

Frekvenční analýza signálů a návrhy číslicových filtrů

Bc. Roman Perutka

Diplomová práce
2025



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2024/2025

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	Bc. Roman Perutka
Osobní číslo:	A23439
Studijní program:	N0613A140022 Informační technologie
Specializace:	Softwarové inženýrství
Forma studia:	Kombinovaná
Téma práce:	Frekvenční analýza signálů a návrhy číslicových filtrů
Téma práce anglicky:	Frequency Analysis of Signals and Design of Digital Filters

Zásady pro vypracování

- Uvedte základní pojmy z oblasti frekvenční analýzy číslicových signálů.
- V programovacím jazyce C# realizujte vybrané vhodné algoritmy pro frekvenční analýzu číslicových signálů.
- Uvedte základní pojmy z oblasti číslicové filtrace.
- V programovacím jazyce C# realizujte návrh FIR filtru s využitím metody inverzní Fourierovy transformace a oken.
- V programovacím jazyce C# naprogramujte algoritmus návrhu filtru s využitím metody analogových prototypů.
- V programovacím jazyce C# vytvořte nástroj pro výpočet základních charakteristik filtrů.
- V programovacím jazyce C# vytvořte program pro filtraci signálů vytvořenými filtry.

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

Seznam doporučené literatury:

1. TAN, Li. *Digital Signal Processing*. Elsevier, 2008. ISBN 978-0-12-374090-8.
2. DINIZ, Paulo S. R.; DA SILVA, Eduardo A. B. a NETTO, Sergio L. *Digital Signal Processing*. 2. vydání. Elsevier, 2008. ISBN 978-0-521-88775-5.
3. DOŇAR, Bohuslav a ZAPLATÍLEK, Karel. *MATLAB – začínáme se signály*. BEN, 2006. ISBN 8073002000.
4. OPPENHEIM, Alan V.; WILLSKY, Alan S. a NAWAB, Syed Hamid. *Signals & Systems*. 2. vydání. Prentice-Hall International, 1997. ISBN 9780136511755.
5. VÍCH, Robert a SMÉKAL, Zdeněk. *Číslicové filtry*. Academia, 2001. ISBN 978-80-200-0761-2.
6. HLAVÁČ, Václav a SEDLÁČEK, Miloš. *Zpracování signálů a obrazů*. ČVUT, 2009. ISBN 978-80-01-04442-1.
7. DAVÍDEK, Vratislav; VLČEK, Jaroslav a LAIPERT, Miloš. *Analogové a číslicové filtry*. 2. vydání. ČVUT, 2006. ISBN 80-01-03026-1.

Vedoucí diplomové práce: **prof. Ing. Marek Kubalčík, Ph.D.**
Ústav řízení procesů

Datum zadání diplomové práce: **27. října 2024**
Termín odevzdání diplomové práce: **2. června 2025**



L.S.

doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan

prof. Mgr. Roman Jašek, Ph.D., DBA v.r.
ředitel ústavu

Ve Zlíně dne 17. listopadu 2024

PROHLÁŠENÍ AUTORA DIPLOMOVÉ PRÁCE

Beru na vědomí, že

- odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., v platném znění bez ohledu na výsledek obhajoby;
- diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému a bude dostupná k nahlédnutí;
- jedno vyhotovení diplomové práce v listinné podobě bude ponecháno Univerzitě Tomáše Bati ve Zlíně k uložení;
- 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;
- podle § 60 odst. 1 autorského zákona má Univerzita Tomáše Bati ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- podle § 60 odst. 2 a 3 mohu užít své dílo – diplomovou práci – nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tj. k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- 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(a) samostatně a použitou literaturu jsem řádně citoval(a); v případě publikace výsledků budu uveden(a) jako spoluautor;
- odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou obsahově totožné.

Ve Zlíně, dne

.....
podpis autora

ABSTRAKT

Tato diplomová práce se zabývá problematikou frekvenční analýzy číslicových signálů a návrhem číslicových filtrů typu FIR (filtr s konečnou impulzní odezvou) a IIR (filtr s nekonečnou impulzní odezvou). Hlavním cílem práce je vytvořit program ve vývojovém prostředí Visual Studio ve formě Windows Form application, kde budou realizovány algoritmy DFT (diskrétní Fourierova transformace) a FFT (rychlá Fourierova transformace), návrhy FIR a IIR filtrů a filtrace zadaného vstupního signálu těmito filtry. Dále jsou v praktické části popsány možnosti programu a příklady. V teoretické části jsou uvedeny základní principy realizovaných algoritmů.

Klíčová slova: Číslicové signály, Frekvenční analýza, Fourierova transformace, DFT, FFT, Filtr, FIR, IIR

ABSTRACT

This diploma thesis deals with the frequency analysis of digital signals and the design of digital filters of the FIR (finite impulse response filter) and IIR (infinite impulse response filter) type. The main goal of the thesis is to create a program in the Visual Studio development environment in the form of a Windows Form application, where the DFT (discrete Fourier transform) and FFT (fast Fourier transform) algorithms will be implemented, the design of FIR and IIR filters and the filtering of the specified input signal by these filters. Furthermore, the practical part describes the program options and examples. The theoretical part presents the basic principles of the implemented algorithms.

Keywords: Digital signals, Frequency analysis, Fourier transform, DFT, FFT, Filter, FIR, IIR

Tímto bych chtěl poděkovat svému vedoucímu práce profesoru Ing. Marku Kubalčíkovi, Ph.D. za odbornou pomoc, rady a vřelý přístup při zpracovávání této práce.

OBSAH

ÚVOD.....	11
I TEORETICKÁ ČÁST	12
1 SIGNÁLY A JEJICH TYPY	13
1.1 SPOJITÝ SIGNÁL	13
1.2 DISKRÉTNÍ SIGNÁL	14
1.3 ANALOGOVÝ SIGNÁL.....	14
1.4 DIGITÁLNÍ SIGNÁL.....	14
1.5 TYPY SIGNÁLŮ PODLE SPOJITOSTI.....	14
1.6 VZORKOVÁNÍ	15
1.7 VZORKOVACÍ TEORÉM.....	15
1.8 KVANTOVÁNÍ	16
2 FOURIEROVA ŘADA A FOURIEROVA TRANSFORMACE.....	17
2.1 FOURIEROVA ŘADA	17
2.2 DIRICHLETOVY PODMÍNKY	17
2.3 SPOJITÁ FOURIEROVA TRANSFORMACE	18
2.4 DISKRÉTNÍ FOURIEROVA TRANSFORMACE	19
2.5 RYCHLÁ FOURIEROVA TRANSFORMACE – PRINCIP COOLEY-TUKEY	19
3 ČÍSLICOVÁ FILTRACE.....	21
3.1 ZÁKLADNÍ TYPY FREKVENČNÍCH ČÍSLICOVÝCH FILTRŮ	21
3.1.1 Dolní propust.....	21
3.1.2 Horní propust	22
3.1.3 Pásmová propust	22
3.1.4 Pásmová zádrž.....	23
3.2 FILTRY PODLE IMPULSNÍ ODEZVY	23
3.2.1 Filtry s konečnou impulsní odezvou (FIR)	23
3.2.2 Funkce oken	25
3.2.3 Filtry s nekonečnou impulsní odezvou (IIR)	28
II PRAKTICKÁ ČÁST	32
4 TAB 1: DISKRÉTNÍ FT A RYCHLÁ FT.....	33
4.1 METODA DISKRÉTNÍ FOURIEROVY TRANSFORMACE	35
4.2 METODA RYCHLÉ FOURIEROVY TRANSFORMACE	36
5 TAB 2: NÁVRH FIR FILTRU	41
5.1 VÝPOČET VAH FILTRU	43
5.1.1 Horní a dolní propust	43
5.1.2 Pásmová zádrž a propust	45

5.2	FUNKCE OKEN	47
5.2.1	Obdélníkové	47
5.2.2	Hamming	48
5.2.3	Hann	49
5.2.4	Trojúhelníkové	50
5.2.5	Blackman	51
6	TAB 3 FIR FILTR.....	52
6.1.1	Příklad FIR Filtrace	54
7	TAB 4 NÁVRH IIR FILTRU	55
7.1	ANALOGOVÝ PROTOTYP BUTTERWORTH 2. ŘÁD.....	55
7.2	DOLNÍ PROPUST	56
7.3	HORNÍ PROPUST	59
7.4	PÁSMOVÁ ZÁDRŽ.....	60
7.5	PÁSMOVÁ PROPUST	63
8	TAB 5 IIR FILTR.....	66
8.1.1	Příklad IIR Filtrace	66
	ZÁVĚR.....	69
	SEZNAM POUŽITÉ LITERATURY	70
	SEZNAM OBRÁZKŮ	75
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	77
	SEZNAM PŘÍLOH.....	78

ÚVOD

V dnešní době, kdy číslicové technologie pronikají do všech oblastí lidské činnosti, hraje číslicové zpracování signálů (DSP – Digital Signal Processing) klíčovou roli v mnoha aplikacích. Od telekomunikací, přes zpracování zvuku a obrazu, až po řízení průmyslových procesů – všude tam, kde je potřeba analyzovat, upravit nebo filtrovat signál, se uplatňují metody číslicové filtrace a frekvenční analýzy. Schopnost efektivně rozložit signál do frekvenčních složek a následně potlačit nežádoucí části je nezbytným předpokladem pro dosažení vysoké kvality přenosu nebo zpracování dat.

Tato diplomová práce se zabývá problematikou frekvenční analýzy číslicových signálů a návrhem číslicových filtrů typu FIR (filtr s konečnou impulzní odezvou) a IIR (filtr s nekonečnou impulzní odezvou). Hlavním cílem práce je vytvořit interaktivní program ve vývojovém prostředí Visual Studio ve formě aplikace Windows Forms, který uživateli umožní analyzovat vstupní signál pomocí diskrétní Fourierovy transformace (DFT) a její optimalizované verze – rychlé Fourierovy transformace (FFT). Dále program nabídne nástroje pro návrh číslicových filtrů typu FIR a IIR a jejich aplikaci na zvolený signál.

Teoretická část práce se věnuje popisu základních principů číslicového zpracování signálů, včetně principu DFT, FFT a návrhu číslicových filtrů FIR a IIR.

Praktická část se soustředí na návrh a implementaci aplikace, včetně popisu použitých algoritmů a grafického uživatelského rozhraní. Jsou zde rovněž uvedeny příklady využití programu s konkrétními ukázkami signálů a jejich zpracováním, které slouží k ověření funkčnosti navrženého softwarového řešení a jeho schopnosti efektivně analyzovat a filtrovat číslicové signály.

Tato práce si klade za cíl propojit teoretické poznatky s praktickým využitím v podobě nástroje, který může sloužit nejen jako výuková pomůcka, ale i jako základ pro další rozšiřování v oblasti DSP. Přílohou práce je kompletní zdrojový kód programu.

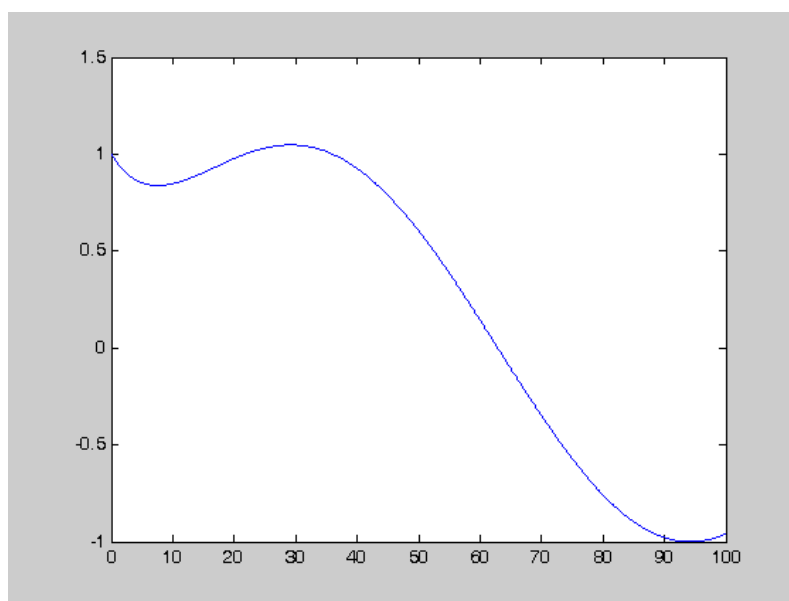
I TEORETICKÁ ČÁST

1 SIGNÁLY A JEJICH TYPY

1.1 Spojitý signál

Jako spojité signály označujeme signály, které mají definovanou hodnotu v každém časovém okamžiku. Signály, které jsou spojité, je při ukládání do elektroniky nutné převést na diskrétní a digitální. Postupy k těmto úpravám nazýváme vzorkování a kvantování.

