\|^2} \cdot e_n \quad (1.13)$$

3. *Pro každé $f \in \ell_2$ a $\epsilon > 0$ existuje konečná lineární kombinace:*

$$f_N = \sum_{n=1}^N c_n e_n \quad (1.14)$$

pro niž platí: $\|f - f_N\| < \epsilon$.

Výše uvedené definice jsou převzaty z [2] a [3].

2 Slepianovy sekvence – „DPSS“

Slepianovy sekvence nebo-li Discrete Prolate Spheroidal Sequences (DPSS) jsou jedním z nástrojů vhodných pro získání spektra signálu. Charakteristické jsou tím, že pracují v diskrétní oblasti a na omezeném časovém intervalu odpovídajícímu určitému počtu vzorků $N = F_s \cdot T$ (v následující práci bude uvažován normalizovaný časový interval $t = \langle -1; +1 \rangle$, nebude-li stanoveno jinak). To je rozdíl kupříkladu oproti Fourierově transformaci, kde potřebujeme znát signál v čase $t = (-\infty; +\infty)$, takže mimo okno pozorování signálu musíme například předpokládat, že je signál nulový, předpokládat jeho periodicitu apod. Signál, jehož spektrum se pomocí DPSS snažíme vyjádřit, musí mít frekvenčně omezené spektrum – musí být *bandlimited* signálem. Z podstaty vzorkování, viz definice 1.1.2, výše uvedené splňují všechny vzorkované signály (vyšší frekvenční složky nejsou kvůli Shannon–Nyquistova teorému zaznamenány) [4].

2.1 Definiční konvoluční vztah DPSS

Definice 2.1.1 *Diskrétní Slepianovy sekvence $\psi_n(K, W, l)$ jsou definovány jako reálné řešení soustavy rovnic pro každé reálné k :*

$$\sum_{l=1}^K \frac{\sin(2\pi W(k-l))}{\pi(k-l)} \psi_n(K, W, l) = \lambda_n(K, W) \psi_n(K, W, k) \quad (2.1)$$

kde k odpovídá diskrétnímu času (násobku vzorkovací periody),

K je celkový počet vzorků v čase (délka diskrétního signálu),

ψ_n je n -tá Slepianova sekvence,

λ_n jsou vlastní čísla nabývající sestupných kladných hodnot. Vyjadřují koncentraci sekvencí,

*W zde odpovídá normalizované šířce pásma *bandwidth*.*

Pozn.: Máme-li normalizovanou frekvenční doménu, normalizovaná *bandwidth* W nabývá hodnoty nejvýše $\frac{1}{2}$. Již uvedené vyplývá ze skutečnosti, že v normalizované frekvenční doméně je vzorkovací frekvence T_s rovna 1.

V případě, že $k = l$, je výraz $\frac{\sin(2\pi W(k-l))}{\pi(k-l)}$ roven $2W$. Pro *bandwidth* W platí, že bandlimit je pak $2\pi W < \pi$. [4][5]

2.2 Spektrum spojité Slepianovy transformace

Definice 2.2.1 Pro bandlimited signál v časové doméně $f(t)$ platí, že může být vyjádřen pomocí Slepianových sekvencí následovně (inverzní Slepianova transformace):

$$f(t) = \sum_{n=0}^{\infty} a_n \psi_n(c, t) \quad (2.2)$$

Přičemž a_n jsou Slepianovy koeficienty, které lze získat následovně pomocí tzv. dvojí ortogonalit:

$$a_n = \lambda_n(c) \int_{-\infty}^{+\infty} f(t) \psi_n(c, t) dt = \int_{-1}^{+1} f(t) \psi_n(c, t) dt \quad (2.3)$$

Ekvivalentní výpočet Slepianových koeficientů pro diskrétní transformaci je následující:

$$a_n = \sum_{k=1}^N f(k) \psi_n(c, k) \quad (2.4)$$

Koeficienty a_n lze následně využít k vykreslení spektra signálu. [4]

Pozn.: ač je zde spojitá Slepianova transformace uvedena, nebude v práci aplikována, vzhledem k tomu, že je dle vztahu (2.3), jehož zpracování není na běžném počítači téměř možné.

2.3 Funkce „DPSS“ v prostředí Matlab

V prostředí Matlab je k získání Slepianových sekvencí k dispozici funkce `dpss`. Syntaxe ji dovoluje volat více způsoby, z nichž některé umožňují vytvoření vlastní databáze dat pro `dpss` pomocí funkce `dpsssave` a následného užití dat z této databáze, volbu interpolace (lineární či spline – křivková, angl. „linear“, „spline“) a jiné možnosti [6]. V základu je však funkce volána zejména následovně:

```
[dps_seq, lambda] = dpss(seq_length, time_halfbandwidth, Ni)
```

Kde jednotlivé položky znamenají [6]:

- `seq_length` – délku sekvence, v diskrétní oblasti to znamená počet vzorků N . Datový typ – pozitivní integer.
- `time_halfbandwidth` – určuje Time-half-bandwidth product $NW = N \cdot W$, kde N je počet vzorků sekvence a $[-W, W]$ efektivní šířka pásma sekvence. Obvykle se volí např. 2,5. Jedná se o skalární hodnotu, pro kterou musí platit, že $2 \cdot NW < \text{seq_length}$ (což je analogický přepis vztahu $W < 0,5$, je-li normovaná vzorkovací frekvence $\omega_S = 1$, viz také 1.1.3).

- N_i – počet Slepianových sekvencí, které budou na výstupu. Lze zapsat jako přirozené číslo, anebo jako dvousložkový vektor \mathbf{v} , jehož dvě složky jsou přirozená čísla. V tomto případě budou vykresleny sekvence od $\mathbf{v}(1)$ do $\mathbf{v}(2)$. *Pozn.: Matlab používá označení `dps_seq`, number of sequences, viz [6].*
- `dps_seq` – Slepianovy sekvence. Jedná se o matici, jednotlivé sloupce jsou Slepianovy sekvence seřazené od 0. po N_i , počet řádků odpovídá `seq_length` a počet sloupců N_i počtu žádaných sekvencí. Není-li N_i zadáno, je počet sloupců $2 \cdot NW$ (Time-half-bandwidth product).
- `lambda` – udává poměr koncentrace energie ve frekvenční doméně. Sloupcový vektor o délce odpovídající počtu sekvencí N_i .

Pozn.: u spojitých Slepianových sekvencí se namísto parametru NW při generování diskrétních v Matlabu objevuje parametr c . Vztah mezi nimi definuje 2.3.1. [7]

Definice 2.3.1 *Pro ekvivalenci parametru NW v diskrétní doméně a parametru c ve spojitě časové doméně platí následující:*

$$c = \Omega T / 2 \quad (2.5)$$

kde je Ω šířka spektra a T doba pozorování signálu (odpovídající $N \cdot T_S$ v diskrétní doméně). Pro Ω platí následující:

$$\Omega = 2\pi W / T_S \quad (2.6)$$

Z rovnic pak lze odvodit:

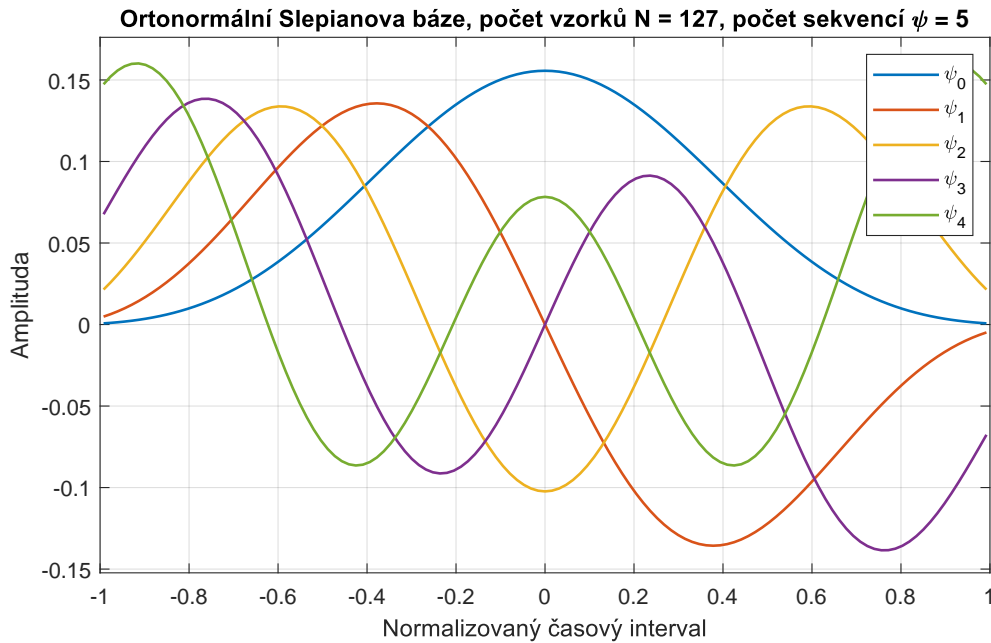
$$c = 2\pi \frac{W}{T_S} \cdot \frac{T}{2} = 2\pi \frac{W}{T_S} \cdot \frac{NT_S}{2} = \pi NW \quad (2.7)$$

Při dosazení vhodných parametrů je snadné generovat bázi *dpss*, tak jako tomu je na obr. 2.1, který ukazuje prvních 5 Slepianových sekvencí ψ_0 až ψ_4 . Je vytvořena s parametry `seq_length = 127`, `time_halfbandwidth = 2,5` a $N_i = 127$. Báze je ortonormální, tzn. ortogonální viz rovnice (1.9) a normalizovaná (1.10).

Vzhledem k tomu, že parametr N_i je shodný s parametrem `seq_length = 127`, jedná se o tzv. úplnou bázi, kdy počtu vzorků odpovídá počet sekvencí.

2.4 Prověření ortogonality Slepianovy báze

Obrázek 2.1 ukazuje Slepianovu ortonormální bázi. Ortonormalitu lze ověřit následovně: z rovnice (1.9) plyne, že suma součinů hodnot dvou různých funkcí ortonormální báze je nulová, z rovnic (1.8) a (1.10) zase vyplývá, že norma funkce je rovna 1 a normu také získáme sumou kvadrátů hodnot dané funkce. Kvadrát lze uvažovat



Obr. 2.1: Slepianova ortonormální báze, prvních 5 sekvencí.

jako součin k -tého prvku se sebou samým, tedy jako skalární součin funkce se sebou sama. Těchto poznatků lze využít při maticovém násobení.

Je-li matice \mathbf{A} násobena maticí transponovanou \mathbf{A}^T , získáme výslednou matici \mathbf{V} , kde hodnota prvku na pozici $[j, k]$ bude rovna **sumě součinů** (skalárnímu součinu) prvků j -tého řádku matice \mathbf{A} s k -tým řádkem matice \mathbf{A}^T . Vzhledem k tomu, že druhá matice je transponovaná, získáváme v matici \mathbf{V} skalární součiny všech kombinací řádků matice \mathbf{A} .

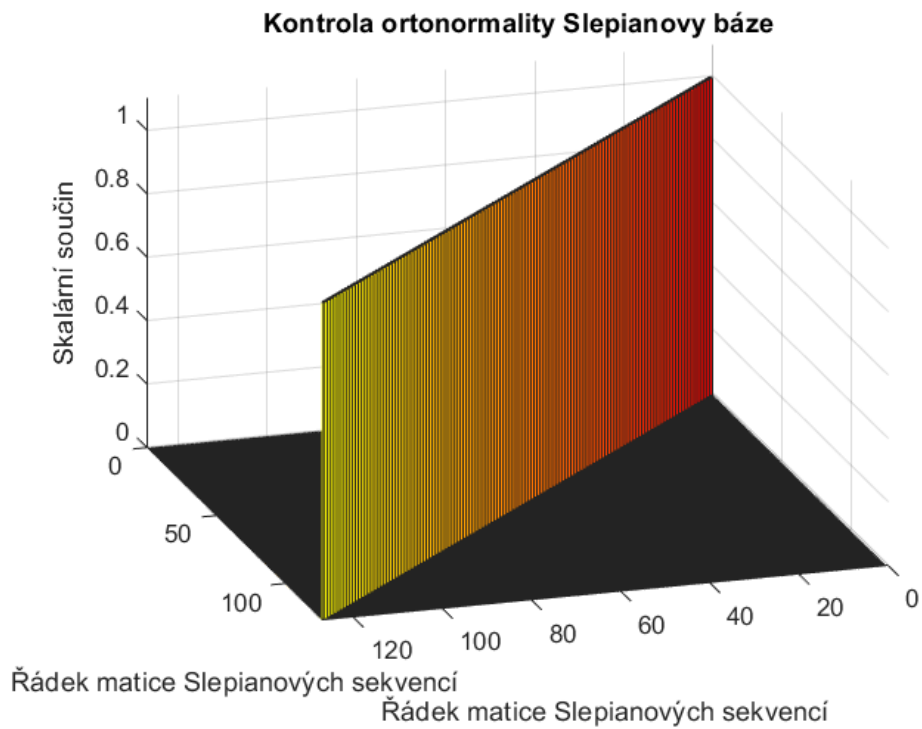
Pokud bude matice \mathbf{A} maticí Slepianových sekvencí, získáme matici \mathbf{V} , která bude jednotkovou maticí $\mathbf{V} = \mathbf{I}$ (všechny hodnoty budou rovny 0, na hlavní diagonále budou rovny 1). Získáme-li jinou než jednotkovou matici, pak pro bázi neplatí definice 1.2.6 a 1.2.7 a báze není ortonormální, potažmo ortogonální. Výše popsané realizuje v Matlabu následující kód. Funkce `bar(check)` vykresluje výslednou matici \mathbf{V} jako trojrozměrnou, ověření, že se opravdu jedná o jednotkovou matici a že je báze ortonormální, ukazuje obr. 2.2. Obrázek 2.3 naproti tomu ukazuje chybu ortonormality $\mathbf{V} - \mathbf{I}$. Vzhledem k tomu, že se jedná o řád 10^{-16} , lze to považovat za chybu datového typu, jelikož Matlab počítá obvykle na 16 platných číslic [8].

Výpis 2.1: Generování Slepianovy ortogonální báze v Matlabu.