[1][2][3][4]

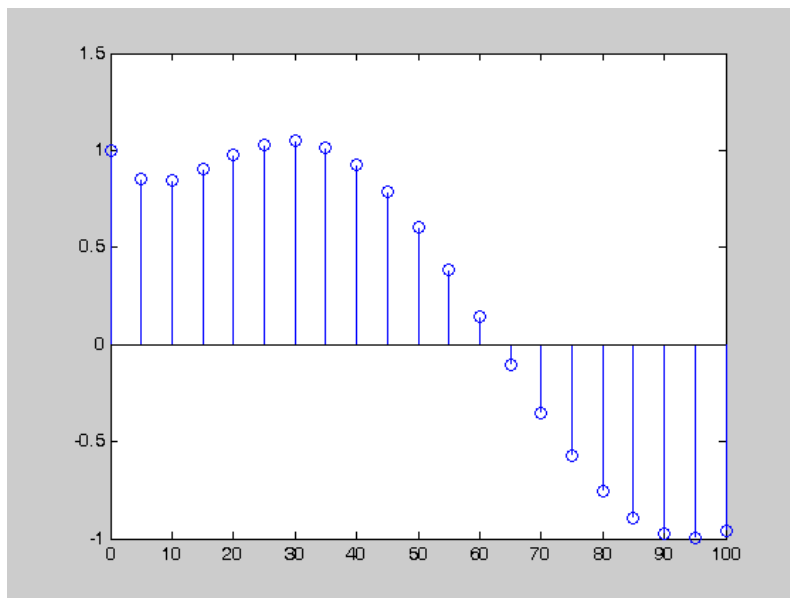


Obrázek 1: Spojitý signál

Takové signály se vyskytují všude kolem nás – hlas, kmitavé pohyby, srdeční aktivita jsou všechno jevy, které se dají popsat spojitým signálem.

1.2 Diskrétní signál

Signál označíme jako diskrétní, pokud nabývá hodnoty jen v určitých časových intervalech, tyto časové intervaly označujeme jako vzorkovací periodu. Převrácená hodnota vzorkovací periody je vzorkovací frekvence. [1][2][3][4]



Obrázek 2: Diskrétní signál

1.3 Analogový signál

Signál označujeme jako analogový, pokud je jeho obor hodnot (hodnoty, kterých může signál nabývat) spojitý. [1][2][3]

1.4 Digitální signál

Digitální signál vzniká z analogového signálu procesem kvantování, kde dochází k zaokrouhlování hodnot do jednotlivých kvantovacích hladin. [1][2][3][4]

1.5 Typy signálů podle spojitosti

- Spojitý analogový signál
- Diskrétní analogový signál
- Číslicový signál

Signály v přírodě se většinou vyskytují jako spojité analogové, pro potřeby počítačů převádíme na digitální (číslíkový) signál, který je možný uložit a upravovat.

1.6 Vzorkování

V určitých podmínkách můžeme spojitý signál reprezentovat a znovu složit pomocí jeho hodnot neboli vzorků v časových okamžicích rovnoměrně rozložených v průběhu tohoto signálu. [1][2][3][4]

Vzorkování je proces, při kterém převádíme spojitý signál na diskrétní, abychom s ním mohli počítat a byli ho schopni uložit/použít v elektronice. Toto probíhá výběrem bodů, které spojitý signál budou reprezentovat. Získáváme množinu diskrétních bodů, tedy diskrétní signál, který představuje původní spojitý signál s dostatečným detailem pro naše výpočty. Kvůli této vlastnosti je nutné vhodně stanovit vzorkovací frekvenci. [1][3]

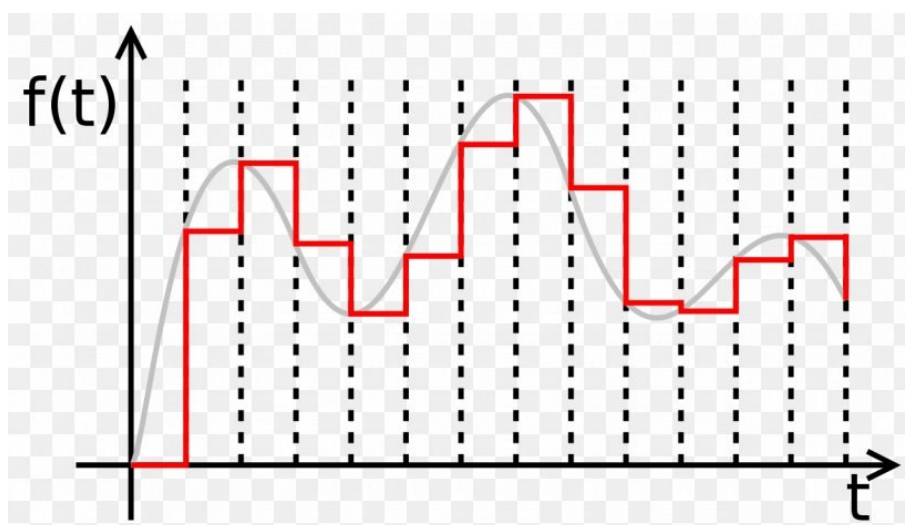
Tento postup využívá tzv. **Vzorkovací teorém (Nyquistův-Shannon-Kotelnikův teorém)**. [5]

1.7 Vzorkovací teorém

Požívá se v mnoha odvětvích lidské činnosti, například ve filmech, kdy je počet snímků, tedy vzorků, tak početný, že se pro nás obraz zdá naprosto plynulý.

Teorém nám říká, že spojitý signál je možné přesně reprezentovat diskrétním signálem, je-li vzorkovací frekvence alespoň dvakrát větší než maximální frekvence vzorkovaného signálu. [5]

Při nedodržení tohoto pravidla může dojít k aliasingu, což jsou výrazné odchylky při zpětném převodu signálu z diskrétního na spojitý. [6]



Obrázek 3: Vzorkování signálu [7]

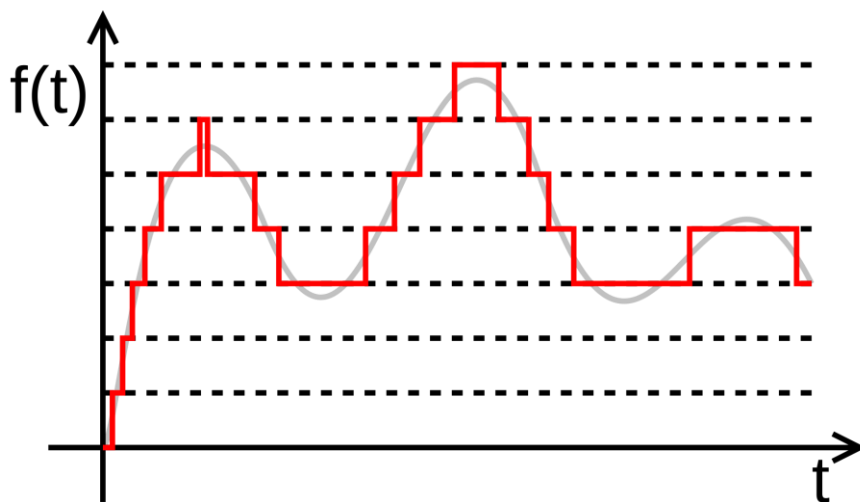
1.8 Kvantování

Kvantování je proces, kdy se do různých úrovní snažíme rozdělit hodnoty, kterých signál nabývá, abychom opět dosáhli diskretizace signálu, tentokrát v oboru hodnot.

Při tomto procesu dochází k nevratné ztrátě informace důsledkem zaokrouhlování, proto je nutné vhodně nastavit hladiny, respektive prostor mezi hladinami, aby ztráta informace neovlivnila celkovou kvalitu signálu a informace, kterou signál přenáší, zůstala nezměněná. [8]

Podobný koncept se používá v počítačové grafice, kde je předmětem hloubka barvy. Více hladin znamená více bitů pro ukládání barvy a tím vyšší kvalita, ale větší velikost obrázku.

A/D převodník, který v počítači poskytuje tyto funkce, pracuje s bity, čili ve dvojkové soustavě. Kvůli tomu je počet hladin mocninou dvou, aby šla dobře vyjádřit pomocí bitů. Počet hladin může tedy být 2, 4, 8, 16, 32, ..., 65 536, ... [9]



Obrázek 4: Kvantování signálu [10]

2 FOURIEROVA ŘADA A FOURIEROVA TRANSFORMACE

2.1 Fourierova řada

Fourierova řada je aproximační metodou pro funkce pomocí řady goniometrických funkcí, konkrétně sinů a kosinů.

Pokud máme signál, který splňuje následující podmínky, je možné ho rozložit na nekonečnou řadu harmonických funkcí, které se liší od sebe amplitudou, kmitočtem a počáteční fází, tuto řadu nazýváme Fourierova řada. [11][12]

- Je periodický
- Je spojitý
- Musí konvergovat
- Musí splňovat Dirichletovy podmínky [13]

2.2 Dirichletovy podmínky

- Funkce musí být v rámci své periody ohraničená
- Funkce má v rámci své periody konečný počet nespojitostí prvního druhu
- V rámci své periody má funkce konečný počet lokálních extrémů, je po částech monotónní a po částech hladká

Tato řada má podobu součtů sinů a kosinů násobených koeficienty.

Trigonometrický tvar Fourierovy řady [11]:

$$s_p(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n * \cos n\omega_0 t + b_n * \sin n\omega_0 t)$$

Existují i další tvary pro tuto řadu:

Kosinový tvar Fourierovy řady [14]:

$$s_p(t) = \frac{c_0}{2} + \sum_{n=1}^{\infty} (c_n * \cos n\omega_0 t + \varphi_n)$$

Komplexní (exponenciální) tvar Fourierovy řady [14]:

$$s_p(t) = \sum_{n=1}^{\infty} (c_n * e^{jn\omega_0 t})$$

Cílem je získat frekvenční charakteristiku a spektrum periodických signálů. Spektrum periodických signálů je diskrétní, vzdálenost mezi čarami spektra je rovna kmitočtu časového průběhu.

Pro popis aperiodických signálů nelze použít Fourierovu řadu.

2.3 Spojitá Fourierova Transformace

Pro popis aperiodických signálů je nutné využít takzvanou Fourierovu transformaci.

Jedná se o matematický proces, který rozkládá aperiodický signál na jeho frekvenční složky. Transformuje signál z časové do frekvenční domény. Díky tomu můžeme signál analyzovat z hlediska jeho frekvenčních vlastností. [15][16]

$$X_{(f)} = \int_{-\infty}^{\infty} x_{(t)} * e^{-j2\pi ft} dt$$

Kde:

$X_{(f)}$ je hodnota signálu ve frekvenční doméně

$x_{(t)}$ je hodnota signálu v časové doméně

t je časový okamžik

f je frekvence signálu

Výsledkem tohoto výpočtu je frekvenční spektrum aperiodického signálu, které je tvořené komplexní funkcí. Spektrum takových signálů je spojitě. [15][16]

2.4 Diskrétní Fourierova Transformace

Pro práci se signály v počítačích musíme ze spojitých signálů vytvořit diskrétní signály vzorkováním a kvantováním. Pro tyto signály pak používáme upravený vztah Fourierovy transformace – diskrétní Fourierova transformace (DFT). Z integrace se nám stává suma a pohybujeme se od nultého až po poslední prvek podle vztahu: [17]

$$X_k = \sum_{n=0}^{N-1} x_n * e^{\frac{-j2\pi kn}{N}}$$

Kde:

N je počet koeficientů

n je pořadové číslo hodnoty signálů

k je pořadové číslo současné počítané hodnoty

vztah $e^{jx} = \cos x + j \sin x$ pro přepis exponentu na goniometrický tvar

2.5 Rychlá Fourierova Transformace – princip Cooley-Tukey

Jedná se o algoritmus rozděl a panuj – složitý problém rozkládáme na dílčí problémy, kde se složitost buď rozloží, nebo úplně zmizí. V tomto případě se rozkladem na dílčí sumy zbavujeme exponentu a tím i násobení v rámci jednotlivých členů. Tento postup pro rychlou transformaci nazýváme podle autorů Cooley-Tukey.

Rychlá FT je optimalizací DFT, kde celkovou sumu rozdělujeme na dílčí sumy lichých a sudých koeficientů (dle jejich pořadí), jejichž součet je ekvivalentní jako naše původní suma podle vztahu: [18][19]

$$X_k = \sum_{n=0}^{N-1} x_n * e^{\frac{-j2\pi kn}{N}}$$

$$X_k = \sum_{n=0}^{N/2-1} x_{2n} * e^{\frac{-j2\pi kn}{N}} + e^{\frac{-j2\pi k}{N}} \sum_{n=0}^{N/2-1} x_{2n+1} * e^{\frac{-j2\pi kn}{N}}$$

Dělení probíhá vždy na sudou(koeficienty 0, 2, 4, 6, ...) a lichou(koeficienty 1, 3, 5, 6, ...) část, kde sudá část (levá suma) je bez koeficientu a pravá část k sobě získává koeficient $e^{\frac{-j2\pi k}{N}}$, tento koeficient je jiný při každém řádu rozdělování sumy podle vztahu:

$$Ck = e^{\frac{-j2^r \pi k}{N}}$$

Kde:

r je pořadí rozdělování sumy, pro původní sumu $r=0$, pro první rozdělování $r=1$, pro druhé rozdělování $r=2$

Podle vzorce by při dalším rozdělení na sudou a lichou sumu koeficient byl:

$$e^{\frac{-j2^2 \pi k}{N}} = e^{\frac{-j4 \pi k}{N}}$$

Rozdělování sum pokračuje, dokud nemáme sumy o velikosti 1, díky čemuž můžeme Eulerovo číslo v sumě odstranit, protože počítaný koeficient je vždy nultý, tudíž celý exponent je také roven 0 - sčítáme pouze velikosti koeficientů násobené přidanými koeficienty lichých sum.

Pro dva koeficienty by FFT vypadala následovně:

$$X_k = \sum_{n=0}^{N/2-1} x_0 + e^{\frac{-j2\pi k}{N}} \sum_{n=0}^{N/2-1} x_1$$

Pro čtyři koeficienty by FFT vypadala následovně:

$$X_k = \sum_{n=0}^{N/4-1} x_0 + e^{\frac{-j4\pi k}{N}} \sum_{n=0}^{N/4-1} x_2 + e^{\frac{-j2\pi k}{N}} \left(\sum_{n=0}^{N/4-1} x_1 + e^{\frac{-j4\pi k}{N}} \sum_{n=0}^{N/4-1} x_3 \right)$$

Tato metoda je **použitelná jen v případě**, že počet počítaných koeficientů je mocnina dvou (2, 4, 8, 16). Pokud tato podmínka není splněna, použije se algoritmus DFT nebo je počet hodnot signálu doplněn nulovými, to ale vede ke zkreslení.

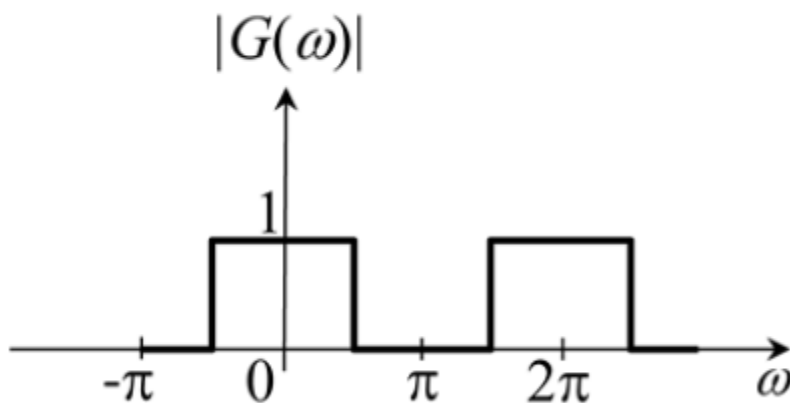
3 ČÍSLICOVÁ FILTRACE

3.1 Základní typy frekvenčních číslicových filtrů

Frekvenčně selektivní filtry propustí složky signálu, jen pokud jejich frekvence vyhovují frekvenčním vlastnostem filtru. Frekvenční vlastnosti filtru jsou dány mezními frekvencemi. Složky signálu, které nevyhovují, jsou filtrem odstraněny.

3.1.1 Dolní propust

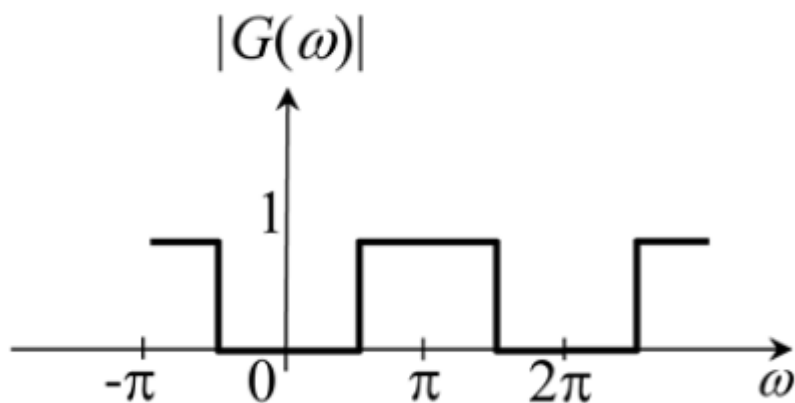
Signály projdou přes filtr, pokud je jejich frekvence menší, než je mezní frekvence filtru dolní propusti. [20][21][22]



Obrázek 5: Ideální dolní propust [23]

3.1.2 Horní propust

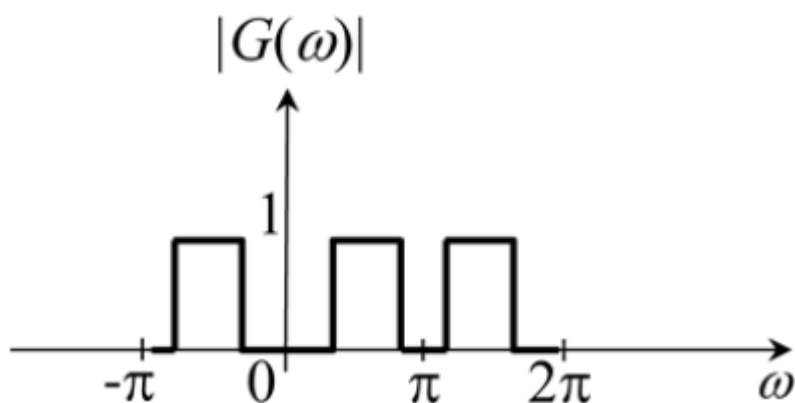
Signály projdou přes filtr, pokud je jejich frekvence větší, než je mezní frekvence filtru horní propusti. [20][21][22]



Obrázek 6: Ideální horní propust [23]

3.1.3 Pásmová propust

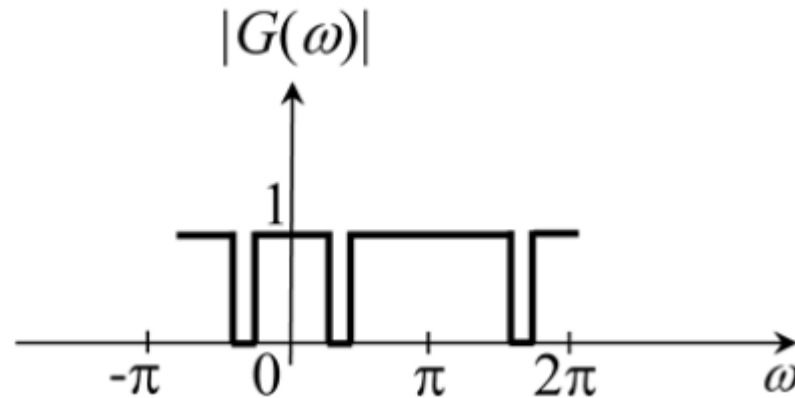
Pásmová propust využívá pás frekvencí, které jsou filtrem propuštěny, k tomu využívá dvě mezní frekvence – horní a dolní mez. Signály, jejichž frekvence se nenachází v tomto pásmu jsou odstraněny. [20][21][22]



Obrázek 7: Ideální pásmová propust [23]

3.1.4 Pásmová zadrž

Pásmová zadrž využívá pás frekvencí podobně jako propust, ale s opačným přístupem. Signály, jejichž frekvence se nachází v tomto pásmu jsou odstraněny. [20][21][22]



Obrázek 8: Ideální pásmová zadrž [23]

3.2 Filtry podle impulsní odezvy

Existují dva typy digitálních filtrů – s konečnou a nekonečnou odezvou.

3.2.1 Filtry s konečnou impulsní odezvou (FIR)

Impulsní odezva FIR filtru je charakterizována konečným počtem hodnot. Filtry tohoto typu jsou vždy stabilní. Nevýhodou těchto filtrů je, že vyžadují vysoký řád.[24][25][26][27]

Řád ovlivňuje průběh výpočtu a počet hodnot, které filtr spočítá a které se účastní jako váhy u jednotlivých vstupů signálu při průchodu filtrem.

Metody návrhu:

- Inverzní Fourierova transformace a okna
- Vzorkování kmitočtové charakteristiky

V práci je použita metoda Inverzní Fourierova transformace a okna.

Inverzní Fourierova transformace z frekvenční charakteristiky:

$$b_{(n)} = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(\omega) e^{j\omega k} d\omega = \frac{1}{2\pi} \int_{-\omega_h}^{\omega_h} e^{j\omega k} d\omega = \frac{\omega_h}{\pi} * \frac{\sin(\omega_h k)}{k\omega_h}$$

Kde:

$$\omega_h = \frac{2\pi f_c}{f_s}$$

k je číslo závislé na řádu

f_c je mezní frekvence

f_s je vzorkovací frekvence

Diferenční rovnice pro výpočet výstupní hodnoty signálu z filtru:

$$y(k) = b_0 x(k) + b_1 x(k-1) + b_2 x(k-2) + \dots + b_m x(k-m)$$

Kde:

m je řád filtru

b váha filtru/koefficient filtru

k je diskretní čas, kde (k) je současná hodnota a $(k-1)$ je hodnota signálu o jednu časovou jednotku dříve

x je vstupní hodnota signálu

$y(k)$ je výstupní hodnota signálu

Na tuto výstupní hodnotu se poté aplikuje váha z vybraného okna násobením.

Výhody FIR filtru: [24]

- Jednoduchý návrh filtru
- Stabilita
- Není v základu rekurzivní

- Linearita

Nevýhody FIR filtru: [24]

- Pro kvalitní řešení je nutný velký řád filtru
- Výpočetní náročnost

3.2.2 Funkce oken

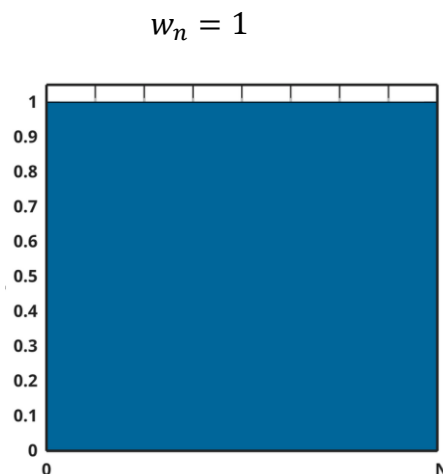
Okna neboli váhové funkce slouží k lepší aproximaci ideální amplitudové frekvenční charakteristiky filtru skutečnou charakteristikou, než je tomu při prostém výběru koeficientů, tedy použití obdélníkového okna.

Okna mají velikost a počet prvků závislé na řádu filtru – pro specifický řád filtru je nutné spočítat hodnotu této funkce a výsledné hodnoty aplikovat na odpovídající hodnotu z filtru (filtrovanou hodnotu signálu). [27][28]

V práci jsou implementované následující okna:

- **Obdélníkové okno**

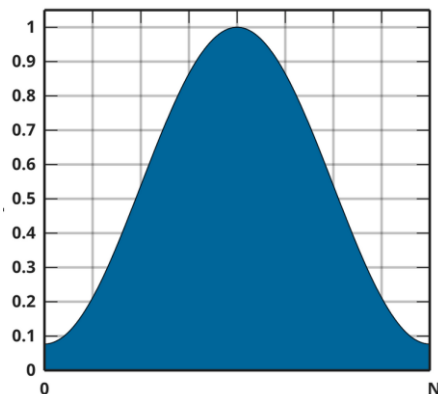
Okno, kde všechny členy nabývají hodnoty 1, takže okno nemá žádný vliv na váhy, výstupem jsou potom původní koeficienty filtru.