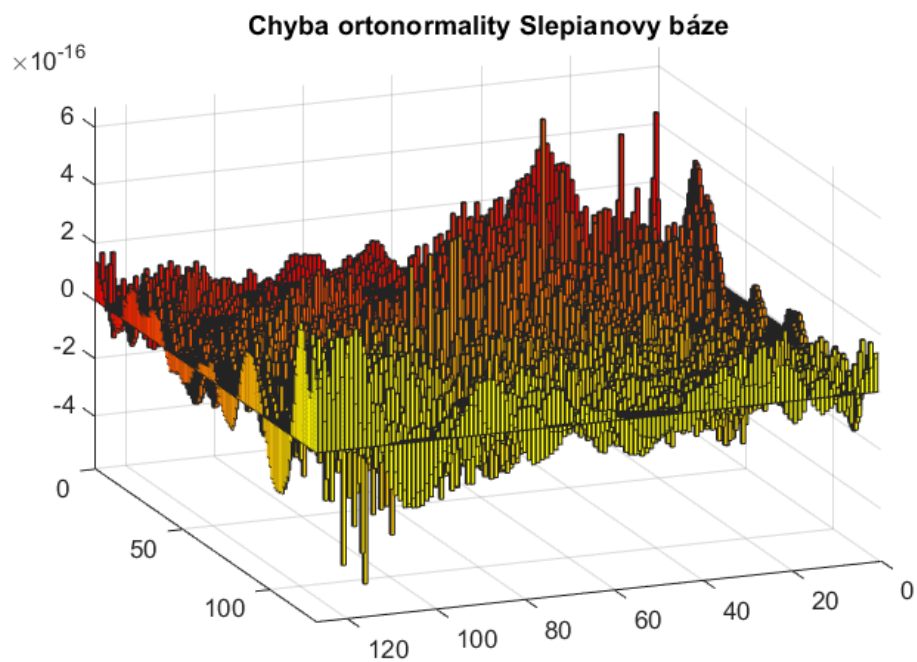
```
num_seq = 127; %počet sekvencí Slepianovy ortogonální báze
N = 127;      %počet vzorků signálu
NW = 2.5;
[dps_seq, lambda] = dpss(N, NW, num_seq);

check = dps_seq*dps_seq';
bar3(check) %vykreslení matice ve 3D do aktuálního okna
min(min(eye(N) == round(check,10)))
```

Pozn.: poslední řádek ve výše uvedeném kódu ověřuje, zda je jednotková matice o rozměru N shodná s výsledkem kontroly ortonormality. Zaokrouhleno (`round`) bylo z důvodu, že Matlab vrací výsledky blížící se 0/1, řádově se od těchto hodnot liší o 10^{-12} a méně (což je potvrzeno na obr. 2.3), zjevně se jedná o nepřesnost výpočtu, proto si můžeme dovolit zaokrouhlit na 10 platných číslic, což je předáno druhým parametrem. Dále dvě funkce `min` hledají nejnižší hodnotu v boolovské matici porovnání. Vyjde `true`, matice jsou shodné (každý prvek v jednotkové matici se shoduje s korespondujícím prvkem v matici pro kontrolu ortonormality).



Obr. 2.2: Důkaz ortonormality Slepianovy ortonormální báze.



Obr. 2.3: Chyba ortonormality Slepianovy báze. Jedná se o výpočet rozdílu matic $V - I$.

3 Slepianova transformace

Při zpracování signálů (signal processingu) je obvykle potřeba signál rozložit na jednotlivé frekvenční složky. To se děje pomocí transformace signálu $f(t)$ z časové oblasti do frekvenční oblasti $F(\omega)$, v níž získáme tzv. spektrální koeficienty. Zpětnou transformací poté získáme ze spektra časový průběh. Obecně lze tyto transformace popsat v diskrétní doméně následovně 3.0.1.

Definice 3.0.1 *Obecná přímá transformace je definována následovně:*

$$a_m = K_P \cdot \sum_{n=0}^{N-1} \overline{\Psi(m, n)} \cdot f(n) \quad (3.1)$$

Její inverze takto:

$$f(n) = K_I \cdot \sum_{m=0}^{M-1} \Psi(m, n) \cdot a_m \quad (3.2)$$

kde a_m jsou spektrální koeficienty, $f(n)$ je transformovaný signál, K_I , K_P jsou obecně konstanty, Ψ je báze matice transformace. Je-li báze ortogonální, pak z definice 1.2.6 vyplývá:

$$f(n) = K_I \sum_{m=0}^{M-1} \Psi \|\Psi\|^{-1} K_P \cdot \sum_{n=0}^{N-1} \overline{\Psi} \|\Psi\|^{-1} \cdot f(n) = \Psi(\Psi, f(n)) \cdot \frac{K_I K_P}{\|\Psi\|^{-2}} \quad (3.3)$$

Z rovnice plyne, že K_I a K_P zajišťují normalizaci spektra a_m a jejich součin je roven: $K_I \cdot K_P = \|\Psi\|^{-2}$. [2]

Jedním z nejběžnějších nástrojů pro výše popsané je Fourierova transformace $\mathcal{F}\{f(t)\}$. Pro diskrétní signály máme Diskrétní Fourierovu transformaci, nicméně pro porovnání se Slepianovou transformací bude v následujících kapitolách uváděna zejména Fast Fourier Transform (FFT).

Fast Fourier Transform (rychlá Fourierova transformace) se od Diskrétní Fourierovy transformace (DFT) liší výpočtem. FFT je účinnější algoritmus pro výpočet DFT. FFT má lineárnímickou ($O = 2N \cdot \log_2(N)$) náročnost, DFT kvadratickou ($O = N^2$) [9]. Z tohoto důvodu nebude v definicích rozlišováno mezi FFT a DFT, jelikož se jejich výsledek neliší.

3.1 Definice DFT

Definice 3.1.1 *Diskrétní Fourierova transformace (DFT) signálu je definována následovně:*

$$a_m = \sum_{k=0}^{N-1} f(k) e^{-jm \frac{2\pi}{N} k} \quad (3.4)$$

kde $m = 0, 1, 2, \dots, N-1$

Definice 3.1.2 *Inverzní diskrétní Fourierova transformace (IDFT) signálu je definována:*

$$f(k) = \frac{1}{N} \sum_{m=0}^{N-1} a_m e^{jm \frac{2\pi}{N} k} \quad (3.5)$$

kde $k = 0, 1, 2, \dots, N-1$

kde a_m je m -tý koeficient DFT

$f(k)$ je k -tý vzorek vzorkovaného signálu

N je celkový počet vzorků. [9]

Vektor koeficientů a_m tvoří spektrum DFT. Slepianova transformace a spektrum je zapsána v definici 2.2.1.

3.2 Slepianova transformace zvoleného signálu

Mějme signál $f(t)$ o předpisu:

$$f(t) = 3 \cdot \sin(2\pi t) + \frac{3}{2} \cdot \sin(6\pi t) + \frac{3}{4} \cdot \sin(10\pi t)$$

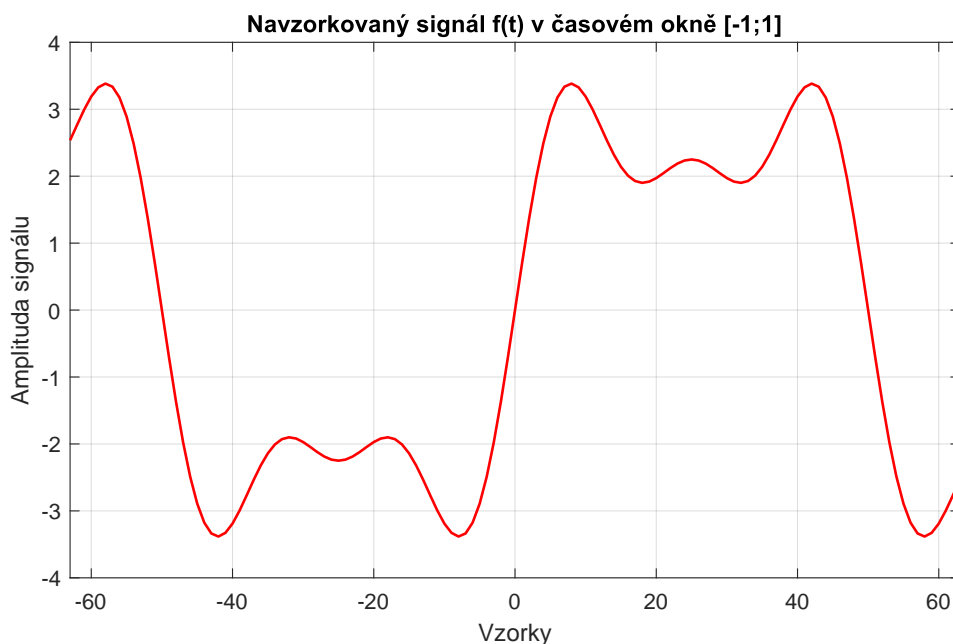
vzorkovaný se vzorkovací periodou $T_S = 0,01s$. Takový signál je frekvenčně omezený (bandlimited) a jeho ukázka v normalizovaném časovém intervalu $\langle -1, 1 \rangle$ je na obr. 3.1. Navzorkovaný signál $f(t)$ zobrazený v normalizovaném časovém intervalu má celkem 127 vzorků, onen časový interval trvá od -63 . do 63 . vzorku.

Nyní vypočteme diskrétní Slepianovu transformaci (ST) signálu za pomoci definice 2.2.1. Získaných koeficientů a_n je tolik, kolik Slepianových sekvencí tvoří bázi (největší báze je úplná báze, kde je počet sekvencí roven počtu vzorků N). Pomocí Slepianovy báze z kapitoly 2.3 lze v Matlabu provést Slepianovu transformaci signálu $f(t)$ takto:

Výpis 3.1: Výpočet koeficientů Slepianova spektra.

```
a = []; %vektor koeficientů a_n
for j = 1:num_seq %projdi celou bázi
    a(j) = temp_var1 * dps_seq(:,j); %výpočet a_j
end
```

`num_seq` zde odpovídá počtu sekvencí v bázi `dps_seq` a výstupem je vektor `a` Slepianových koeficientů (spektrum) viz obr. 3.5. Jak je z obrázku patrné, uplatní se zejména prvních 30 sekvencí z úplného Slepianova spektra. Kód lze zjednodušit na maticové násobení `a = temp_var1 * dps_seq`, odpovídá-li v matici Slepianových sekvencí jeden sloupec jedné sekvenci.



Obr. 3.1: Navzorkovaná část signálu $f(t)$. Jedná se o 127 vzorků.

3.3 Inverzní Slepianova transformace

U inverzních transformací je signál zpětně rekonstruován ze spektra. Definice 2.2.1 určuje, že zpětná rekonstrukce proběhne jako suma všech součinů sekvencí a jim příslušných koeficientů. S výhodou lze opět použít maticové násobení, matici sekvencí je nutno transponovat, aby její sloupec odpovídal vždy vzorku v časové doméně při násobení vektorem spektrálních koeficientů \mathbf{a} :

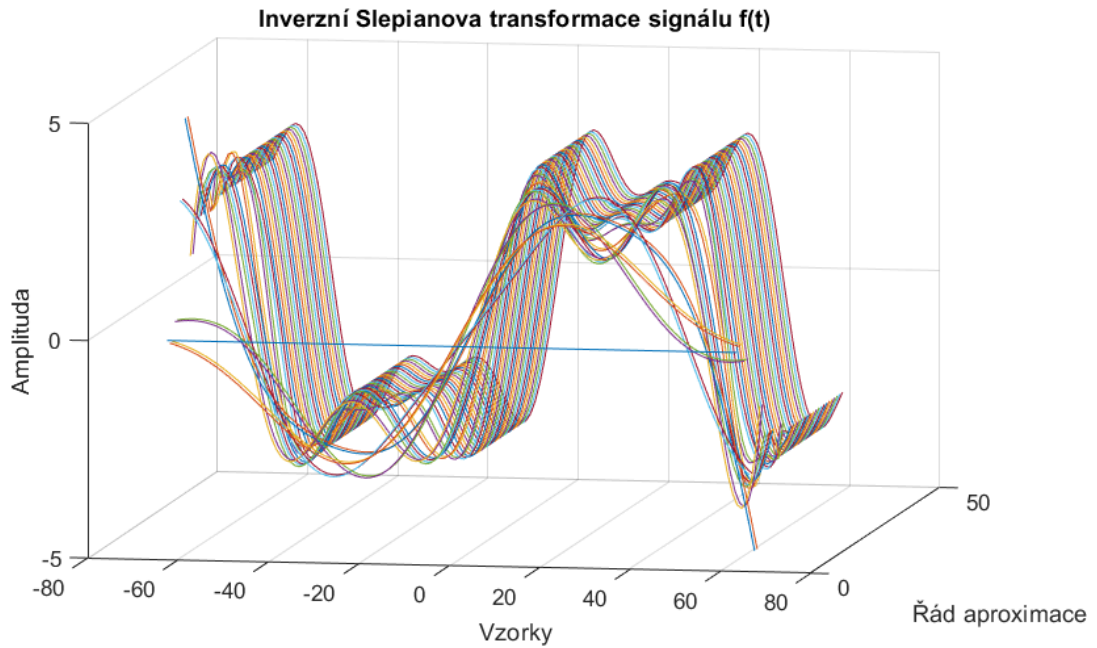
Výpis 3.2: Maticové násobení v Matlabu.

```
signal = a*dps_seq';
```

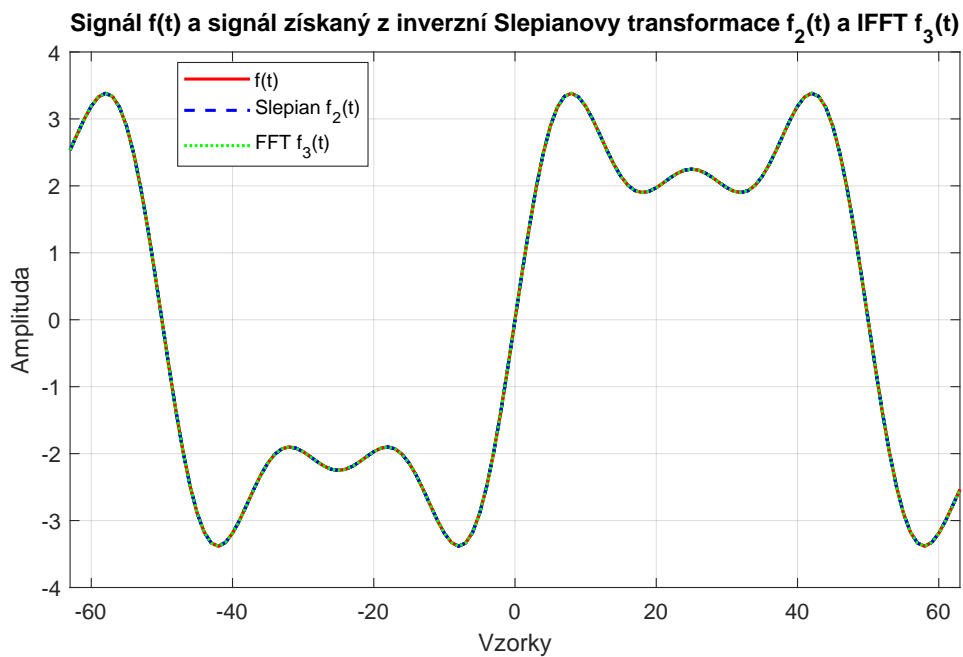
Kde `dps_seq` je transponovaná báze Slepianových sekvencí a \mathbf{a} je matice příslušných koeficientů. Obr. 3.2 ukazuje postupnou inverzní Slepianovu transformaci po jednotlivých iteracích tvořících řád aproximace. Výsledek včetně srovnání s původním signálem a inverzní rychlou Fourierovou transformací (IFFT) je na obr. 3.3.