Obrázek 9: Obdélníkové okno [29]

- **Hammingovo okno**

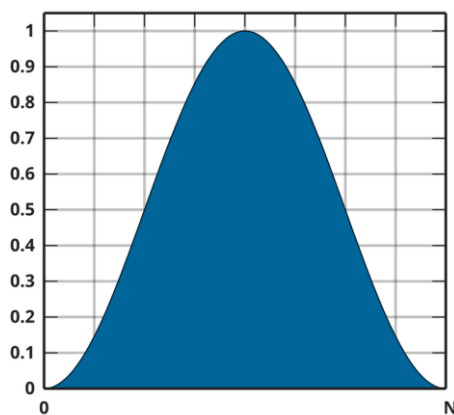
$$w_n = 0.54 - (0.46 * \cos(\frac{2\pi n}{N-1}))$$



Obrázek 10: Hammingovo okno [30]

- **Hannovo okno**

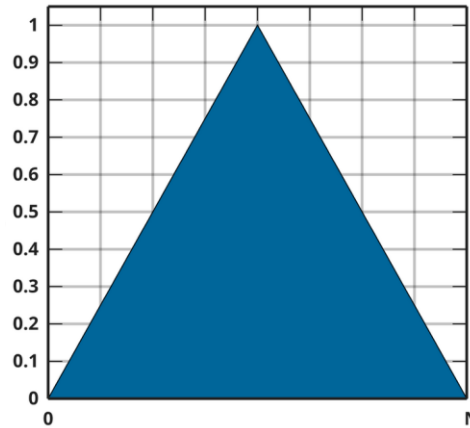
$$w_n = 0.5 * (1 - \cos(\frac{2\pi n}{N-1}))$$



Obrázek 11: Hannovo okno [31]

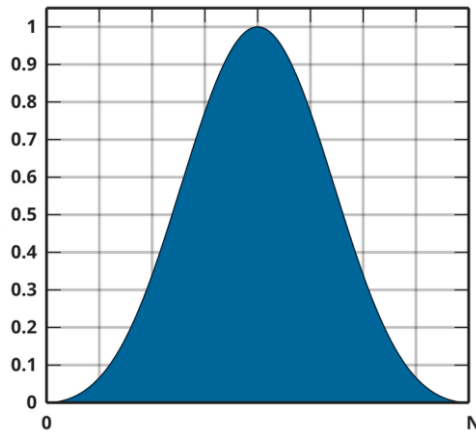
- **Trojúhelníkové okno**

$$w_n = 1 - \left| \frac{n - \frac{N-1}{2}}{\frac{N-1}{2}} \right|$$



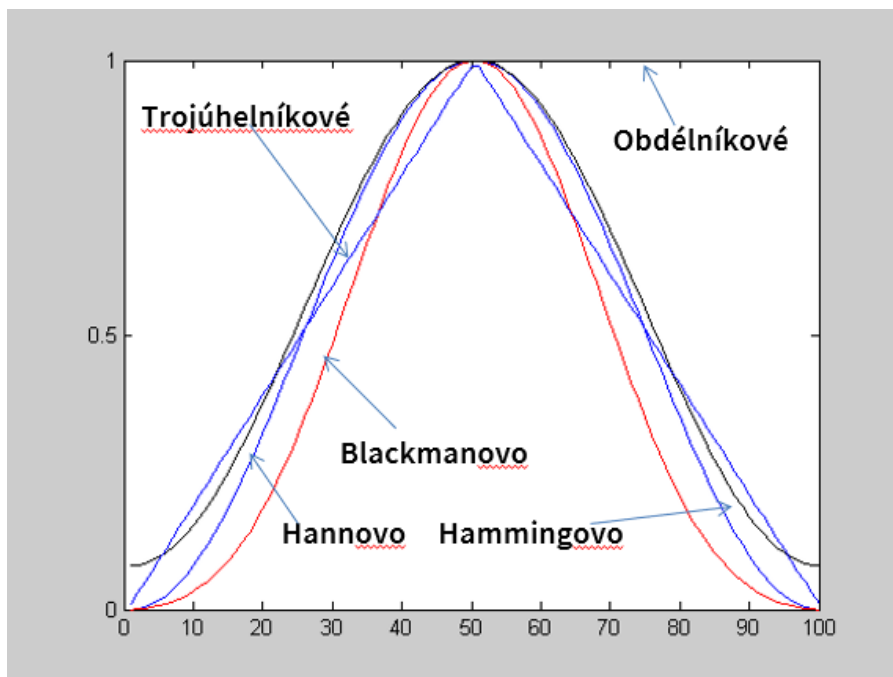
Obrázek 12: Trojúhelníkové okno [32]

- **Blackmanovo okno**



Obrázek 13: Blackmanovo okno [33]

$$w_n = 0.42 - 0.5 \cos\left(\frac{2\pi n}{N-1}\right) + 0.08 \cos\left(\frac{4\pi n}{N-1}\right)$$



Obrázek 14: Srovnání oken

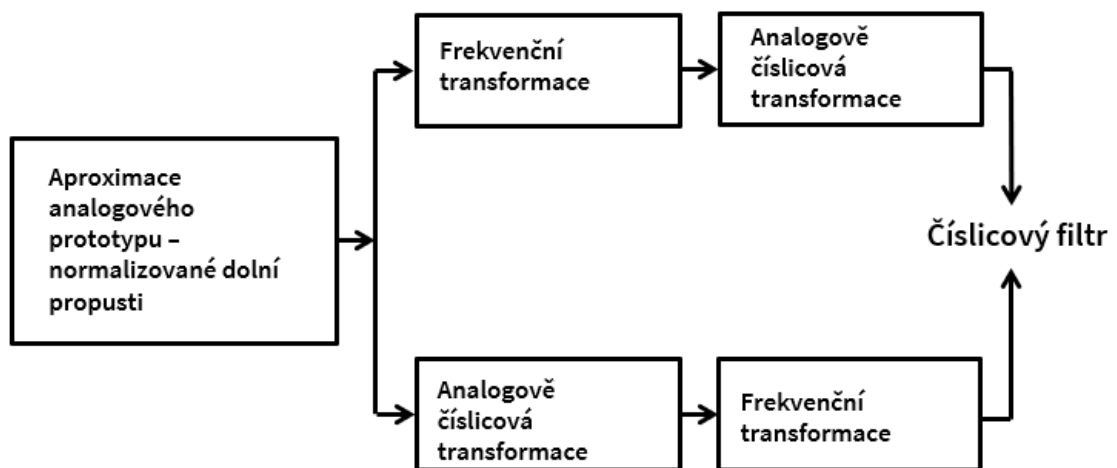
3.2.3 Filtry s nekonečnou impulsní odezvou (IIR)

IIR filtry jsou charakterizované svou zpětnou smyčkou, kde výstupní hodnoty mají vliv na budoucí výstupní hodnoty. Toto způsobuje, že hodnoty výstupního signálu nebudou konečném čase nabývat nulových hodnot, vzniká nám „ozvěna“ způsobená zpětnou smyčkou. [34][35][36]

Metody návrhu:

- Metoda analogových prototypů
- Metoda přímého návrhu v rovině Z.

V práci je použita metoda analogových prototypů a je popsána níže.



Obrázek 15: Schéma návrhu IIR filtru

Metoda je založená na využití takzvané normalizované dolní propusti LP_N .

$$LP_N = \frac{1}{\text{čitatel prototypu}}$$

Příklady analogových prototypů:

- Butterworth
- Bessel
- Chebyshev typ I
- Chebyshev typ II
- Eliptický prototyp

V práci je použit prototyp Butterworth 2. řád. Na něm jsou ukázány i následující výpočty.

Vybereme vhodný analogový prototyp této propusti a podle tabulky nahradíme za s .

1	$(s + 1)$
2	$(s^2 + \sqrt{2}s + 1)$
3	$(s + 1)(s^2 + s + 1)$
4	$(s^2 + \sqrt{2 - \sqrt{2}}s + 1)(s^2 + \sqrt{2 + \sqrt{2}}s + 1)$
5	$(s + 1)(s^2 + \varphi^{-1}s + 1)(s^2 + \varphi s + 1)$
6	$(s^2 + \sqrt{2 - \sqrt{3}}s + 1)(s^2 + \sqrt{2}s + 1)(s^2 + \sqrt{2 + \sqrt{3}}s + 1)$

Obrázek 16: Butterworth prototypy pro jednotlivé řády [37]

Typ frekvenční transformace	Převodní vztahy	Parametry
$LP_N \rightarrow LP$	$s \rightarrow \frac{s}{\omega_p}$	ω_p - mezní úhlová frekvence dolní propusti
$LP_N \rightarrow HP$	$s \rightarrow \frac{\omega_p}{s}$	ω_p - mezní úhlová frekvence horní propusti
$LP_N \rightarrow BP$	$s \rightarrow \frac{s^2 + \omega_0^2}{s\omega_\Delta}$	$\omega_0^2 = \omega_{n1}\omega_{n2}$, $\omega_\Delta = \omega_{n2} - \omega_{n1}$ ω_{n1}, ω_{n2} mezní úhlové frekvence pásmové propusti
$LP_N \rightarrow BS$	$s \rightarrow \frac{s\omega_\Delta}{s^2 + \omega_0^2}$	$\omega_0^2 = \omega_{n1}\omega_{n2}$, $\omega_\Delta = \omega_{n2} - \omega_{n1}$ ω_{n1}, ω_{n2} mezní úhlové frekvence pásmové zadržky

Obrázek 17: Převodní vztahy z normované dolní propusti

$$LP_N = \frac{1}{s^2 + \sqrt{2}s + 1}$$

$$LP = \frac{1}{\left(\frac{s}{\omega_p}\right)^2 + \sqrt{2} \frac{s}{\omega_p} + 1}$$

Vzniklý výraz je nutné upravit na tvar bez složeného zlomku a nahradíme za hodnoty zástupné koeficienty **a** pro čitatel, **b** pro jmenovatel. Dále provedeme Bilineární transformaci pro přechod z roviny **s** do roviny **z**. [38]

$$\frac{a_2 s^2 + a_1 s + a_0}{b_2 s^2 + b_1 s + a_0}$$

Vztah pro bilineární transformaci: [38]

$$s = \frac{2(z - 1)}{T(z + 1)}$$

V rovině z opět upravíme na tvar bez složeného zlomku a spočítáme jednotlivé koeficienty **c** a **d**. Koeficienty **c** jsou v rovnici pro vstupní hodnoty kladné a koeficienty **d** jsou pro výstupní hodnoty záporné.

$$\frac{c_2 z^2 + c_1 z + c_0}{d_2 z^2 + d_1 z + d_0}$$

Normované koeficienty filtru získáme vydělením všech koeficientů koeficientem d_{max} , konkrétně zde d_2 .