3.4 Rychlá Fourierova transformace FFT

V programu Matlab lze volat rychlou Fourierovu transformaci pomocí funkce `fft`, její inverzi provede `ifft`. Pomocí volání těchto funkcí lze zjistit Fourierovo spektrum signálu $f(t)$ a zpětně jej rekonstruovat. Takto získané spektrum a rekonstrukce signálu je na obr. 3.6 a 3.3 (včetně porovnání se Slepianovou transformací).

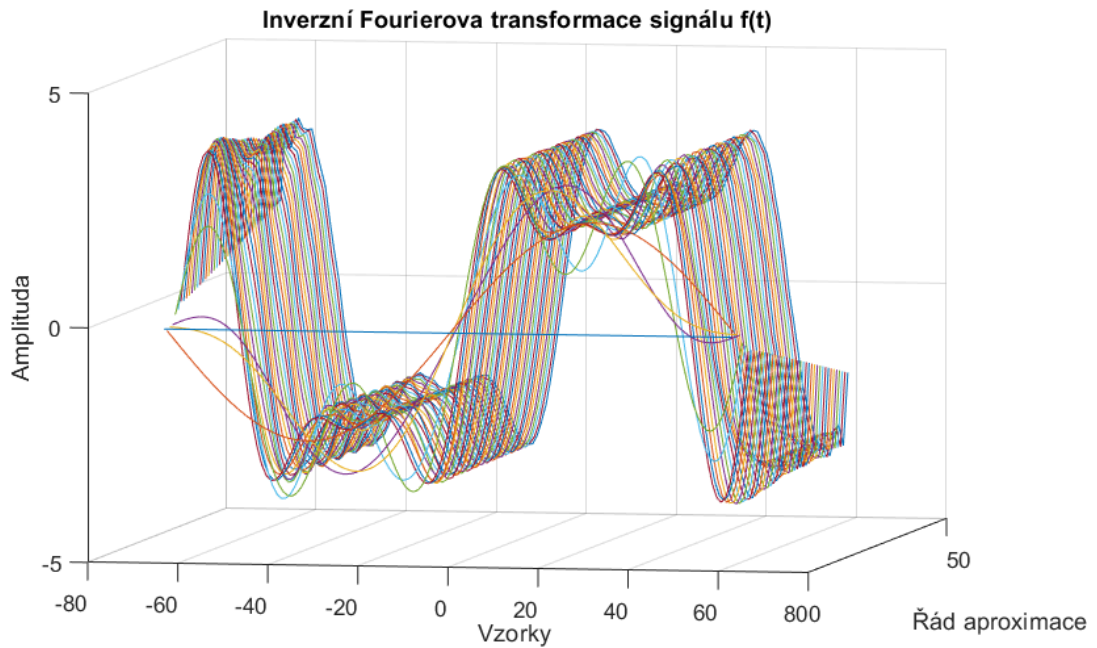


Obr. 3.2: Výsledný signál z inverzní Slepianovy transformace. Každá iterace algoritmu přidává jednu složku spektra vynásobenou příslušným koeficientem a odpovídá příslušnému řádu aproximace. Zobrazeno prvních cca 50 aproximací signálu.

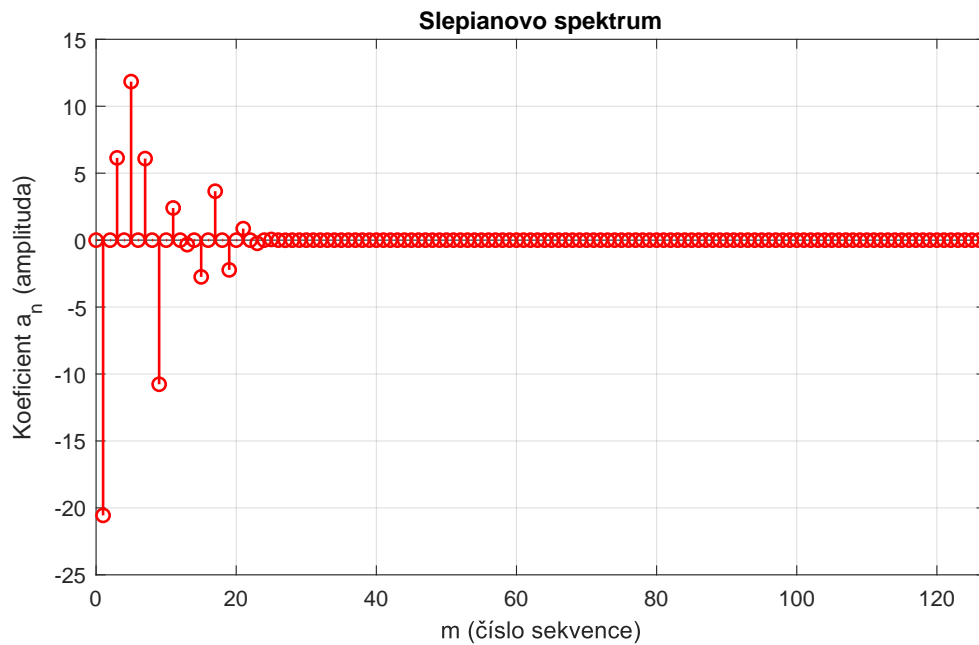


Obr. 3.3: Výsledný signál z inverzní Slepianovy transformace, IFFT a původní signál. U obou transformací byly použity úplné báze/všechny složky spektra.

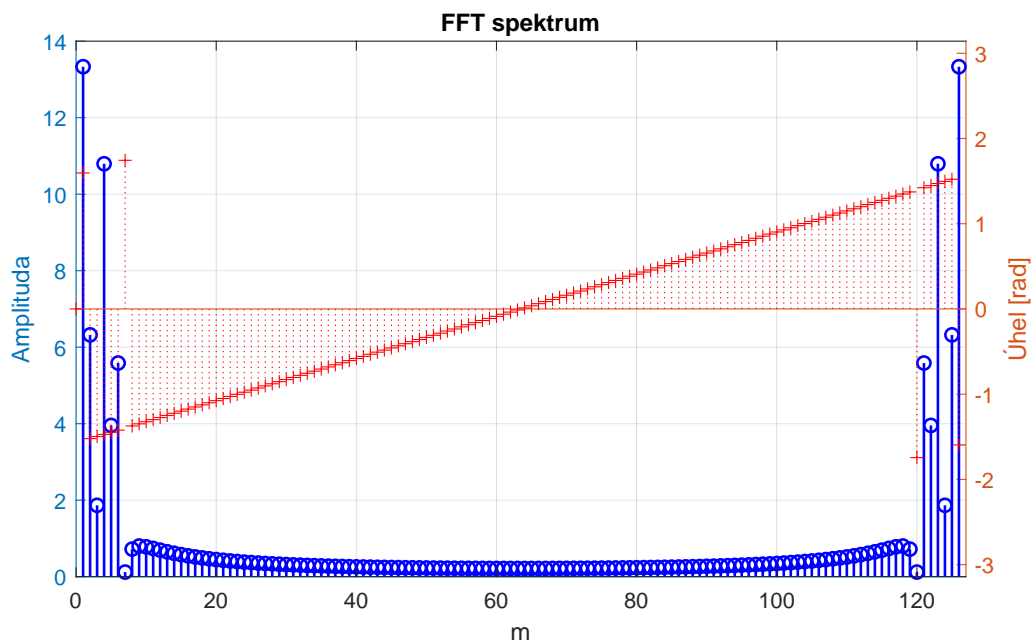
Jak lze na obr. 3.3 pozorovat, jsou-li použity úplné báze a všechny složky spektra, rozdíl není znatelný. Porovnávat obě transformace budeme jinak, například srovnáním aproximačního postupu viz obr. 3.2 pro Slepianovu transformaci a obr. 3.4 pro FFT. Již letným pohledem lze soudit, že daný signál má při n -té aproximaci hladší průběh u Slepianovy transformace, než u FFT (obr. 3.2 a 3.4).



Obr. 3.4: Výsledný signál z inverzní FFT transformace. Každá iterace algoritmu přidává jednu složku spektra – roste řád aproximace. Zobrazeno prvních 42 aproximací signálu.



Obr. 3.5: Výsledkem Slepianovy transformace signálu $f(t)$ je jeho spektrum.



Obr. 3.6: FFT spektrum amplitudové (vertikální osa vlevo) a fázové (osa vpravo).

3.5 Rozdíly ve spektrech

Rozdíl ve spektrech (zohledníme-li symetrii Fourierova spektra) je patrný ve velikosti jednotlivých spektrálních složek. U Slepianova spektra – obr. 3.5 – mají složky blízké 0 větší amplitudu, ale ta se se stoupajícím indexem složky velmi rychle přiblíží k nulovým hodnotám. Nenulové koeficienty mají zde pouze liché složky. U Fourierova spektra na obr. 3.6 mají nenulové koeficienty téměř všechny složky a se stoupajícím indexem amplituda klesá, uvažujeme-li pouze polovinu spektra, druhá polovina je totiž symetrická, přesto je však amplituda stále relativně daleko od 0. Proto i při postupující aproximaci signálu do vyššího řádu (obsahující i vyšší složky symetrického spektra) je stále výrazná změna tvaru signálu. Přesněji se těmito rozdíly bude zabývat také sekce 3.6.

3.6 Srovnání Slepianovy transformace s FFT

Pro porovnání výše uvedených transformací signálu $f(t)$ (také na obr. 3.1) byl zvolen následující postup, kdy probíhá iteračně zpětná transformace a je počítána odchylka:

1. Spočítáno $\varepsilon = 1\%$ peak-to-peak hodnoty signálu, tato hodnota bude využita dále.

$$\varepsilon = 0,01 \cdot (f_{max} - f_{min})$$

2. Aproximace zpětné transformace \hat{f}_m – použita pouze jedna složka spektra (u souměrného FFT spektra souměrně složka napravo a nalevo od středu spektra).
3. Výpočet odchylky hodnot signálu a aktuální aproximace – výsledkem je vektor odchylky \mathbf{v}_O .

$$\mathbf{v}_O = f - \hat{f}_m.$$

4. Celková odchylka E_C – je dána součtem všech prvků ve \mathbf{v}_O , průměrná odchylka E_A (viz obr. 3.7) bodu se získá podílem celkové odchylky a počtu vzorků (bodů).

$$E_C = \sum \mathbf{v}_O$$

$$E_A = \frac{E_C}{N}, \text{ kde } N \text{ je počet vzorků.}$$

5. Jestliže je průměrná odchylka menší než ε , zapsán řád aproximace m .

$$E_A < \varepsilon$$

6. Jestliže je maximální odchylka ve \mathbf{v}_O menší než ε , je zapsán řád aproximace m_{max} .

$$\max \{\mathbf{v}_O\} < \varepsilon$$

7. Vypočtena další aproximace – použita další složka spektra. Výsledek přičten k poslední aproximaci, získána nová aproximace \hat{f}_{m+1} .

8. Jestliže nebyly použity všechny složky spektra, pokračuje se bodem č. 3.

Výše popsaný algoritmus byl realizován v programu Matlab, viz výpis 3.3:

Výpis 3.3: Výpočet odchyly v jednotlivých aproximacích signálu.

```
%Slepianova inverzní transformace, pro IFFT podobně
not_yet = true;
not_yet2 = true;
aprox_celk = zeros(N,1);
dif_vect = [];
for it = 1:N
aprox = a(it)*dps_seq(:,it);      %Iterace inverzní S. t.
aprox_celk = aprox_celk + aprox;%přičti k dalším aprox.
difference = abs(signal'-aprox_celk);%vektor odchyly
%vektor sum odchylek jednotlivých iterací
%dif_vect/N je vektor průměrných odch. bodu v iteracích
dif_vect = [dif_vect sum(difference)];

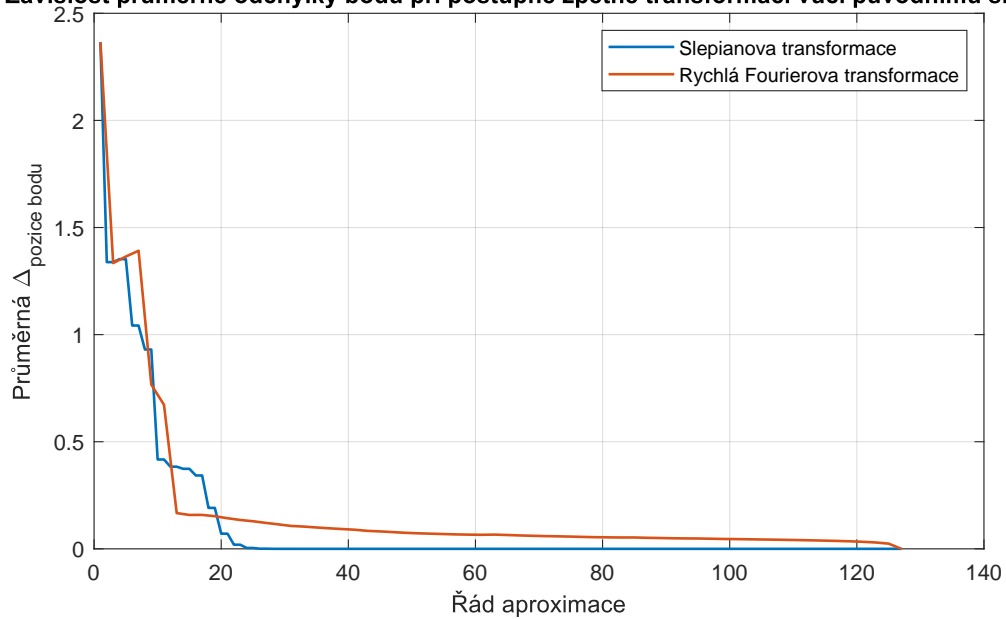
%zapiš iteraci, kdy je max odch. < 1 % peak-to-peak hodn.
if (max(difference)<(Ampl2/100) && not_yet)
    SL_count = it;
    not_yet = false;      %bylo zapsáno, již nepřepisuje
end

%zapiš iteraci, kdy je průměrná odch. < 1 % ...
if (sum(difference)/N<(Ampl2/100) && not_yet2)
    SL_count_avg = it;
    not_yet2 = false;      %bylo zapsáno, již nepřepisuje
end
end
end
```

Hodnotu průměrné odchyly znázorňuje obr. č. 3.7. Což potvrzuje myšlenku uvedenou v sekci 3.5, že Slepianovo spektrum má od určité složky téměř nulové amplitudy, jelikož přidání oněch složek nemá na odchyly znatelný vliv (odchyly je již blízka 0). Z toho lze odvodit, že pro dostatečnou rekonstrukci signálu $f(t)$ z harmonických složek by postačovala *neúplná* Slepianova báze, odhadem by stačilo prvních 30 sekvencí. U FFT je třeba použít všechny složky spektra. Řády aproximace, kdy poklesla průměrná či maximální odchyly pod 1 % peak-to-peak hodnoty signálu, jak bylo uvedeno v algoritmu na počátku sekce 3.6, jsou uvedeny v tabulce 3.1.

Stejně srovnání bylo posléze provedeno pro 100 generovaných harmonických signálů, které byly generovány následovně, viz výpis 3.4:

Závislost průměrné odchylky bodu při postupné zpětné transformaci vůči původnímu signálu



Obr. 3.7: Průběhy odchylky iterační zpětné transformace a původního signálu $f(t)$.

Tab. 3.1: Tabulka porovnání odchylek pro Slepianovu a rychlou Fourierovu transformaci, signál $f(t)$.

Typ odchylky	Řád aproximace Slepian	Řád aproximace FFT
Maximální	22	125
Průměrná	22	57

Výpis 3.4: Generování 100 harmonických signálů.

```

a = rand;
n = 1;
t_vector = t_vector+a*tmp; %náhodné posunutí času
tmp2 = mod(5*rand,5);      %náhodná hodnota
k = 3:2:(2*tmp2+1);       %hodnoty k, počet náhodný
%výpočet 1. hodnoty y_0 signálu
y_0 = amp*sin(2*pi*t_vector);
%výpočet dalších složek - sinusovek - signálu
for i = k                  %náhodný počet cyklů
    amp = amp/(tmp2-0.8);   %změna amplitudy
    y = amp*sin(2*pi*k(n)*t_vector);%výpočet signálu
    y_n = y_n + y;         %přidána ke stávajícímu

```

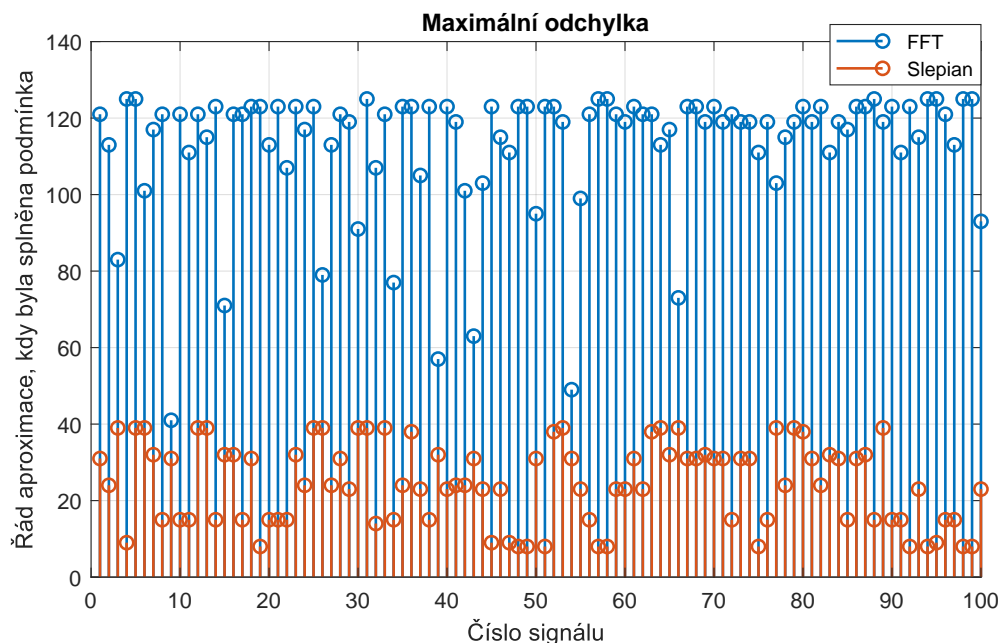
```

    n = n + 1;
end
y = y_0 + y_n;      %přidána první složka
tmp3 = rand;
if tmp3 < 0.5        %náhodně pronásob exp. funkcí
    y_e = 0.9*exp(t_vector);
    y = y.*y_e;
elseif tmp3 < 2/3 && tmp3 >= 1/3
    y_e = 2*exp(-t_vector);
    y = y.*y_e; end % y je generovaná funkce

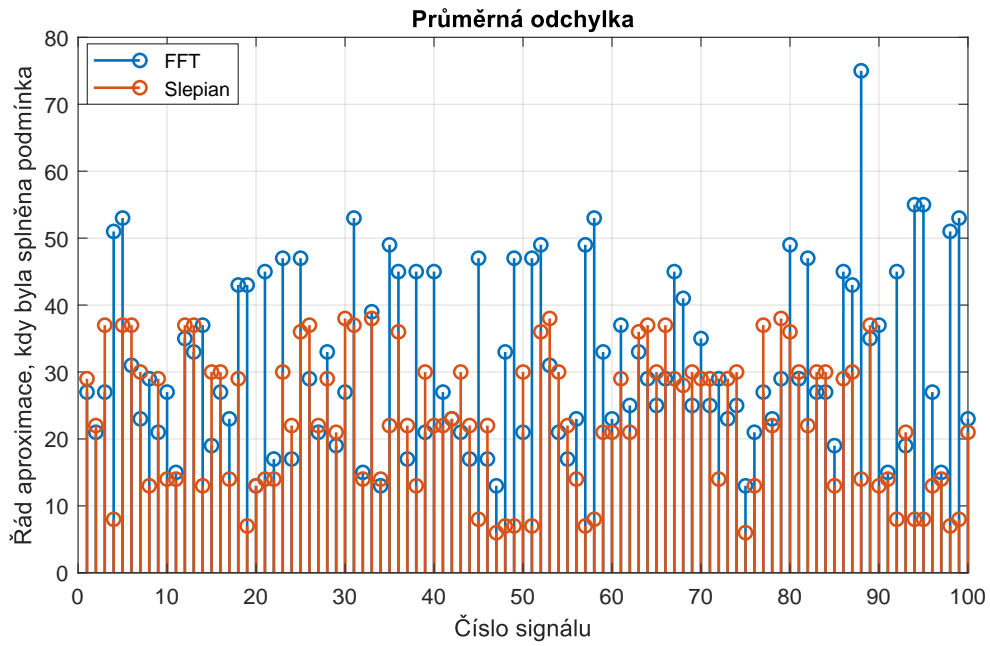
```

Jak lze na výpisu kódu 3.4 vidět, jedná se vzájemně posunuté harmonické funkce skládající se nejvýše z 5 sinusových signálů a některé z nich jsou pronásobeny exponenciálou. Pro tento případ je ekvivalentem tabulky 3.1 obr. 3.9 a 3.8, popř. histogramy 3.11 a 3.12.

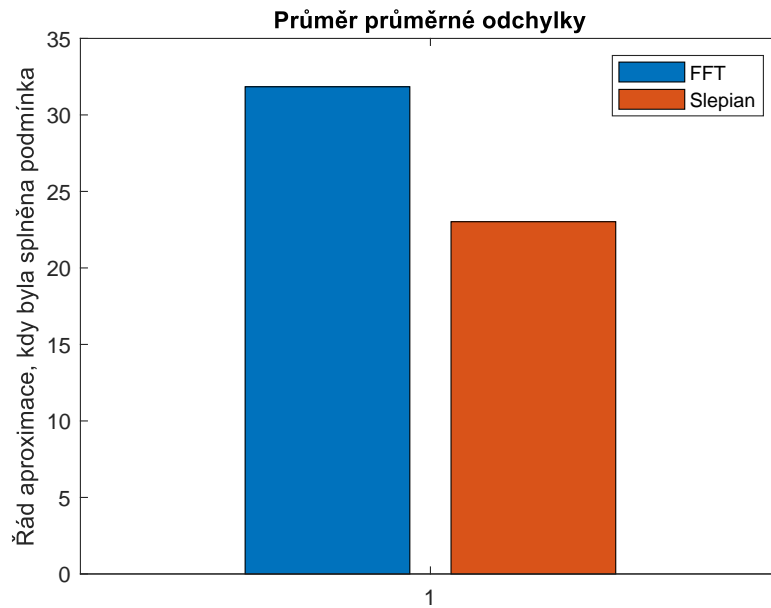
Srovnání se 100 signály potvrzuje, že by zejména u harmonických signálů o jednotkách složek (bude-li složek tolik, kolik je vzorků, je třeba báze úplná v každém případě) postačovala neúplná Slepianova báze, jde-li nám o nejvyšší povolenou odchylku vůči původnímu signálu, kdežto u FFT je třeba ukládat celé spektrum. Pokud by šlo o řád aproximace splnění podmínky pro průměrnou odchylku, rozdíl se maže i pro malý počet harmonických funkcí, z nichž se signál skládá, viz obr. 3.10.



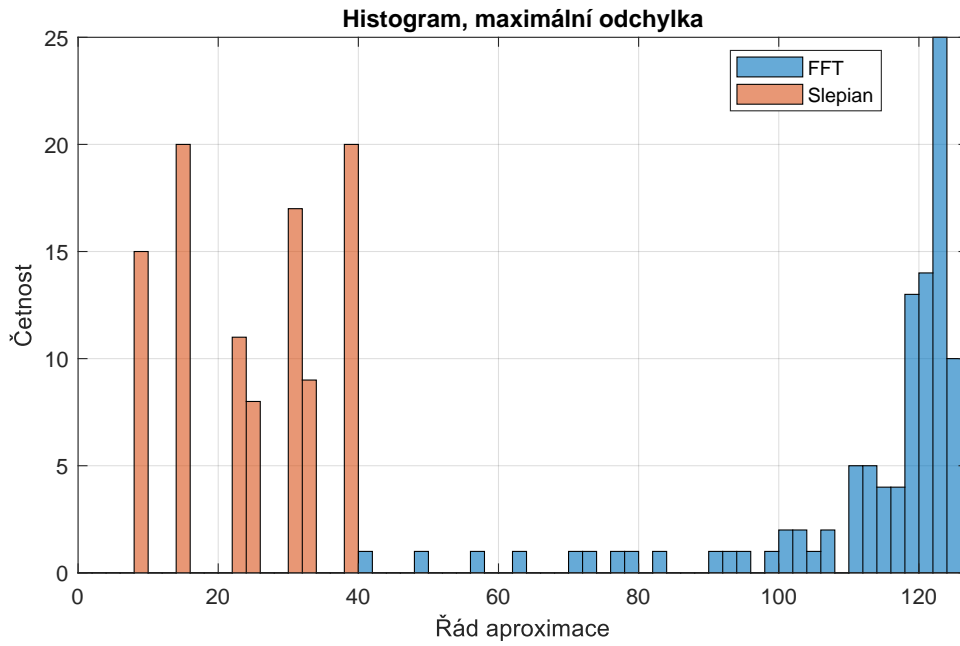
Obr. 3.8: Řád aproximace, kdy pro FFT a Slepianovu bázi maximální odchylka splnila podmínku (vysvětleno na začátku sekce 3.6).



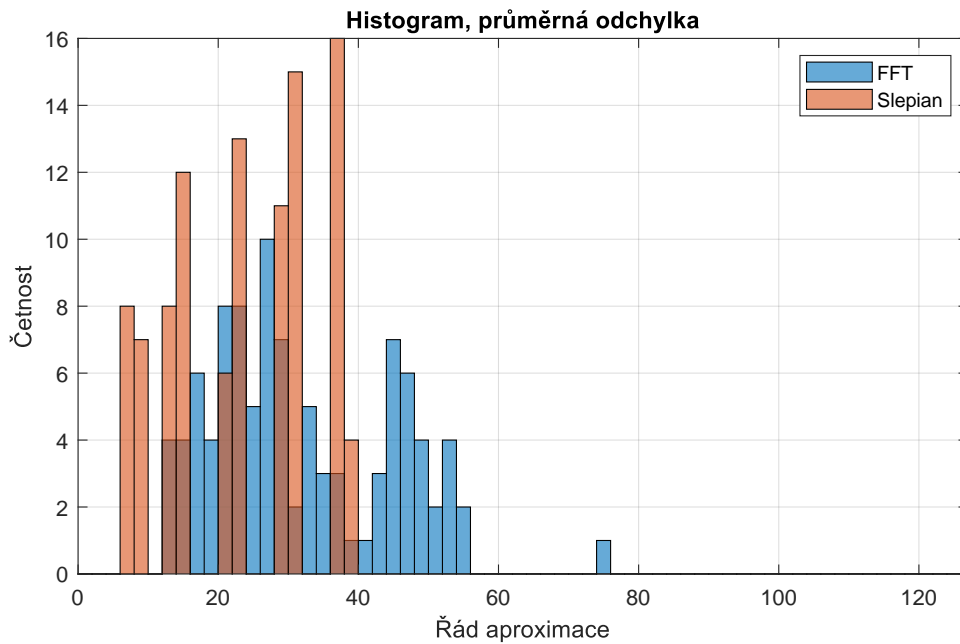
Obr. 3.9: Řád aproximace, kdy pro FFT a Slepianovu bázi průměrná odchylka splnila podmínku (vysvětleno na začátku sekce 3.6).



Obr. 3.10: Průměrný řád aproximace, kdy pro FFT a Slepianovu bázi průměrná odchylka splnila podmínku (vysvětleno na začátku sekce 3.6).



Obr. 3.11: Histogram řádu aproximace, kdy pro FFT a Slepianovu bázi maximální odchylka splnila podmínku (vysvětleno na začátku sekce 3.6).



Obr. 3.12: Histogram řádu aproximace, kdy pro FFT a Slepianovu bázi průměrná odchylka splnila podmínku (vysvětleno na začátku sekce 3.6).

4 Využití „DPSS“ při extrapolaci signálu

4.1 Definice extrapolace

Definice 4.1.1 Máme časově omezeně pozorovaný signál $x(t) \in B_\Omega$, kde B_Ω je podprostor Hilbertova prostoru ℓ_2 ortogonálních a zároveň bandlimited funkcí. Signál $x(t)$ pozorujeme v intervalu Λ , pak pod pojmem extrapolace rozumíme hledání chování signálu mimo interval pozorování Λ , resp. v jeho okolí (sousedních časových hodnotách). [10]

Pojem extrapolace ve zkratce tedy znamená případ, kdy jsou naměřená data odhadována i mimo interval pozorování. Slepianovy sekvence jsou jedním z nejlepších nástrojů pro tento účel, je-li signál rovnoměrně vzorkován (jiné způsoby než pomocí Slepianovy báze uvádí např. [10]). Důležitou roli během extrapolace hrají jednak λ_n (vlastní čísla nabývající sestupných kladných hodnot, která vyjadřují koncentraci sekvencí, viz 2.1.1 a jejich hodnoty viz 4.3) a také parametr c (viz 2.3 pro jeho definici a ekvivalenci s parametrem NW v prostředí Matlab), který musí být ovšem vhodně zvolen, viz definice 4.1.2.