Výhody IIR filtru: [24]

- Pro kvalitní řešení stačí malý řád
- Výpočetní náročnost je nízká díky malému řádu

Nevýhody IIR filtru: [24]

- Složitější návrh filtru
- Nemusí být stabilní
- Je rekurzivní

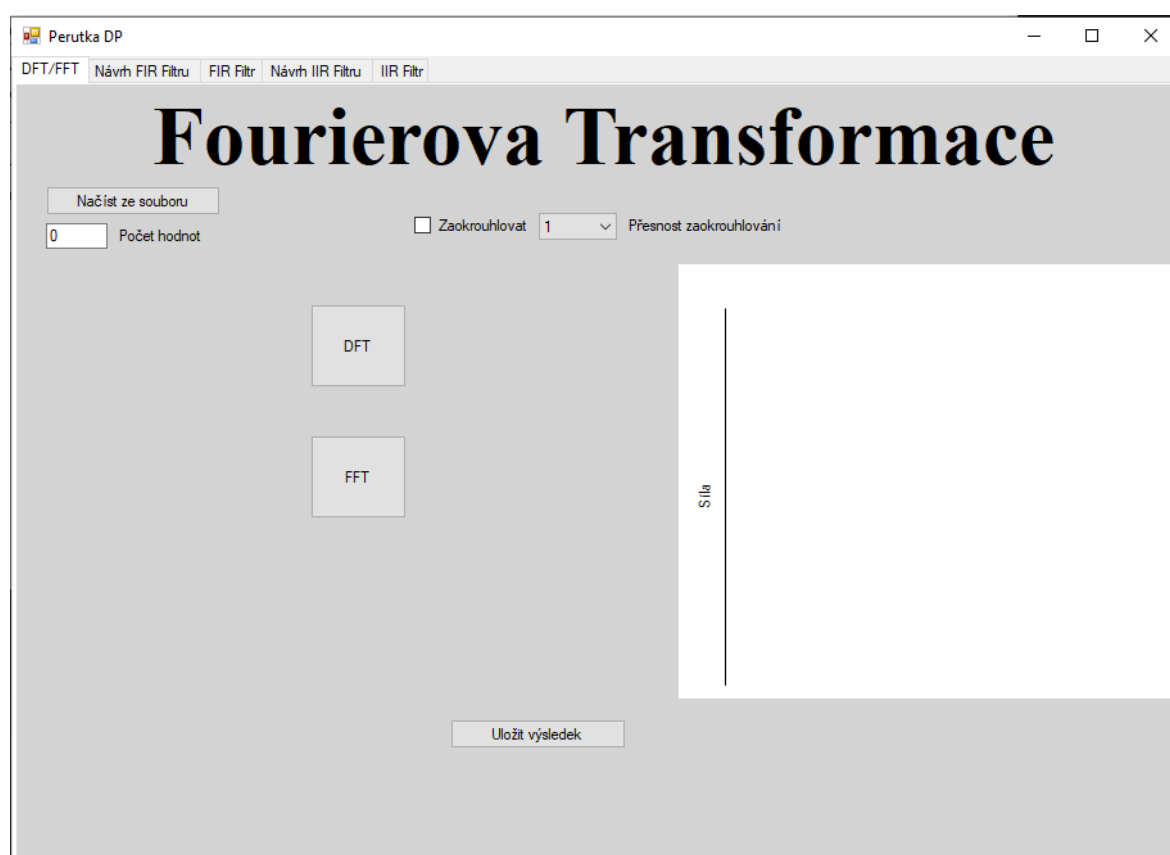
II PRAKTICKÁ ČÁST

4 TAB 1: DISKRÉTNÍ FT A RYCHLÁ FT

První část programu se zabývá Diskrétní Fourierovou transformací reálných hodnot signálu na frekvenční spektrum signálu.

Popis funkce okna programu:

Vstupem jsou hodnoty signálu v časové doméně, tedy klasický číslicový signál. Tento signál je Fourierovou transformací převeden do frekvenční domény a graf zobrazuje signál ve frekvenční doméně. Máme možnost povolit zaokrouhlování na jedno až pět desetinných míst. Výslednou frekvenční charakteristiku můžeme opět uložit do souboru.

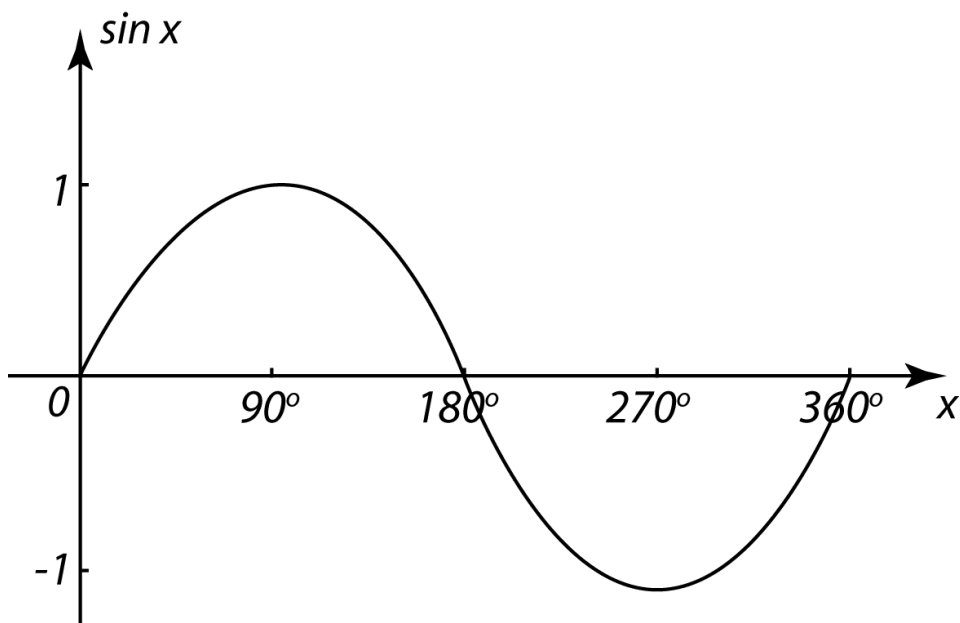


Obrázek 18: Program, Fourierova transformace

Máme dva způsoby pro zadání diskretních vstupních hodnot signálu:

1. Vyplnit hodnoty ručně
2. Naimportovat hodnoty ze souboru

Příklad pro sinusovou funkci:



Obrázek 19: Funkce sinus [39]

Vzorkováním a kvantováním tohoto signálu na osm částí získáme následující hodnoty:

1. 0
2. 0.707
3. 1
4. 0.707
5. 0
6. -0.707
7. -1
8. -0.707

Tyto hodnoty jsou naším vstupem do Fourierovy transformace. Dále si vybereme metodu pro převod do frekvenční domény.

4.1 Metoda diskretní Fourierovy transformace

Diskretní Fourierova transformace probíhá sumou sinusových a kosinusových hodnot násobených velikostí signálu x_n podle vzorce:

$$X_k = \sum_{n=0}^{N-1} x_n * e^{\frac{-j2\pi kn}{N}}$$

Kde:

N je počet hodnot signálu

n je pořadové číslo hodnoty signálu

k je pořadové číslo spektrálního koeficientu

j je indikátor imaginárního čísla (nepočítáme)

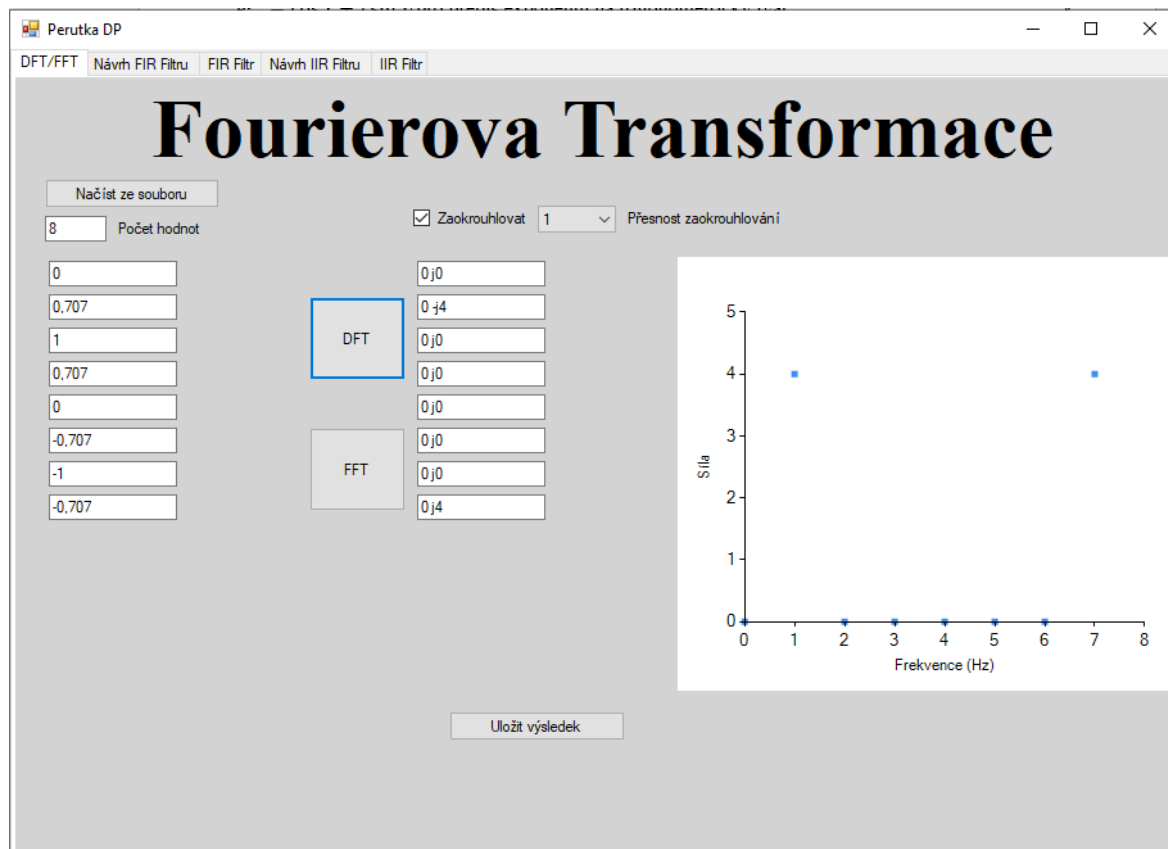
$e^{jx} = \cos x + j \sin x$ pro přepis exponentu na trigonometrický tvar:

$$X_k = \sum_{n=0}^{N-1} x_n * \cos\left(\frac{2\pi kn}{N}\right) + j \sin\left(\frac{-2\pi kn}{N}\right)$$

Kód pro počítání jedné hodnoty podle vzorce výše:

```
for (int i = 0; i <= numberOfCoefficients - 1; i++)
{
    double angle = (-2.0 * Math.PI * (double)targetValue * (double)i)
    / numberOfValues;
    double exponentRealValue = inputValues[i] * Math.Cos(angle);
    double exponentImaginalValue = inputValues[i] * Math.Sin(angle);
    if (checkBoxRoundingAllowed.Checked)
    {
        realSuma += Math.Round(exponentRealValue,
(int)cbboxRoundingTolerance.SelectedValue);
        imaginalSuma += Math.Round(exponentImaginalValue,
(int)cbboxRoundingTolerance.SelectedValue);
    }
    else
    {
        realSuma += exponentRealValue;
        imaginalSuma += exponentImaginalValue;
    }
}
```

Po spočítání všech hodnot máme signál ve frekvenční doméně včetně grafu na obrázku.



Obrázek 20: Program, příklad diskrétní Fourierovy transformace

4.2 Metoda rychlé Fourierovy transformace

Základní vzorec je stále Diskrétní Fourierova transformace, ale bude probíhat dělení sum na sudé a liché koeficienty podle vzorce:

$$X_k = \sum_{n=0}^{N-1} x_n * e^{\frac{-j2\pi kn}{N}}$$

Kde:

N je počet hodnot signálu

n je pořadové číslo hodnoty signálu

k je pořadové číslo spektrálního koeficientu

j je indikátor imaginárního čísla (nepočítáme)

$e^{jx} = \cos x + j \sin x$ pro přepis exponentu na trigonometrický tvar.

Nyní bude probíhat dělení sumy na sudé a liché pořadí hodnot v každé sumě. Sumu potřebujeme rozdělit třikrát, protože máme osm hodnot. Nyní jsme ve stavu suma (0, 1, 2, 3, 4, 5, 6, 7).

$$X_k = \sum_{n=0}^{N-1} x_n * e^{\frac{-j2\pi kn}{N}}$$

První dělení sumy:

$$X_k = \sum_{n=0}^{N/2-1} x_{2n} * e^{\frac{-j2\pi kn}{N}} + e^{\frac{-j2\pi k}{N}} \sum_{n=0}^{N/2-1} x_{2n+1} * e^{\frac{-j2\pi kn}{N}}$$

Nyní jsme ve stavu suma (0, 2, 4, 6) a suma (1, 3, 5, 7).

Druhé dělení sumy:

$$X_k = \sum_{n=0}^{N/4-1} x_{4n} * e^{\frac{-j2\pi kn}{N}} + e^{\frac{-j4\pi k}{N}} \sum_{n=0}^{N/4-1} x_{4n+2} * e^{\frac{-j2\pi kn}{N}} + e^{\frac{-j2\pi k}{N}} \left(\sum_{n=0}^{N/4-1} x_{4n+1} * e^{\frac{-j2\pi kn}{N}} + e^{\frac{-j4\pi k}{N}} \sum_{n=0}^{N/4-1} x_{4n+3} * e^{\frac{-j2\pi kn}{N}} \right)$$

Nyní jsme ve stavu suma (0, 4), suma (2, 6), suma (1, 5), suma (3, 7).