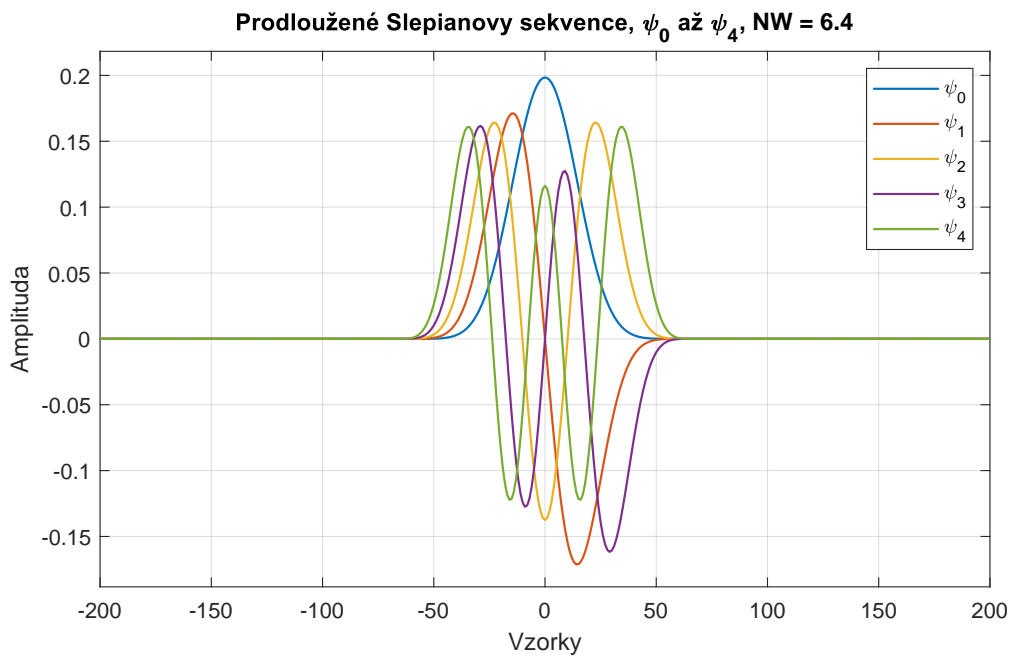
Definice 4.1.2 Směřuje-li parametr c k π/T_S , pak je extrapolace nemožná a výsledky Slepianovy transformace jsou podobné Fourierově transformaci. Ideálně se volí c co nejbližší k bandlimitu Ω , což je v praxi opět ovlivněno vzorkovacím teorémem, potažmo vzorkovací frekvencí. [4]

4.2 Použitý algoritmus pro extrapolaci

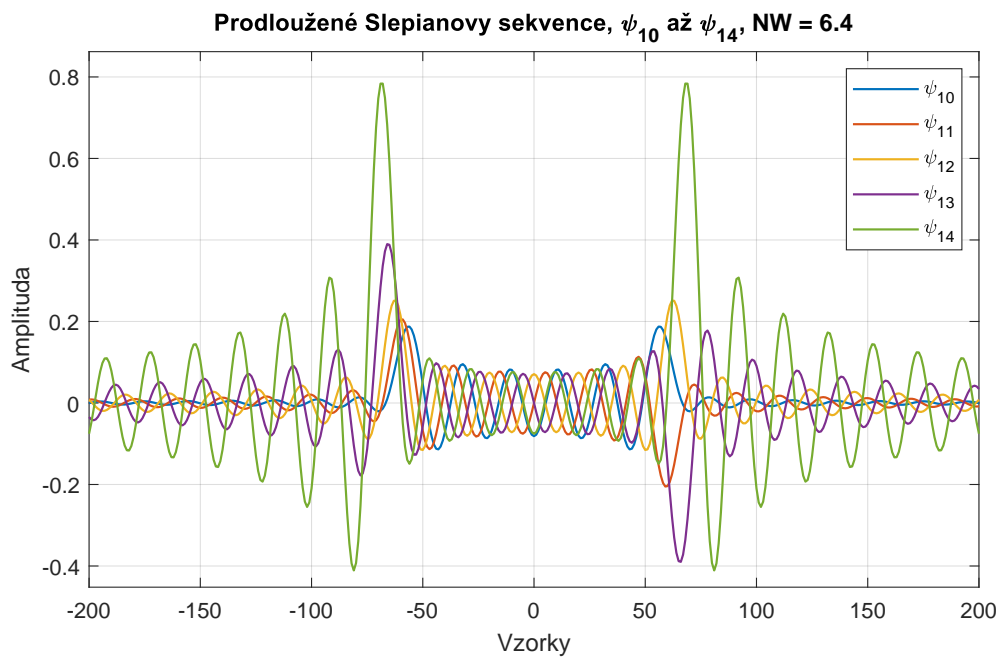
Návrh vychází z definičního konvolučního vztahu viz 2.1.1. Ten je upraven do tvaru:

$$\psi_n(K, W, l) = \frac{\sum_{l=1}^K \frac{\sin(2\pi W(k-l))}{\pi(k-l)} \psi_n(K, W, l)}{\lambda_n(K, W)} \quad (4.1)$$

Kde k jde od 1 až po K . Přičemž nyní je možné prodloužit Slepianovy sekvence ze 127 naměřených vzorků kupříkladu na 401 nebo jiný počet vzorků. Zde již nebude platit, že prodloužená plná báze má stejný počet sekvencí, jako je počet vzorků prodloužené sekvence. Počet sekvencí bude opět nejvýše 127 při úplné bázi. Generovat úplnou bázi zde ovšem nebude možné, problém bude popsán dále.

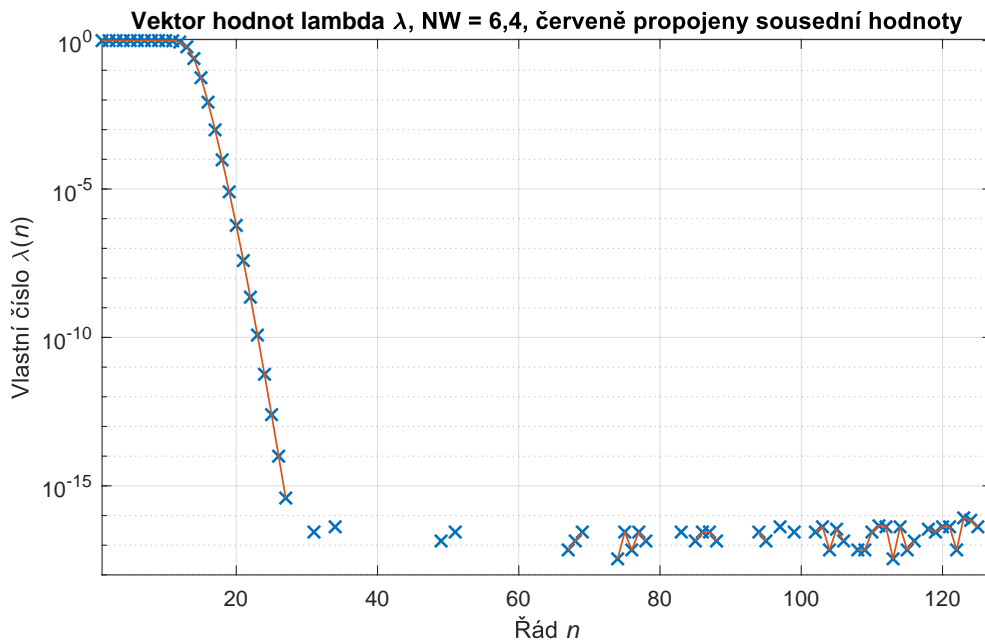


Obr. 4.1: Prodloužená Slepianova báze na 401 vzorků symetricky kolem původních 127 vzorků. ψ_0 až ψ_4



Obr. 4.2: Prodloužená Slepianova báze na 401 vzorků symetricky kolem původních 127 vzorků. ψ_{10} až ψ_{14}

Jak lze vidět na obr. 4.1 a 4.2, ne všechny sekvence zasahují významně i mimo interval pozorování signálu.



Obr. 4.3: Ukázka hodnot λ při parametru $NW = 6,4$. Je patrné, že některé hodnoty nebyly vypočteny (v Matlabu jsou nulové).

Postup: Nejprve je třeba zohlednit následující: jelikož se v rovnici 4.1 dělí lambdou, je dobré zkontrolovat její hodnoty, zda nejsou nulové. Lambda nabývá sestupných hodnot od 1 po hodnoty blízké nule – a vzhledem k přesnosti Matlabu [8] je možné, že některé z hodnot budou kvůli nepřesnosti rovny 0 (tyto nejsou v obrázku 4.3 vyobrazeny, logaritmické měřítko nuly nevykreslí). Proto budeme počítat nové ψ_M pouze pro M takové, že jemu odpovídající lambda je nenulová. Zjednodušeně pak budeme mít pouze prvních M sekvencí, kde $M \leq N$, přičemž původních sekvencí je N . Takové N lze nalézt kupříkladu následovně:

Výpis 4.1: Určení nejvyšší nenulové λ .

```

nonzero = find(lambda); %vypíše indexy nenulových prvků
set_1 = 1:N;           %vektor 1 až 127

% rozdíl množin nonzero a set_1
% nalezne nejmenší index, kde je nulová lambda a odečte 1
% => je získán index poslední nenulové lambda
max_rad = min(setdiff(set_1,nonzero)) - 1;

```

Máme-li stejný testovací signál $f(t) = 3 \cdot \sin(2\pi t) + \frac{3}{2} \cdot \sin(6\pi t) + \frac{3}{4} \cdot \sin(10\pi t)$ a parametr $NW = 6.4$, je M rovno 27. Následně je již možné vypočítat prodloužené Slepianovy sekvence (v konkrétním případě prvních 27) podle rovnice 4.1.

Výpis 4.2: Výpočet prodloužených ψ .

```
for seq_num = 1:max_rad
    for l = -floor(N/2):floor(N/2)
        dps_dl(seq_num, :) = dps_dl(seq_num, :) + ...
        2*c/N*sinc(2*c/N*(k-l))*dps_seq(1+floor(N/2)+1, seq_num);
    end
    dps_dl(seq_num, :) = dps_dl(seq_num, :)/lambda(seq_num);
end
```

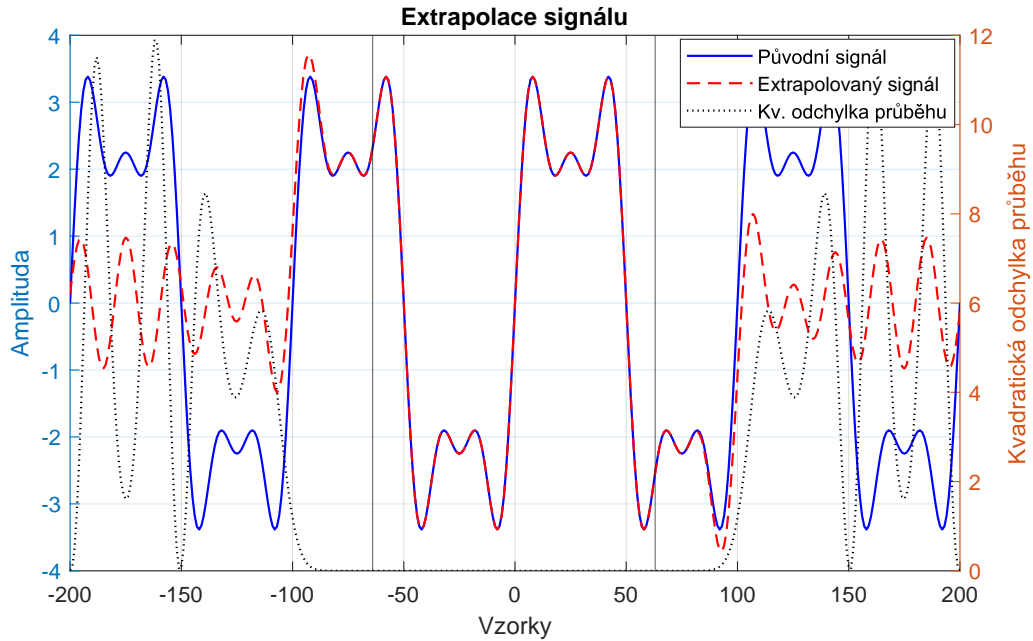
A poté je možné provést inverzní Slepianovu transformaci obdobně jako v sekci 3.3 buď *for cyklem*, anebo příslušně upraveným maticovým násobením – je nutné omezit nejvyšší řád koeficientů spektra. Výsledek je poté na obr. 4.4 včetně kvadratické odchylky signálu.

Výpis 4.3: Inverzní Slepianova transformace prodloužených ψ .

```
for iteration = 1:max_rad
    signal_element = a(iteration)*vysl(iteration, :);
    signal = signal + signal_element;
end
```

```
%varianta s maticovým násobením
signal = a(1:max_rad)*vysl;
```

Je dobré si také povšimnout, že pro extrapolaci mají význam zejména klesající λ od 10. po 27., viz spojitý pokles na obr. 4.3 a také srovnání 4.1 a 4.2.



Obr. 4.4: Extrapolace signálu $f(t)$, $NW = 6,4$. Svislými čarami je naznačen interval měřeného periodického signálu. Extrapolace je velmi přesná pro cca dalších 16 vzorků (samples).

4.3 Nedodržení parametrů

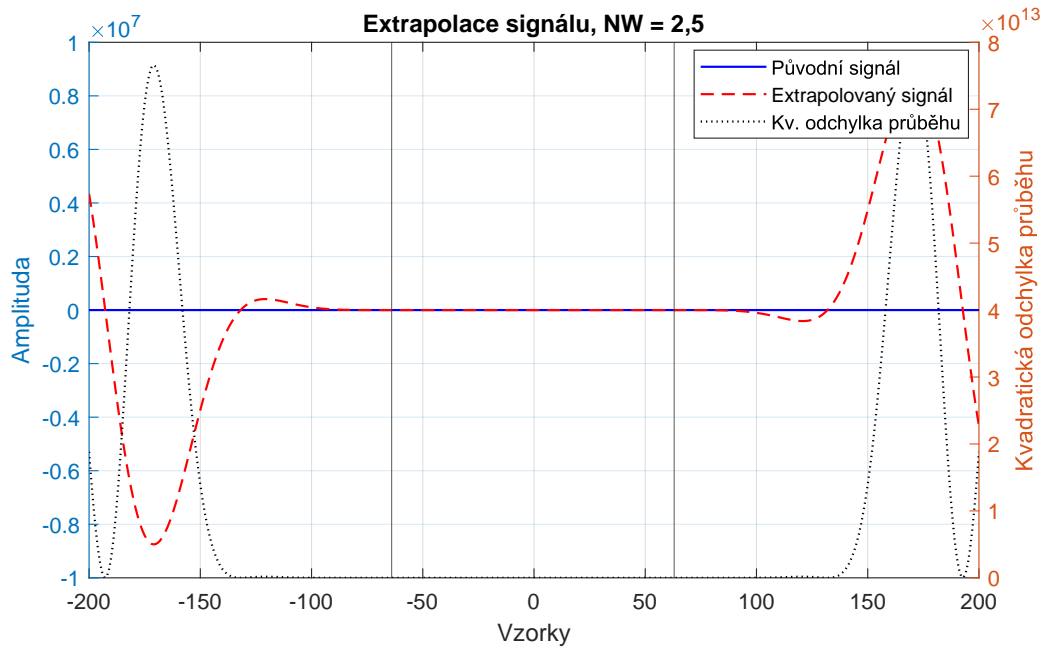
Jak již bylo uvedeno na začátku této kapitoly o extrapolaci, je nutno dodržet určité podmínky, aby byla extrapolace možná.

Správná hodnota parametru NW Jak je odvozeno v definici 2.3.1, parametr c lze převést na NW užívané v Matlabu následovně: $c = \pi NW$. Aby vyšla extrapolace co nejlépe, musí být NW zvoleno co nejbližší k Ω/π (odvozeno z 4.1.2) [4]. Dosazením do rovnice 1.4 lze zjistit hodnotu W :

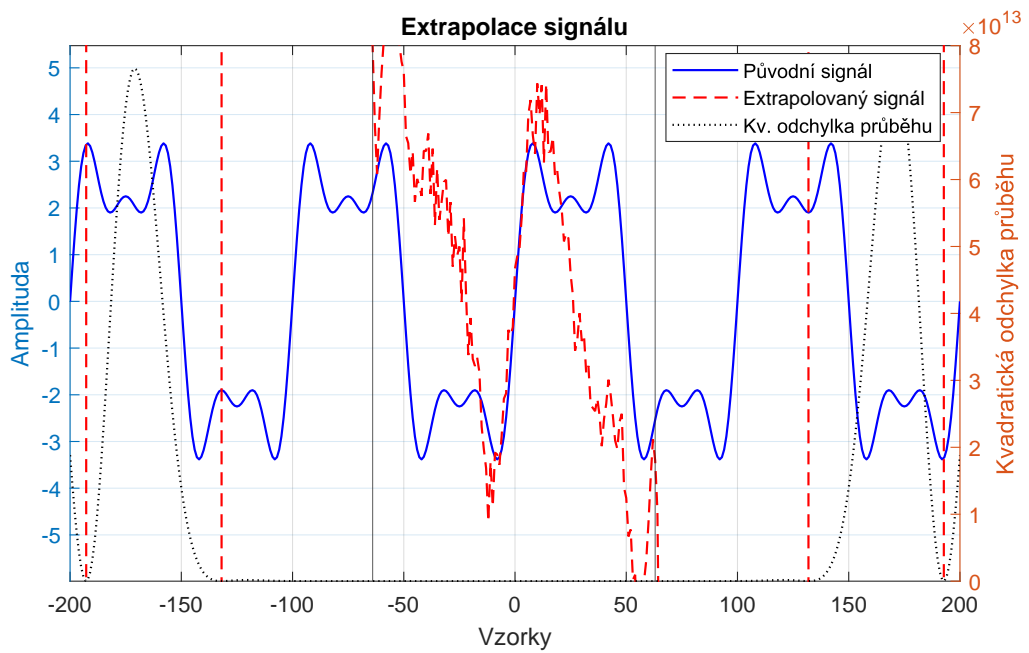
$$W = \frac{\Omega}{2\pi f_S} = \frac{\Omega T_S}{2\pi} = \frac{10 \cdot \pi \cdot 0,01}{2\pi} = 0,05 \quad (4.2)$$

Do 4.2 za Ω lze dosadit nejvyšší frekvenci signálu $\Omega = 10\pi$, vzhledem k tomu, že byl použit stejný testovací signál $f(t)$, činí vzorkovací perioda $T_S = 0,01s$, a pak je možné vyjádřit si vztah pro parametr W . Poté již lze jednoduše dopočítat NW prostým vynásobením počtem vzorků $N = 127$: $NW = 127 \cdot 0,05 = 6,35$. Proto je pro extrapolaci v této kapitole zvolen parametr $NW = 6,4$.

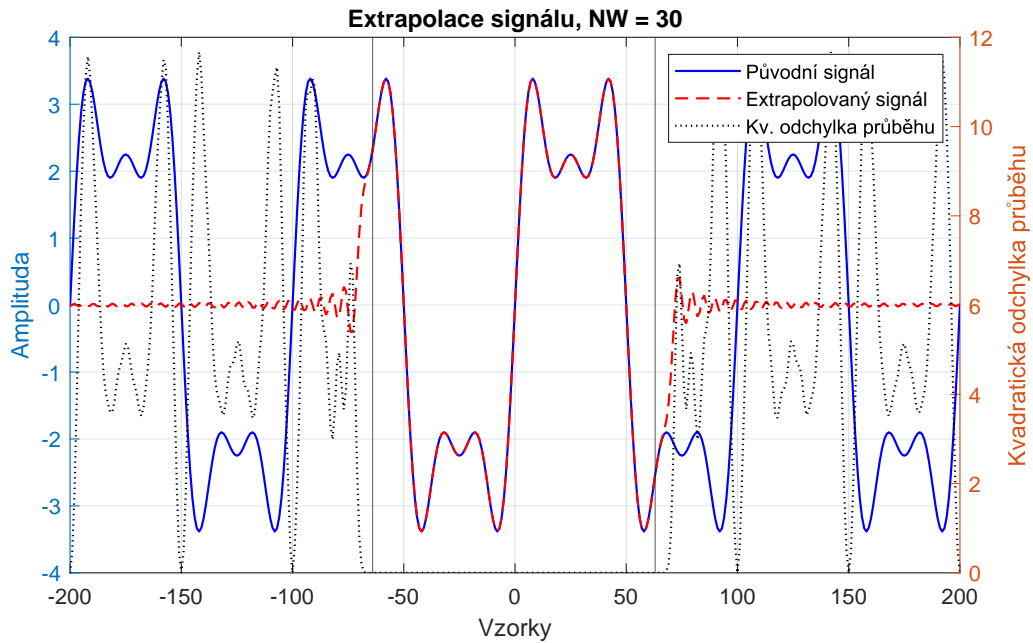
Není-li parametr NW stanoven správně, dojde k nesprávným výsledkům, které ilustrují obrázky 4.5, 4.6 a 4.7.



Obr. 4.5: Extrapolace signálu $f(t)$, $NW = 2,5$. Extrapolovaný signál dosahuje hodnot o několik řádů vyšších.

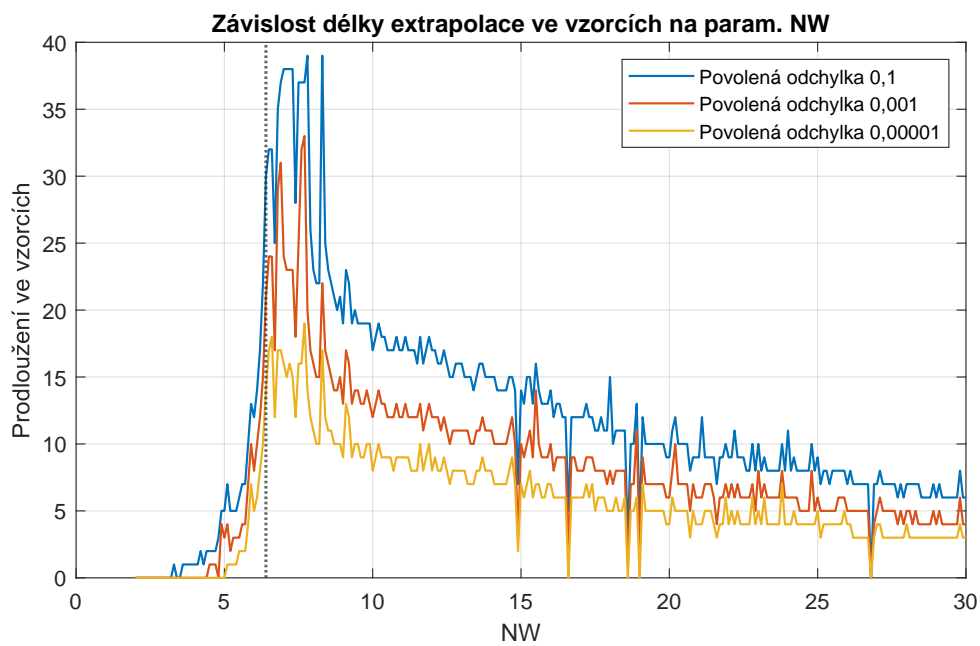


Obr. 4.6: Extrapolace signálu $f(t)$, $NW = 2,5$, detail. Výsledek nelze považovat za rekonstrukci signálu $f(t)$.



Obr. 4.7: Extrapolace signálu $f(t)$, $NW = 30$. V porovnání s obr. 4.4 však extrapolace není tak kvalitní.

Obecně pak lze stanovit závislost délky extrapolace při tolerovatelné kvadratické odchylce vůči předpokládanému periodickému signálu na parametru NW , jak je ilustrováno na obr. 4.8. Graf byl vytvořen tak, že signál byl extrapolován pomocí bází s NW jdoucím od 2 do 30 s krokem 0,1, poté byla extrapolace porovnávána s předpokládaným průběhem a počítána odchylka. Zapsán byl vždy počet vzorků vně intervalu, který měl kvadratickou odchylku od signálu menší, než byla stanovená (0,1; 0,001 nebo 0,000 01). V závislosti na obr. 4.8 si lze všimnout poklesů (zejména mezi $NW = 15$ a $NW = 20$) až k nulovému prodloužení, mohlo by se jednat např. o výpočetní chybu Matlabu, tj. nekvalitně generovanou bázi při daném NW , čemuž nasvědčují i warningy vypisované v Matlabu při generování Slepianovy báze ve znění: `'Matrix is singular to working precision. Results may be inaccurate.'`, které se vždy vyskytují v místech poklesů.



Obr. 4.8: Závislost délky extrapolace při definované kvadratické odchylce na parametru NW . Tečkovaně naznačena používaná hodnota $NW = 6,4$.

5 Využití „DPSS“ při kompresi signálu ze simulátoru řízení vozidla

V kapitole 3.6 se srovnává Slepianova ortogonální báze s Fourierovou. Na první pohled se zdá, že Slepianova aproximuje daný signál lépe, než Fourierova, nicméně toto srovnání bylo provedeno pro harmonické signály o několika složkách, což má také svá úskalí:

- Při čistě naměřené periodě signálu by Fourierova transformace aproximovala lépe, báze se totiž skládá z harmonických funkcí o násobcích základní periody. Toto je ovšem kompenzováno skutečností, že téměř nikdy netrvá naměřený signál právě 1 periodu či celočíselný počet period.
- Skutečné měřené signály se ideálním harmonickým průběhům mohou pouze přibližovat.

Proto je v této kapitole ověřeno využití diskrétní Slepianovy báze a transformace na skutečný signál poskytnutý ze simulátoru řízení vozidla (vč. nahrávacího m. file `load_data.m`) a kompresi dat naměřeného signálu (principem komprese spektra). Transformace pro porovnání se Slepianovou jsou Fourierova a kosinová (všechny diskrétní).

5.1 Diskrétní kosinová transformace

V historii digital signal processingu byly důležité milníky ty, které vedly k lepším algoritmům výpočtů diskrétní Fourierovy transformace (např. algoritmus FFT viz [9]) anebo objevy jiných výpočetně efektivních transformací. Obdobným objevem jako objevem algoritmu FFT je i **diskrétní kosinová transformace** (zkr. DCT) užívaná hojně ve zpracování jedno- (zvukový signál, zde lze najít i velmi specializovaná využití jako např. v algoritmu pro detekce šířky pásma řeči viz např. [11]) a vícerozměrných signálů (obrazy). Diskrétní kosinová transformace tedy bude také porovnáována se Slepianovou a Fourierovou. [12]

Definice 5.1.1 *Diskrétní kosinová transformace signálu má 4 formy, jež jsou definovány maticemi báзовých funkcí $[C_{N+1}^K]_{mn}$ [12]:*

$$[C_{N+1}^I]_{mn} = \left(\frac{2}{N-1}\right)^{\frac{1}{2}} \left[k_m k_n \cos\left(\frac{mn\pi}{N-1}\right) \right] \quad (5.1)$$

kde $m, n = 0, 1, \dots, N-1$

$$[C_{N+1}^{II}]_{mn} = \left(\frac{2}{N}\right)^{\frac{1}{2}} \left[k_m \cos\left(\frac{m\left(n + \frac{1}{2}\right)\pi}{N}\right) \right] \quad (5.2)$$

kde $m, n = 0, 1, \dots, N-1$

$$[C_{N+1}^{III}]_{mn} = \left(\frac{2}{N}\right)^{\frac{1}{2}} \left[k_n \cos \left(\frac{\left(m + \frac{1}{2}\right) n \pi}{N} \right) \right] \quad (5.3)$$

kde $m, n = 0, 1, \dots, N-1$

$$[C_{N+1}^{IV}]_{mn} = \left(\frac{2}{N}\right)^{\frac{1}{2}} \left[\cos \left(\frac{\left(m + \frac{1}{2}\right) \left(n + \frac{1}{2}\right) \pi}{N} \right) \right] \quad (5.4)$$

kde $m, n = 0, 1, \dots, N-1$

$k_p = 1/\sqrt{2}$ když $p = 0$ nebo $p = N$, jinak $k_p = 1$, kde $k_p = k_n, k_m$.

N je obecně délka signálu.

Kosinová transformace je pak definována takto:

$$a_m = \sum_{n=0}^{N-1} [C_{N+1}^K]_{mn} \cdot f(n) \quad (5.5)$$

Její inverze takto:

$$f(n) = \sum_{m=0}^{M-1} [C_{N+1}^K]_{mn} \cdot a_m \quad (5.6)$$

kde a_m jsou spektrální koeficienty a $f(n)$ je transformovaný signál.