Třetí dělení sumy:

$$X_k = \sum_{n=0}^{N/8-1} x_{8n} * e^{\frac{-j2\pi kn}{N}} + e^{\frac{-j8\pi k}{N}} * \sum_{n=0}^{N/8-1} x_{8n+4} * e^{\frac{-j2\pi kn}{N}} + e^{\frac{-j4\pi k}{N}} * \left(\sum_{n=0}^{N/8-1} x_{8n+2} * e^{\frac{-j2\pi kn}{N}} + e^{\frac{-j8\pi k}{N}} \sum_{n=0}^{N/8-1} x_{8n+6} * e^{\frac{-j2\pi kn}{N}} \right)$$

$$\begin{aligned}
& + e^{\frac{-j2\pi k}{N}} \left(\sum_{n=0}^{\frac{N}{8}-1} x_{8n+1} * e^{\frac{-j2\pi kn}{N}} + e^{\frac{-j8\pi k}{N}} \sum_{n=0}^{\frac{N}{8}-1} x_{8n+5} * e^{\frac{-j2\pi kn}{N}} + e^{\frac{-j4\pi k}{N}} \right. \\
& \quad \left. * \left(\sum_{n=0}^{\frac{N}{8}-1} x_{8n+3} * e^{\frac{-j2\pi kn}{N}} + e^{\frac{-j8\pi k}{N}} \sum_{n=0}^{\frac{N}{8}-1} x_{8n+7} * e^{\frac{-j2\pi kn}{N}} \right) \right)
\end{aligned}$$

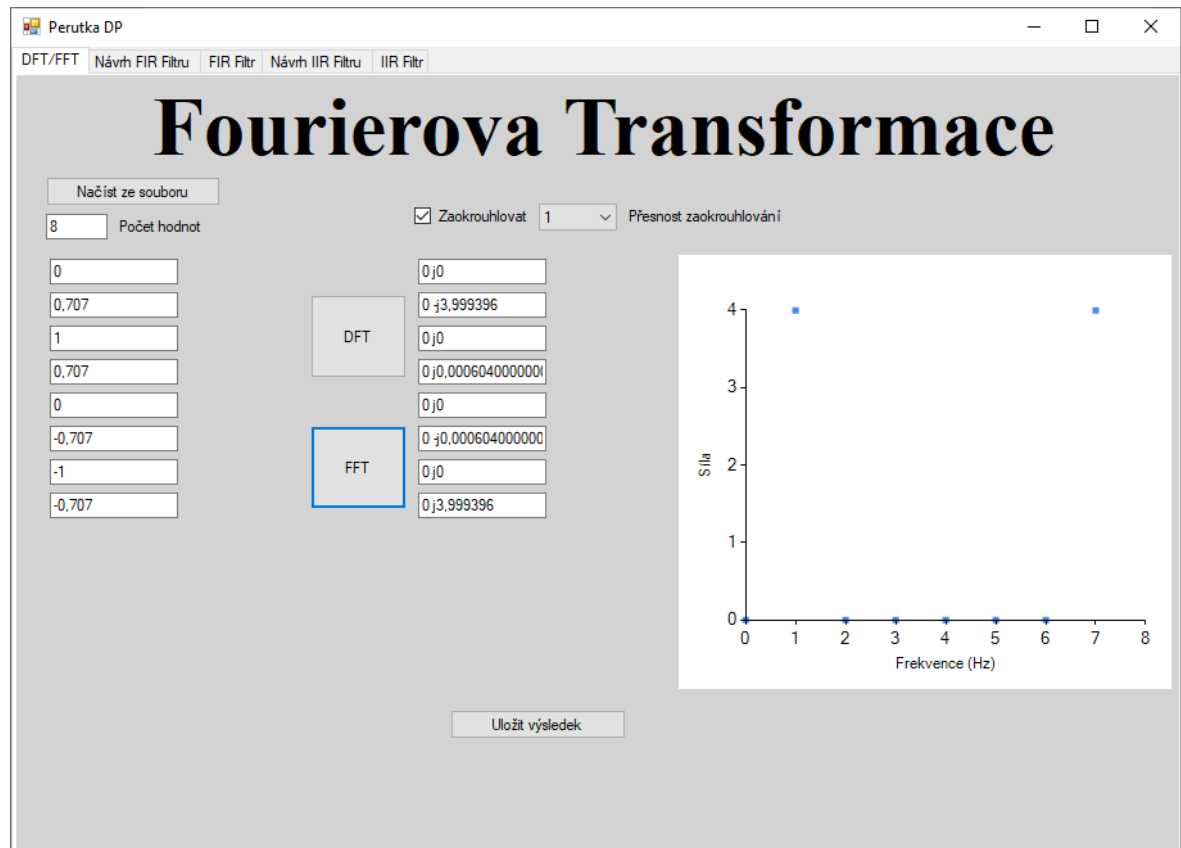
Nyní jsme ve stavu suma (0), suma (4), suma (2), suma (6), suma (1), suma (5), suma (3), suma (7).

V tento moment můžeme celý výraz $e^{\frac{-j2\pi kn}{N}}$ z výpočtu odstranit, protože n bude nabývat vždy hodnoty 0, takže celý exponent je roven 0. Zbude jen násobení koeficienty, které jsme dělením získali.

Zjednodušená konkrétní suma pro tento příklad:

$$\begin{aligned}
X_k &= x_0 + e^{\frac{-j8\pi k}{N}} * x_4 + e^{\frac{-j4\pi k}{N}} * \left(x_2 + e^{\frac{-j8\pi k}{N}} * x_6 \right) \\
&+ e^{\frac{-j2\pi k}{N}} \left(x_1 + e^{\frac{-j8\pi k}{N}} x_5 + e^{\frac{-j4\pi k}{N}} * \left(x_3 + e^{\frac{-j8\pi k}{N}} * x_7 \right) \right)
\end{aligned}$$

Podle této rovnice se spočítají všechny hodnoty a výsledek je vypsán na výstup a zobrazen na grafu. Výsledku jsou stejné jako pro DFT.



Obrázek 21: Program, příklad rychlé Fourierovy transformace

Kód, který vykonává výpočet:

```
if (inputValues.Count == 1)
{
    return inputValues[0];
}
else
{
    List<double> EvenList = new List<double>();
    List<double> OddList = new List<double>();
    for (int i = 0; i < inputValues.Count; i++)
    {
        if (i % 2 == 0)
        {
            EvenList.Add(inputValues[i]);
        }
        else
        {
            OddList.Add(inputValues[i]);
        }
    }
}
```

```
}

    double angle = (-Math.Pow(2, splitOrder) * Math.PI *
(double)targetIndex) / N;
    double cosAngleValue = 0;
    Complex sinAngleValue = 0;

    cosAngleValue = Math.Round(Math.Cos(angle), 3);

    sinAngleValue = Complex.Multiply(new Complex(0, 1),
(double)Math.Round(Math.Sin(angle), 3));

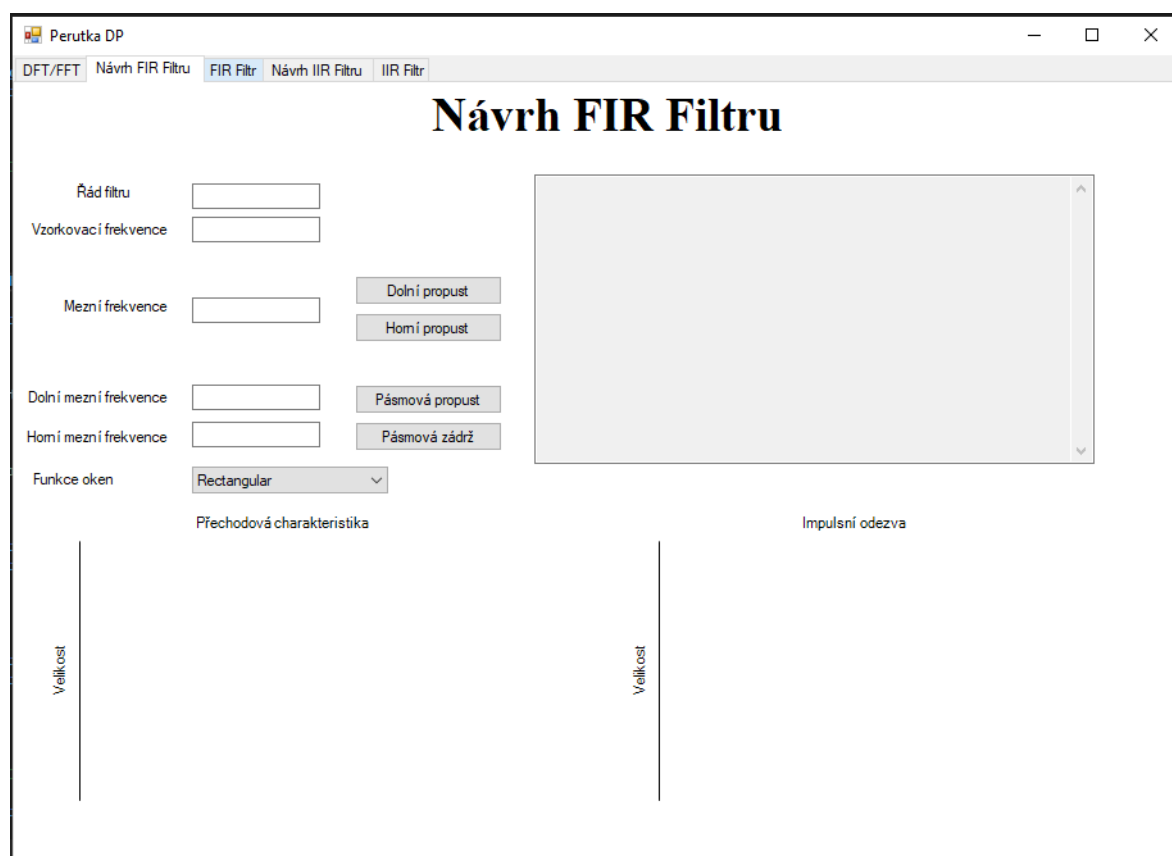
    return Complex.Add(ProcessFFT(EvenList, N, targetIndex, splitOrder
+ 1), (cosAngleValue + sinAngleValue) * ProcessFFT(OddList, N,
targetIndex, splitOrder + 1));
```

5 TAB 2: NÁVRH FIR FILTRU

Další část slouží pro návrh FIR (Finite Impulse Response) filtru, který vytváříme inverzní Fourierovou transformací.

Grafy:

- Přechodová charakteristika – vyjadřuje reakci filtru na posloupnost jedniček
- Impulsní odezva – vyjadřuje reakci filtru na jednu jedničku



Obrázek 22: Program, Návrh FIR filtru

Vyžadovaná vstupní data:

1. Řád filtru – Určuje počet vah a ovlivňuje i samotný výpočet
2. Vzorkovací frekvence – jak často zaznamenáváme hodnotu signálu
3. Mezní frekvence – Pro horní a dolní propust
4. Horní/dolní mezní frekvence – Pro pásmovou zadrž a propust
5. Váhovací funkce (okna) – úprava koeficientů filtru vynásobením odpovídající funkcí okna

Lichý a sudý řád filtru

Kvůli symetrickým vlastnostem se lichý a sudý filtre lehce liší. Při sudém řádu jsou hodnoty symetrické vzhledem k prostřední hodnotě a pro lichý řád je posloupnost koeficientů symetrická hned od nultého koeficientu.

Správný počet hodnot, který budeme počítat je tedy ovlivněn řádem podle tohoto kódu:

```
int k = 0;
if (filterOrder % 2 == 0)
{
    k = filterOrder / 2 + 1;
}
else
{
    k = (filterOrder+1) / 2;
}
```

Kde:

k je počet počítaných hodnot

filterOrder je vyplněný řád filtru

Samotný výpočet pro hodnoty je stejný, na konci se liší symetrie druhé poloviny hodnot.

```
if (filterOrder % 2 == 0)
{
    for (int i = 1; i <= filterOrder / 2; i++)
    {
        finalList.Insert(0, results[i]);
        FinalResultsForChart.Insert(0, resultsForChart[i]);
    }
}
else
{
    for (int i = 0; i <= (filterOrder-1) / 2; i++)
    {
        finalList.Insert(0, results[i]);
        FinalResultsForChart.Insert(0, resultsForChart[i]);
    }
}
```

5.1 Výpočet vah filtru

5.1.1 Horní a dolní propust

Perioda vzorkování:

$$\omega_h = 2 * \pi * \frac{f_c}{f_s}$$

Kde:

f_s je vzorkovací frekvence

f_c je mezní frekvence

Nenultý člen se pro filtr počítá následovně:

$$X_k = \frac{\omega_h}{\pi} * \frac{\sin(\omega_h * k)}{k * \omega_h} * váhaOkna_k$$

Horní propust má opačná znaménka u nenultých členů.

Kde:

X je počítaná váha

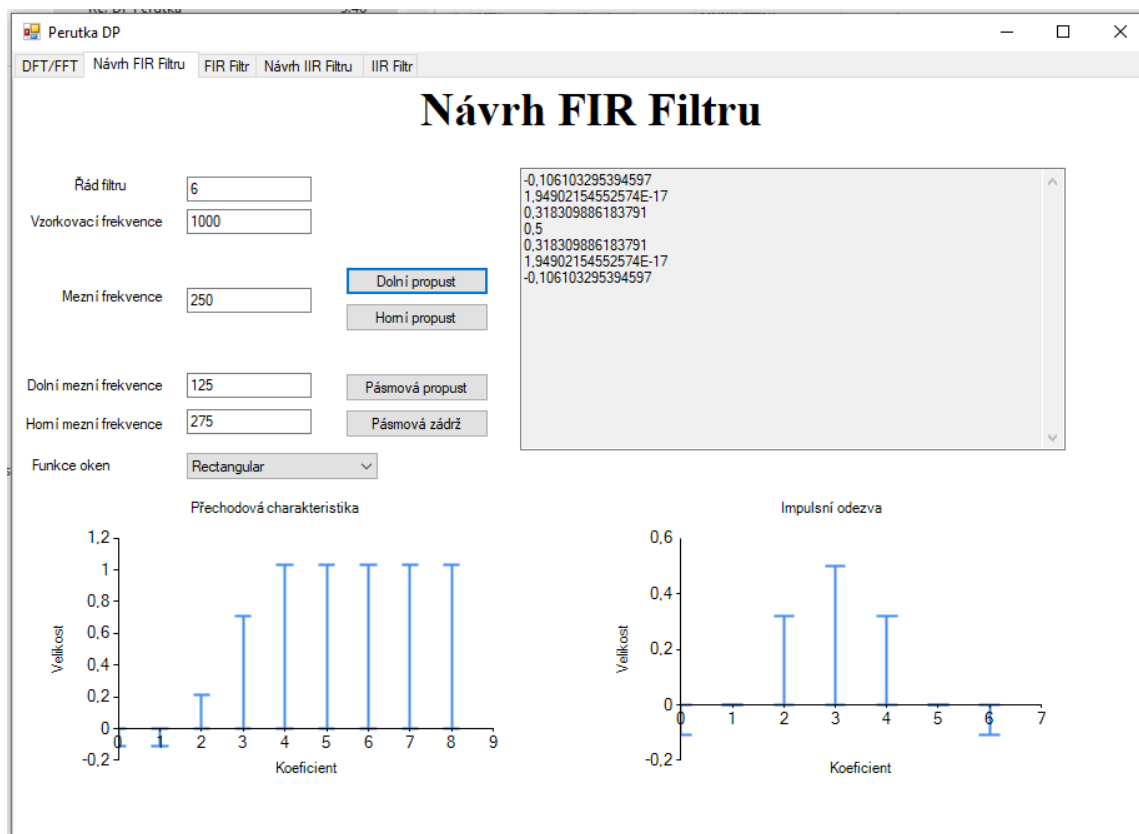
k je pořadové číslo hodnoty

ω_h je perioda vzorkování

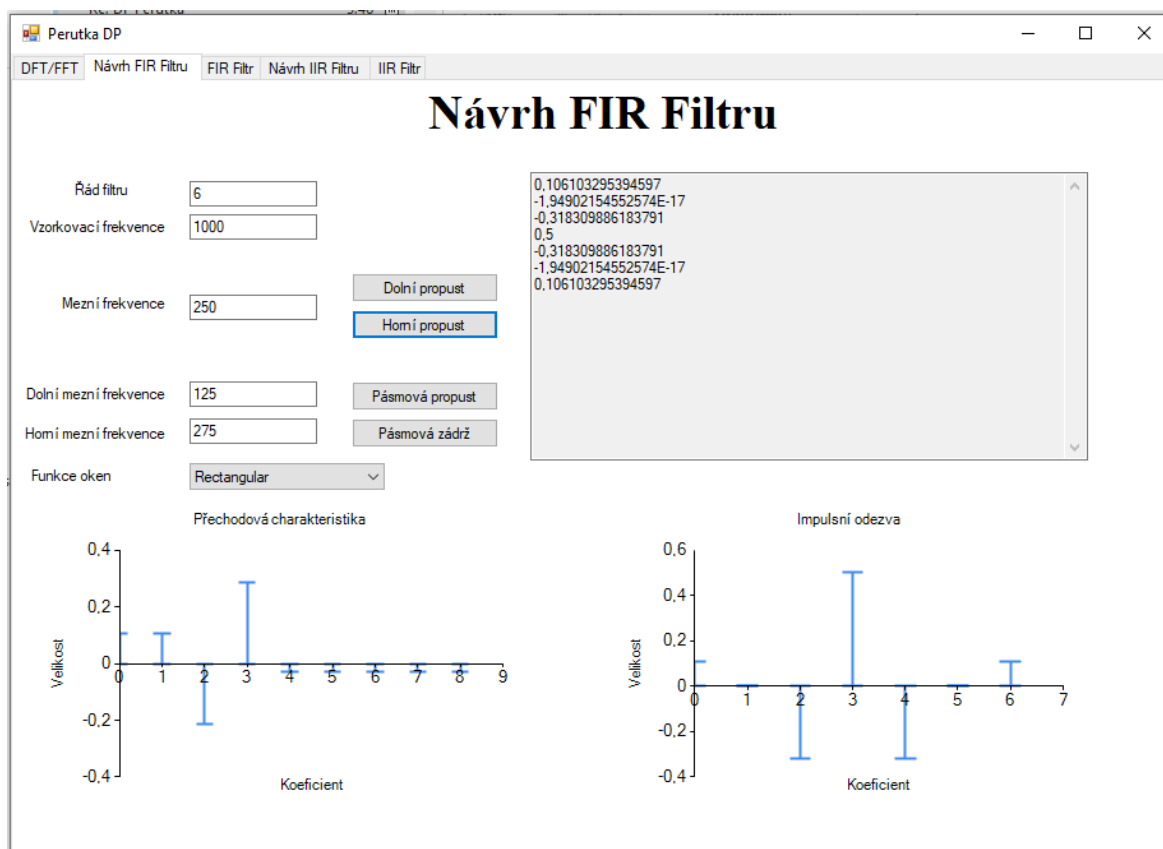
$váhaOkna_k$ je hodnota okna pro tuto hodnotu filtru

Pro nultou hodnotu je $k = 0$, takže nám zbývá: (okno se na nultý člen neaplikuje):

$$X_0 = \frac{\omega_h}{\pi}$$



Obrázek 23: Program, FIR návrh dolní propusti



Obrázek 24: Program, FIR návrh horní propusti

5.1.2 Pásmová zadrž a propust

Perioda vzorkování:

$$\omega_h = 2 * \pi * \frac{f_{c2} - f_{c1}}{f_s}$$

Kde:

f_s je vzorkovací frekvence

f_{c1} je dolní mezní frekvence

f_{c2} je horní mezní frekvence

Nenultý člen se pro filtr počítá následovně:

$$X_k = \left(\frac{1}{\pi * k} * \sin(\omega_{h2} * k) - \frac{1}{\pi * k} * \sin(\omega_{h1} * k) \right) * váhaOkna_k$$

Pásmová zadrž má opačná znaménka u nenultých členů.

Kde:

X je počítaná váha

k je pořadové číslo hodnoty

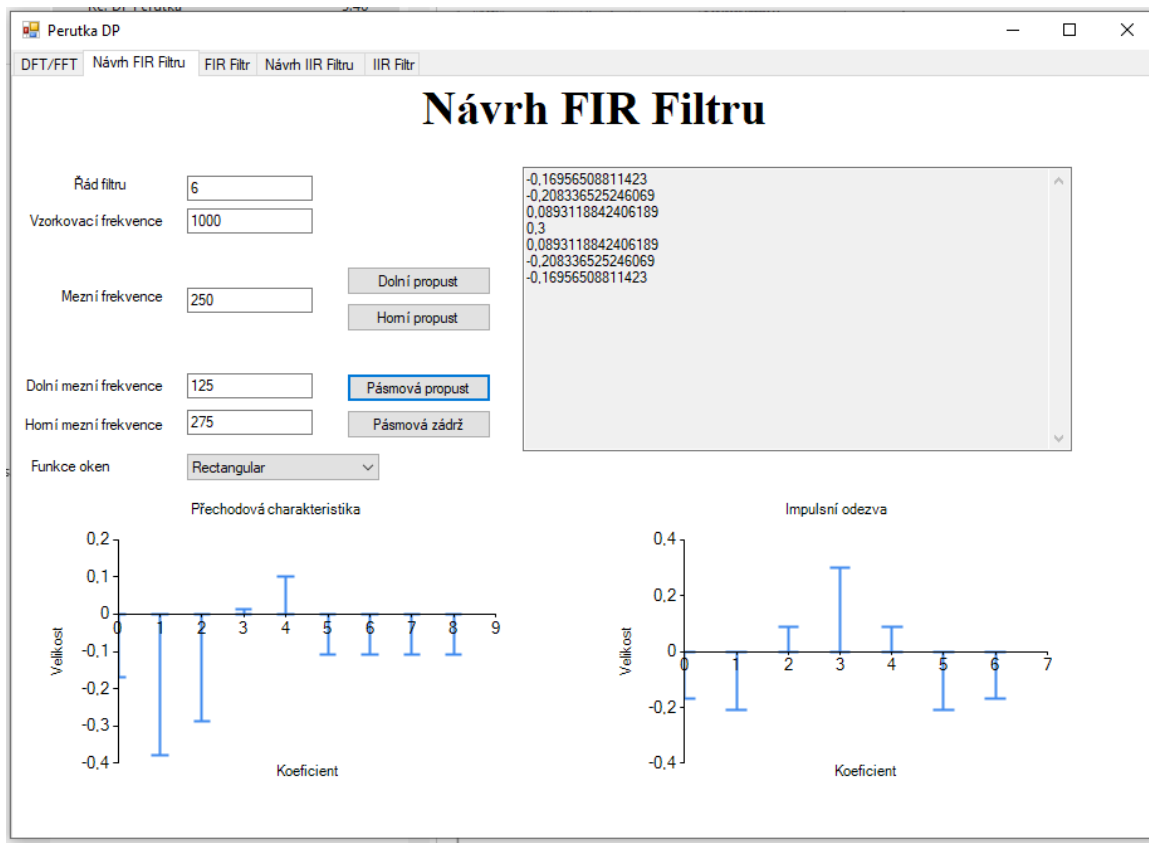
ω_{h1} je perioda vzorkování dolní meze

ω_{h2} je perioda vzorkování horní meze

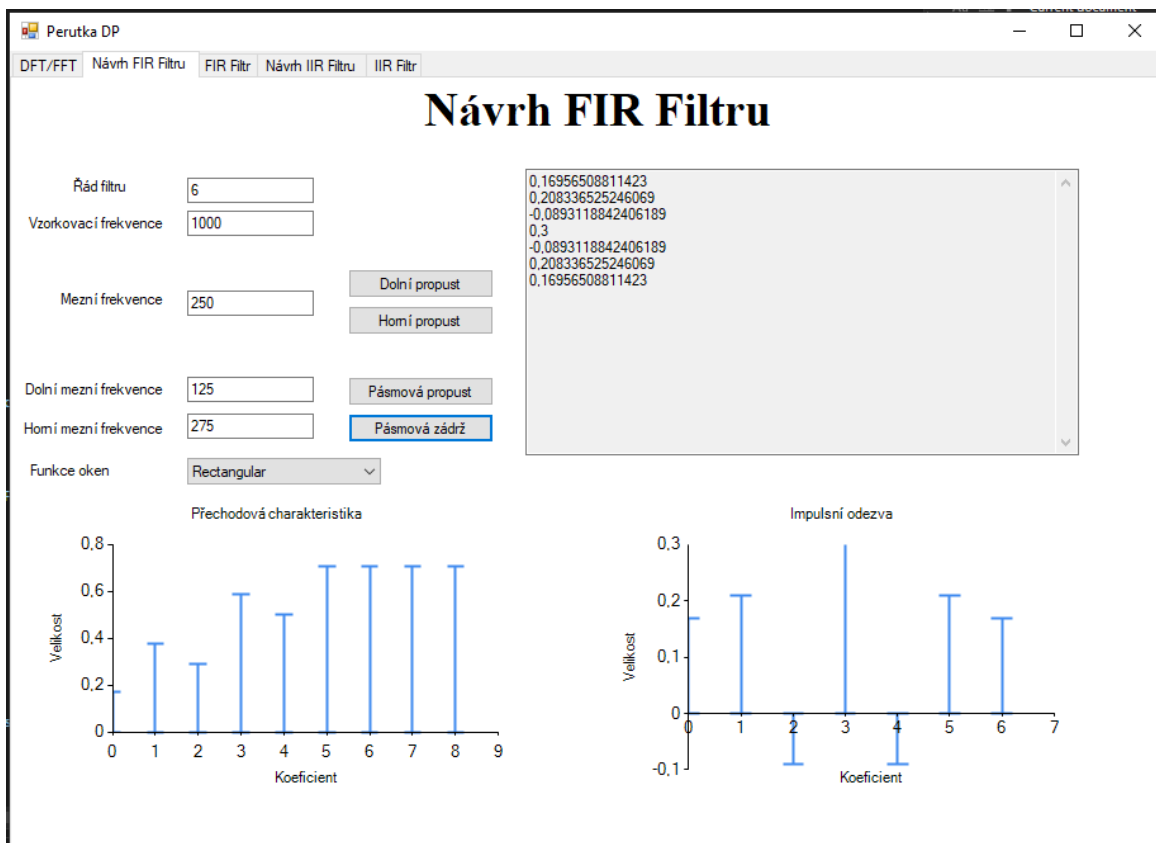
$váhaOkna_k$ je hodnota okna pro tuto hodnotu filtru