Všechny transformace jsou ortogonální (viz def. 1.2.5). Inverzi lze získat prohozením m a n , pak I. a IV. jsou inverzí samy sobě a II. a III. jsou si inverzemi navzájem. Matlab používá ve své knihovní funkci `dct DCT II.`, není-li definováno jinak. Tato funkce je použita dále v práci. Obecně u kosinové transformace platí, že již několik spektrálních složek postačuje k dostatečné rekonstrukci původního signálu. [13]

5.2 Simulátor řízení vozidla

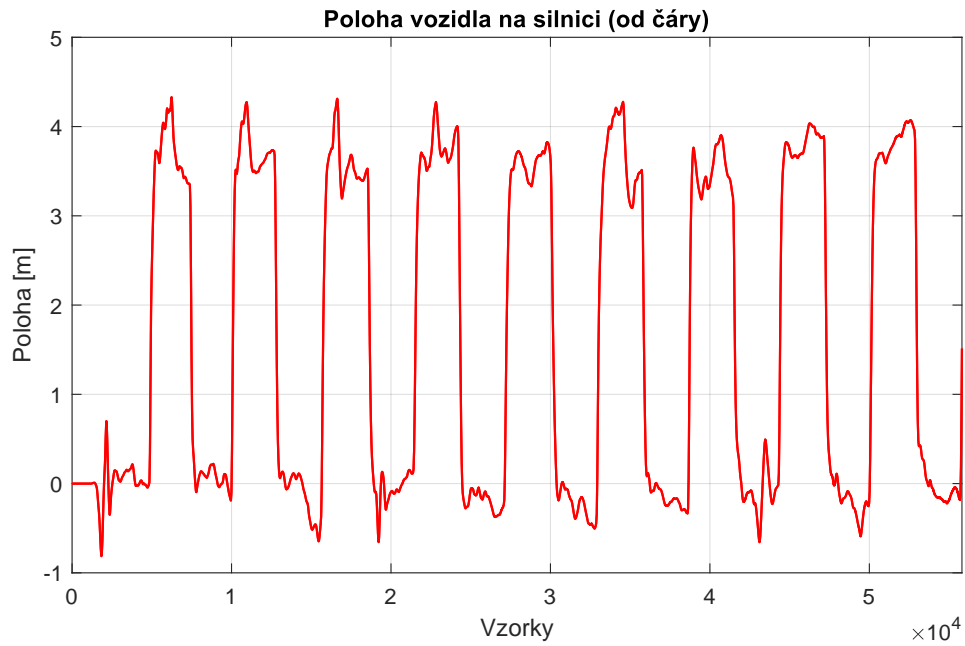
Jedná se o projekt k diplomové práci studenta Davida Michalíka [14] umožňující získávání dat o chování řidiče vozidla. V rámci projektu slouží k získání dat o chování řidiče pedály, volant, řadicí páka, případně jiná obvyklá zařízení připojená k PC jako klávesnice či myš viz obr. 5.1. Samotný Simulátor řízení vozidla navržený na herní platformě Unreal Engine vytváří situace z reálného provozu silničního vozidla, jako např. vběhnutí srny do cesty anebo vynucenou změnu jízdního pruhu na dálnici. Následně jsou z něj ukládána data, například natočení volantu či poloha vozidla na šířce silnice (nejedná se o ujetou vzdálenost, ale o vzdálenost od kraje cesty, popř. ve kterém jízdním pruhu se vozidlo nachází). [14]



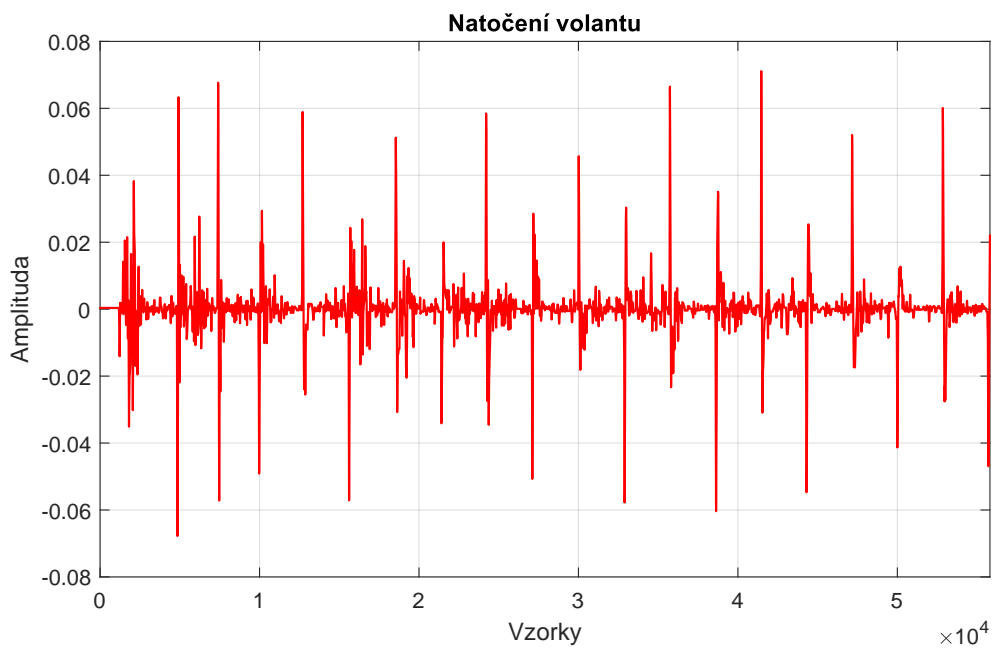
Obr. 5.1: Realizace simulátoru řízení vozidla, zdroj obrázku: [15].

5.3 Komprese naměřených dat

Ze simulátoru byla pro účely komprese dat vybrána následující data viz obr. 5.2 a 5.3. Jedná se o polohu vozidla vzhledem k šířce silnice (umístění v jízdním pruhu) a natočení volantu během jízdy. Obrázky neuvádějí všechna data, ale jen reprezentativní vzorek. V práci byla použita další naměřená data obdobného charakteru.



Obr. 5.2: První signál je poloha vozidla na silnici vzhledem k její šířce.



Obr. 5.3: Druhý signál symbolizuje natočení volantu během řízení, matematicky vzato se jedná o derivaci signálu na obr. 5.2.

Celkově mají data zpracovávaná dále kolem 210 000 vzorků. S mírnou modifikací jako v kapitole 3.6 probíhá srovnání transformací. Signál je vzorkován po úsecích,

každý z nich má 127 vzorků. Poté jsou spočítány transformace – FFT, DCT a ST. Následuje zpětná aproximační transformace signálu (se zvyšujícím se řádem aproximace signálu). Nejprve jsou brány koeficienty nižšího řádu, následně řádů vyšších a je porovnáván signál s původním průběhem pomocí tzv. MSE (mean square error) kritéria – viz definice 5.3.1. Toto je průměrováno pro všechny 127 vzorků dlouhé úseky a na závěr jsou transformace vyhodnoceny co se přesnosti aproximací (alias tzv. řídké reprezentace dat, angl. sparse representation) týče. Přesněji je algoritmus popsán slovně příkazy pro program v následujícím seznamu:

1. Vezmi první spektrální koeficient pro danou transformaci, ostatní inicializuj na nulové hodnoty. Proveď to pro všechny transformace.
2. Vykonej inverzní transformace (IST, IFFT, IDCT) pro délku signálu rovnou 127 a dané spektrální koeficienty.
3. Porovnej výsledky inverzních transformací s originálním signálem (odpovídajícím úsekem signálu), vypočítej MSE pro každou transformaci a ulož je.
4. Přidej další spektrální koeficient v pořadí, zvyš řád aproximace. Proveď to pro všechny transformace a jdi na druhý bod.
5. Jestliže jsi použil všechny koeficienty spektra, vezmi další úsek signálu a pokračuj prvním bodem.
6. Pokud je již celý signál rekonstruován, vypočítej relativní globální kvadratickou chybu (Relative global squared error E_R). Během jejího výpočtu je zohledněna energie jednotlivých signálů.

Definice 5.3.1 *Kvadratická chyba (mean square error, MSE, E_m) je definována následovně:*

$$E_m = \frac{1}{N} \sum_{n=0}^{N-1} [f(n) - \hat{f}_m(n)]^2 \quad (5.7)$$

kde $f(n)$ je původní signál a $\hat{f}_m(n)$ rekonstrukce signálu (m -tá aproximace), N je obecně délka signálu, jednotka kvadratické chyby [E_m] je shodná s kvadrátem jednotky veličiny [v], pro niž je chyba udávána: [E_m] = [v^2]. Také lze definovat Relativní globální kvadratickou chybu (relative global squared error, E_R):

$$E_R = \frac{1}{N} \frac{\sum_{n=0}^{N-1} [f(n) - \hat{f}_m(n)]^2}{\sum_{n=0}^{N-1} f(n)^2} \quad (5.8)$$

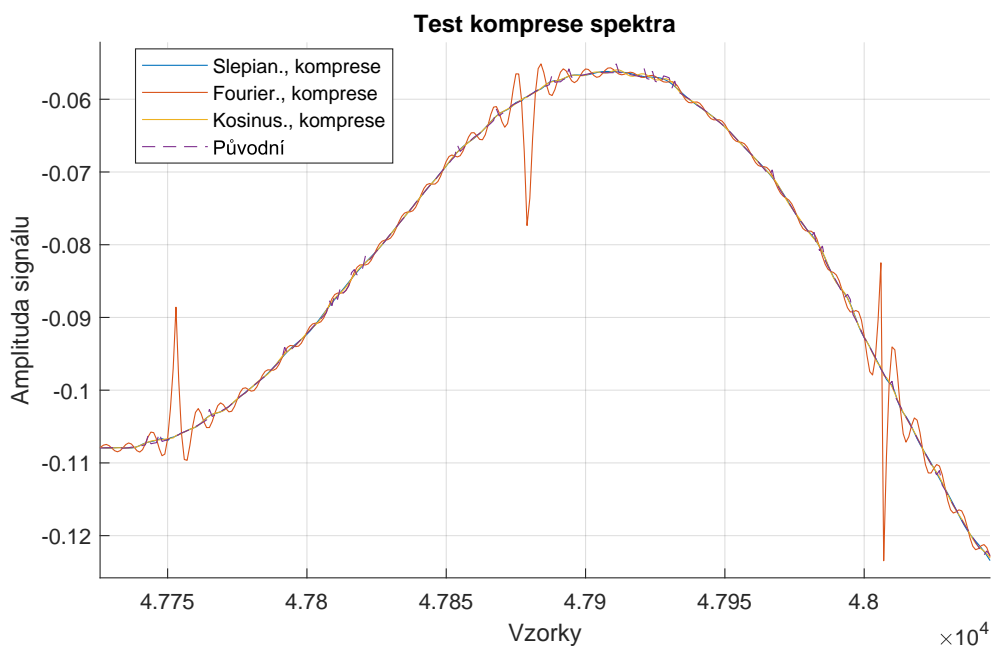
Chyba relativní E_R je udávána v procentech. Zdroj: [12].

Výsledek průběhu relativní kvadratické chyby E_R nejlépe ilustruje obr. 5.7. Nutno zdůraznit, že se jedná o hodnoty, které byly získány průměrem z hodnot získaných pro jednotlivé navzorkované úseky signálů (tomu odpovídá globální chyba). Slepianova transformace má E_R zpočátku značně vyšší, což je dáno neexistující konstantní (DC) složkou, již mají jak kosinová, tak i Fourierova transformace. Následně se ovšem

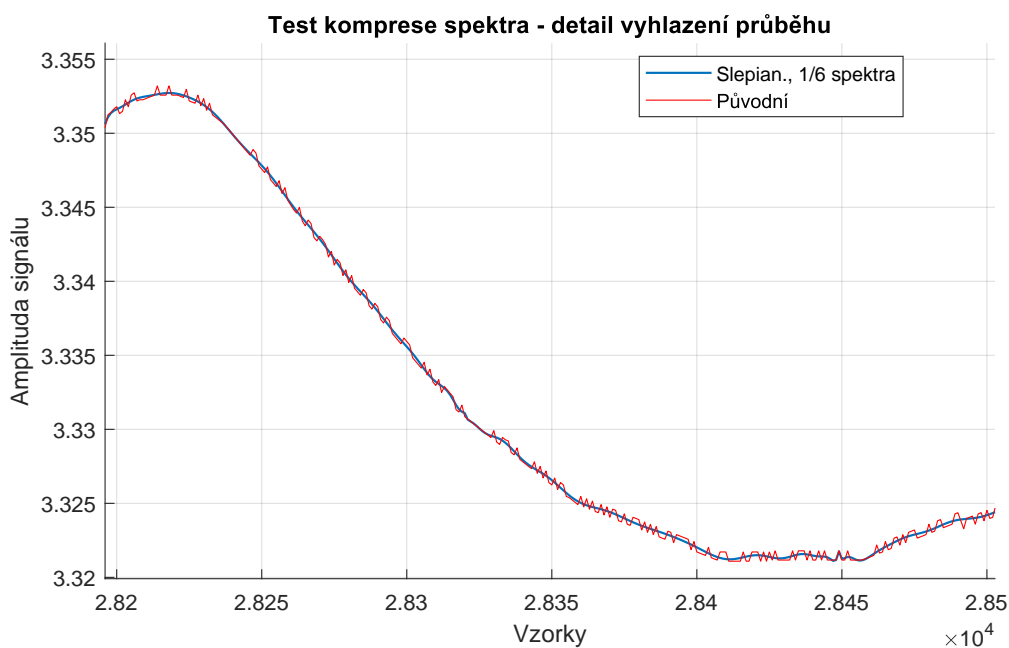
při desáté složce dostane až na hodnoty kolem 10^{-5} , což je o řád méně, než má kosinová transformace. Slepianova transformace má nejrychlejší konvergenci chyby E_R . Co se týče Fourierovy transformace, ta končí na obrázku 5.7 při $N/2$. složce. Oproti kapitole 3.5 bylo zde využito vlastnosti symetrie spektra, což ovšem v případné kompresi dat stejně nepomůže, neboť Fourierovo spektrum je komplexní, a tedy i při uložení pouze 1/2 symetrického spektra ukládáme stejný počet reálných čísel (vždy jedno reálné číslo pro reálnou a jedno pro imaginární část komplexního čísla musí být uloženo, popř. jedno pro amplitudu a druhé pro úhel). Výsledkem obr. 5.7 a obecně algoritmu pro porovnání výše uvedených transformací je opět potvrzení výhodnosti Slepianovy transformace, zde se jeví jako nejlepší již od 10 použitých spektrálních koeficientů z celkových 127.

Možný výsledek komprese signálu s využitím pouze části spektrálních složek (tzv. řídké reprezentace) je ilustrován na obr. 5.4. Je dobré si na něm povšimnout typické vlastnosti Fourierovy transformace, a sice „zakmitání“ rekonstruovaného signálu v místech hraničících s dalšími rekonstruovanými úseky. Další vlastnosti komprimovaného signálu ukazují také obr. 5.5 a 5.6. Zde je zajímavým faktem, že výstup snímače byl očividně ve stavu podobném hazardnímu stavu u logických obvodů a posílal střídající se sousední hodnoty (pulsy), což je pro zpracování signálu nežádoucí, jedná se o signál zašuměný. Díky Slepianově kompresi došlo zde k podstatnému vyhlazení průběhu signálu, což by mohlo být výhodné při dalším zpracování dat, jelikož by komprese provedla zároveň i průměrování a zjednodušení signálu.

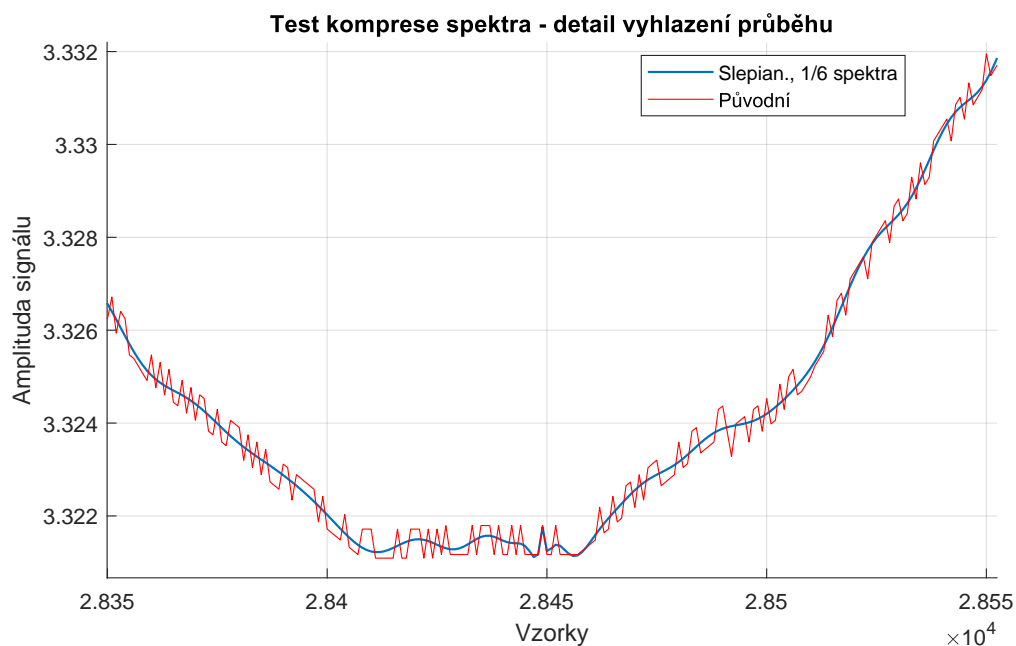
Nakonec je zde ještě uveden graf 5.8, který je obdobou 3.10 v sekci 3.6.



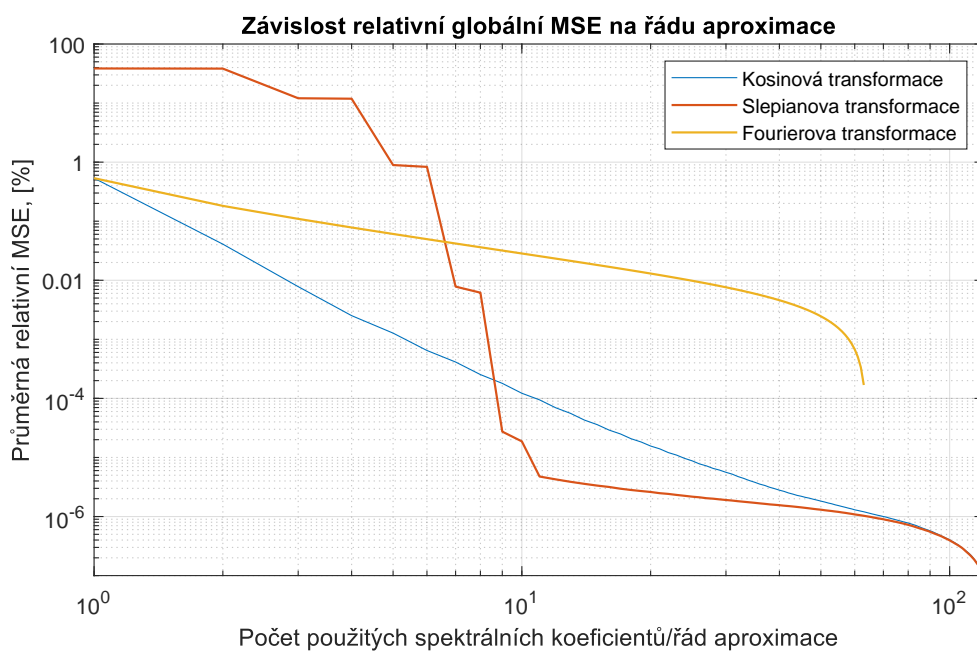
Obr. 5.4: Výřez signálu, kde bylo použito necelé spektrum, a sice pouze 1/4 spektra.



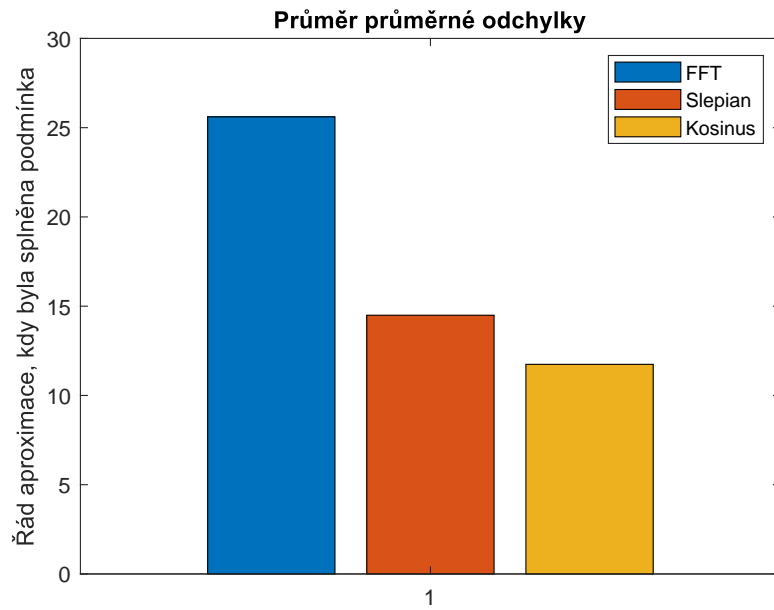
Obr. 5.5: Výřez signálu, kde bylo použito necelé spektrum, a sice pouze 1/6 spektra. Detail ukazuje vyhlazení průběhu (odstranění šumu díky odstranění vyšších frekvenčních složek).



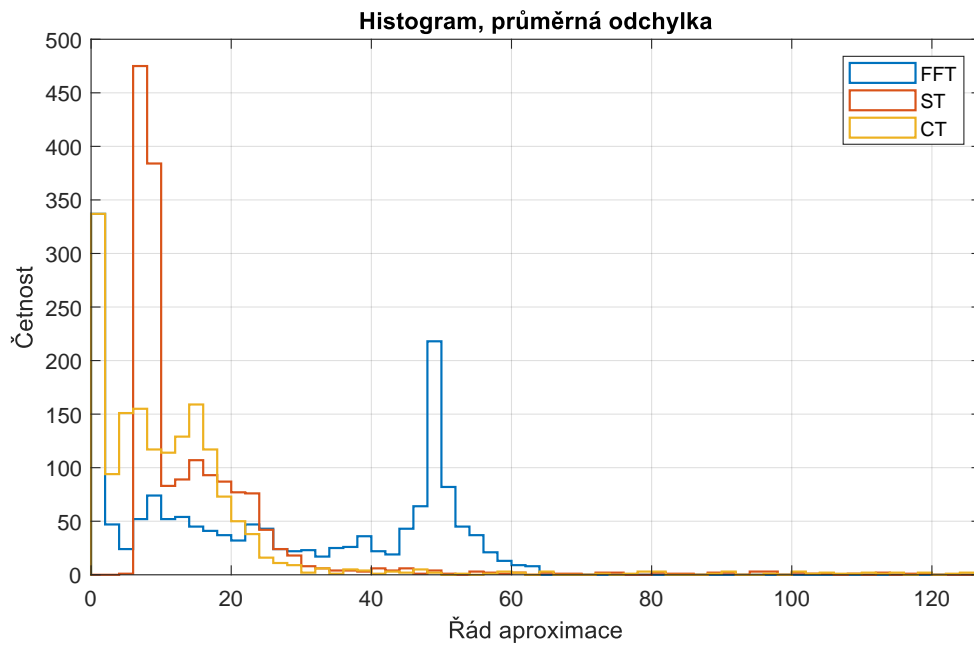
Obr. 5.6: Výřez signálu, kde bylo použito necelé spektrum, a sice pouze 1/6 spektra. Detail ukazuje vyhlazení průběhu (odstranění šumu díky odstranění vyšších frekvenčních složek).



Obr. 5.7: Závislost průměrné globální kvadratické chyby na řádu aproximace signálu.



Obr. 5.8: Průměrný řád aproximace, kdy pro FFT, CT a ST průměrná odchylka splnila podmínku (odpovídá srovnání v sekci 3.6).



Obr. 5.9: Histogram průměrného řádu aproximace, kdy pro FFT, CT a ST průměrná odchylka splnila podmínku (odpovídá srovnání v sekci 3.6) Fourierova transformace opět využívá symetrie spektra, tedy pouze 1/2 spektrálních koeficientů a její nejvyšší řád aproximace je 64.

Závěr

V práci byly shrnuty základní poznatky týkající se Slepianovy transformace a nástrojů pro její použití v aplikaci Matlab. Byly definovány základní pojmy, ověřena například ortogonalita Slepianovy ortogonální báze. Většina věcí byla ilustrována vhodnými obrázky či grafy.

Dále byla Slepianova transformace srovnána s rychlou Fourierovou transformací, popř. i Diskrétní kosinovou transformací. Porovnána byla spektra, ale i zpětná rekonstrukce signálu a to celková i po jednotlivých řádech aproximace, kdy byly postupně do rekonstruovaného signálu přidávány jednotlivé složky spekter, přičemž byla počítána chyba – o kolik se rekonstruovaný signál v dané iteraci liší od původního. Pro signál skládající se z nízkého počtu harmonických složek vycházela lépe Slepianova transformace (ve třetí kapitole). Dle výsledků dosažených v této kapitole by pro uložení signálu stačila pouze 1/2 či 1/3 Slepianova spektra.

Také byla ověřena možnost extrapolace signálu (prodloužení průběhu mimo interval pozorování) s pomocí Slepianovy ortogonální báze. Při správném nastavení parametrů použité báze bylo u zvoleného signálu dosaženo kvalitní extrapolace až pro 16 vzorků vně intervalu pozorování o 127 vzorcích. Signál byl extrapolován na obě strany, tzn., že celková extrapolace jej prodloužila cca o 25 %.

Nakonec byla ověřena možnost komprese spektra u reálného signálu. Různé transformace (Fourierova, kosinová a Slepianova) byly srovnány pomocí relativní globální kvadratické chyby (MSE). Z výsledného průběhu MSE vyplývá, že Slepianova transformace je i pro daný skutečný signál vhodným nástrojem k jeho kompresi, dokonce v některých parametrech je lepší, než v kompresi často používaná kosinová transformace. Slepianova ortogonální báze má tedy spoustu využití minimálně ve zkoumané skupině signálů a rozhodně stojí za další výzkum, kupříkladu co se týče zefektivnění algoritmu výpočtu.

Literatura

- [1] CHEN, Chi-Tsong. *Signals and Systems*. 3rd Edition. New York: Oxford University Press, 2004. ISBN 0-19-515661-7.
- [2] KUFNER, Alois a Jan KADLEC. *Fourierovy řady*. Praha: Academia, 1969.
- [3] HARDY, Godfrey Harold a Werner ROGOSINSKI. *Fourierovy řady*. Přeložil Alois KUFNER. Praha: Nakladatelství technické literatury, 1971.
- [4] MIHÁLIK, O.; MICHALÍK, D.; JURA, P.; FIEDLER, P. *Spectral Super-Resolution and Band-Limited Extrapolation Using Slepian Series*. In 16th IFAC Conference on Programmable Devices and Embedded Systems – PDeS 2019. IFAC-PapersOnLine (ELSEVIER). 52. IFAC, 2019. s. 388-393. ISSN: 2405-8963.
- [5] SLEPIAN, D. *Prolate spheroidal wave functions, Fourier analysis, and uncertainty–V: the discrete case*. Bell System Technical Journal, vol.57, no.5, pp.1371-1430, May 1978
- [6] *MathWorks Help Center: dpss* [online]. The MathWorks, 2022 [cit. 2022-12-26]. Dostupné z URL: <https://www.mathworks.com/help/signal/ref/dpss.html?searchHighlight=dpss&s_tid=srchtitle_dpss_1>
- [7] SLEPIAN, D. a H. O. POLLAK, *Prolate spheroidal wave functions, fourier analysis and uncertainty — I*. Bell System Technical Journal, vol. 40, no. 1, pp. 43-63, Jan. 1961, doi: 10.1002/j.1538-7305.1961.tb03976.x.
- [8] *MathWorks Help Center: digits* [online]. The MathWorks, 2023 [cit. 2022-04-06]. Dostupné z URL: <<https://www.mathworks.com/help/symbolic/digits.html>>
- [9] COCHRAN W. T., et al. *What is the fast Fourier transform?* in Proceedings of the IEEE, vol. 55, no. 10, pp. 1664-1674, Oct. 1967, doi: 10.1109/PROC.1967.5957.
- [10] CADZOW, J. *An extrapolation procedure for band-limited signals*. IEEE Transactions on Acoustics, Speech, and Signal Processing. 1979, 27(1), 4-12. ISSN 0096-3518. Dostupné z: doi:10.1109/TASSP.1979.1163187
- [11] EKSLER, Václav, Milan JELÍNEK a Wolfgang JAEGERS. *Audio bandwidth detection in the EVS codec*. 2015, IEEE Global Conference on Signal and Information Processing (GlobalSIP). 488-492. Dostupné z: doi:10.1109/GlobalSIP.2015.7418243

- [12] RAO, K. R. a P. YIP. *Discrete Cosine Transform: Algorithms, Advantages, Applications*. San Diego, CA: Academic Press, 1990. ISBN 0-12-580203-X.
- [13] *MathWorks Help Center: dct* [online]. The MathWorks, 2023 [cit. 2023-03-27]. Dostupné z URL: <<https://www.mathworks.com/help/signal/ref/dct.html#bvksip-1>>
- [14] MICHALÍK, David. *Simulátor řízení vozidla*. Brno, 2019. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Ing. Miroslav Jirgl, Ph.D.
- [15] MICHALÍK, D., O. MIHÁLIK, M. JIRGL a P. FIEDLER. *Driver Behaviour Modeling With Vehicle Driving Simulator*. In: IFAC PapersOnLine [online]. Elsevier, 2019, s. 180-185 [cit. 2023-03-27]. ISSN 2405-8963. Dostupné z: doi:10.1016/j.ifacol.2019.12.753

Seznam symbolů a zkratek

DCT	z angl. <i>Discrete Cosine transform</i> , Diskrétní kosinová transformace, var. IDCT pro inverzní tr.
FFT	z angl. <i>Fast Fourier transform</i> , Rychlá Fourierova transformace, var. IFFT pro inverzní tr.
ST	z angl. <i>Slepian transformation</i> , Slepianova transformace, var. IST pro inverzní tr.
ℓ_2	Hilbertův prostor diskrétních funkcí ℓ_2
\bar{a}	komplexně sdružené číslo k číslu a
$\psi(t), \psi(c, t)$	Slepianova funkce ve spojitě časové doméně t
$\psi(k), \psi(c, k)$	Slepianova sekvence v diskrétní časové doméně $t = k \cdot T_S$
\hat{f}_m	aproximace signálu f , kde m značí řád aproximace
f_S	vzorkovací frekvence [Hz], [s^{-1}]
ω_S	vzorkovací frekvence [rad/s]
T_S	vzorkovací perioda [s]
Ω	bandlimit, nejvyšší frekvence frekvenčně omezeného signálu [rad/s]
W	normalizovaný bandlimit, $W = 1/2$ za předpokladu, že $\omega_S = 1$

A Obsah elektronické přílohy

V příloze jsou soubory .m obsahující kód programu pro Matlab. Spouštěny byly ve verzi Matlab R2022b.

```
/.....kořenový adresář přiloženého archivu  
├── dpss_main.m  
├── dpss_average.m  
├── dpss_NW_dependency.m  
├── dpss_signal_auto.m  
├── funkce_generate_signal.m  
├── funkce_generate_signal_rnd.m  
├── funkce_load_data.m  
├── 001_23_7_2020_step.csv  
└── 002_21_7_2020_step.csv
```