Pro nultou hodnotu je $k = 0$, takže nám zbývá: (okno se na nultý člen neaplikuje):

$$X_0 = \frac{\omega_h}{\pi}$$



Obrázek 25: Program, FIR pásmové propusti



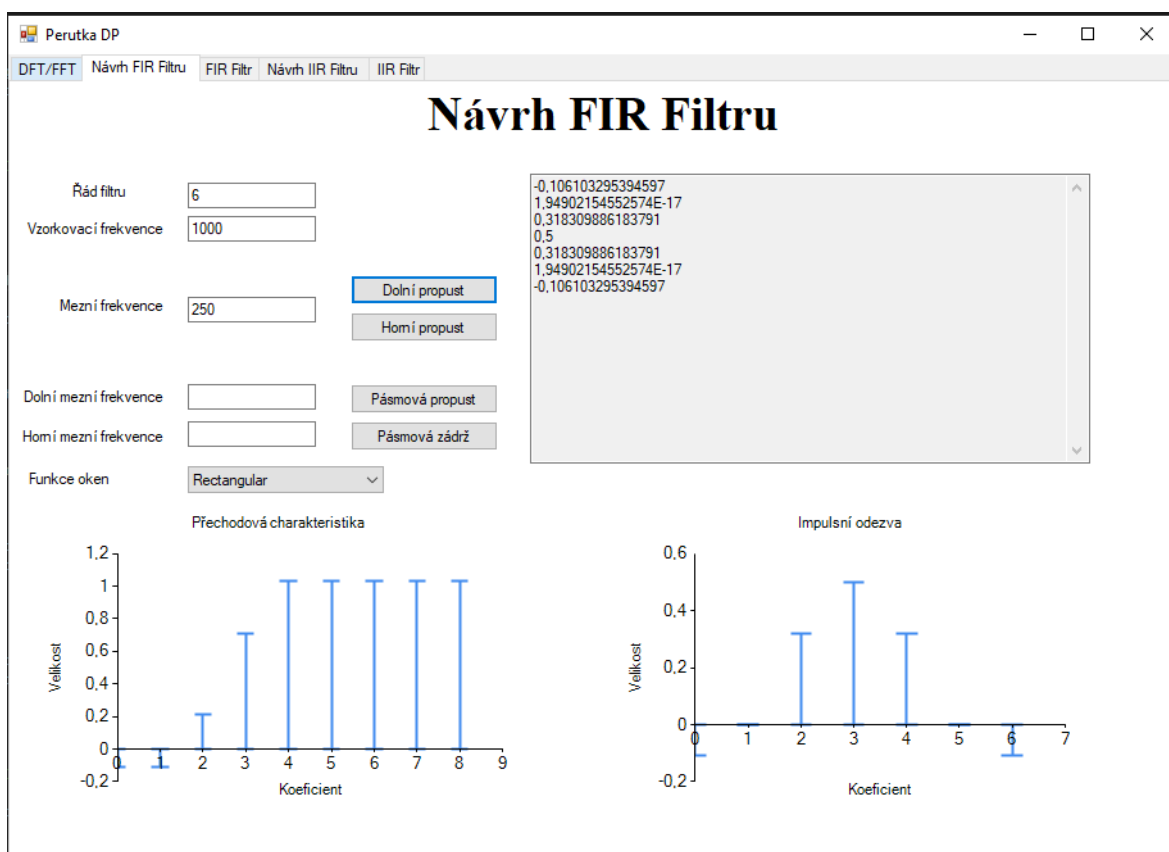
Obrázek 26: Program, FIR pásmové zadrž

5.2 Funkce oken

Okna, jak už bylo řečeno jsou funkce, které nám vrací hodnoty, kterými se násobí odpovídající koeficienty filtru pro zlepšení vlastností filtru. V programu je implementováno pět nejčastějších oken. Přiložený je obrázek z programu, kde je viditelné, jak okna mění hodnoty charakteristik pro filtr se stejnými parametry.

5.2.1 Obdélníkové

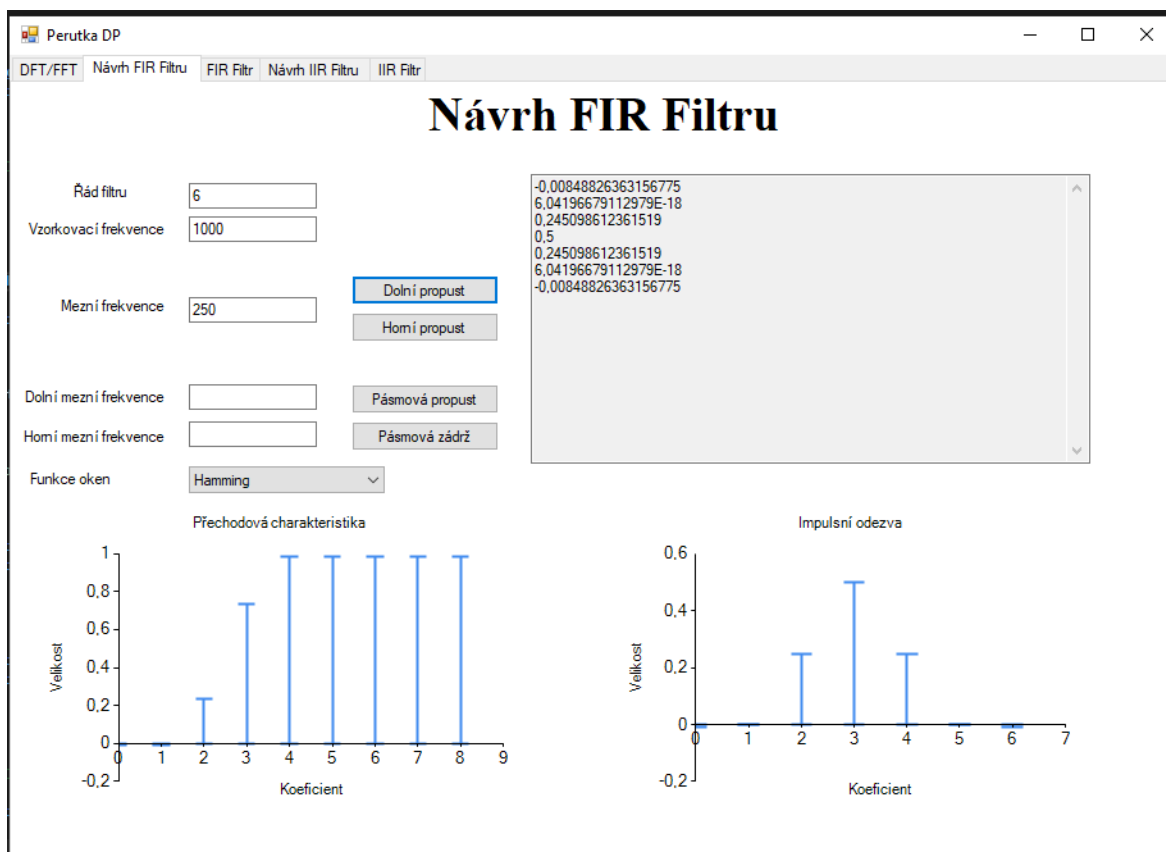
```
for (int n = 0; n < Count; n++)  
{  
    double value = 1;  
    weightList.Add(value);  
}
```



Obrázek 27: Program, příklad s obdélníkovým oknem

5.2.2 Hamming

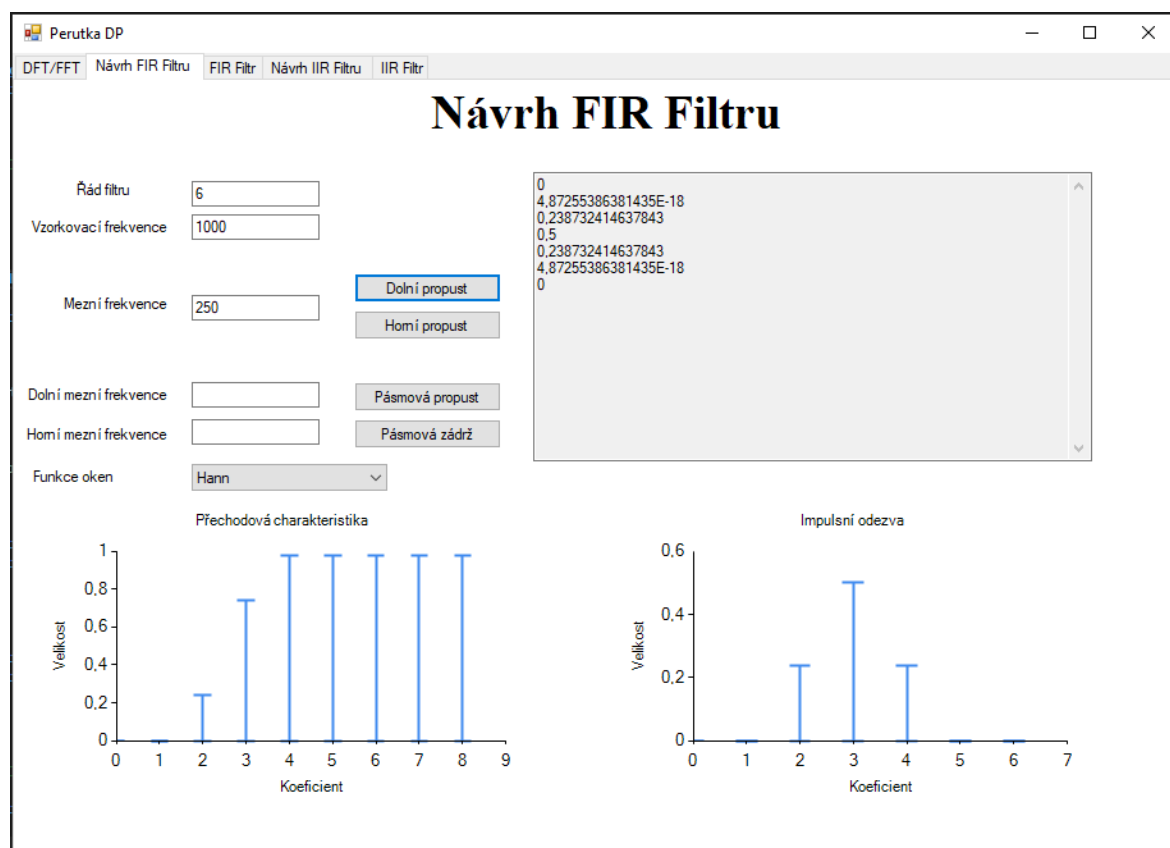
```
for (int n = 0; n < Count; n++)
{
    double value = 0.54 - (0.46*Math.Cos((2 * Math.PI * n) / (Count-1)));
    weightList.Add(value);
}
```



Obrázek 28: Program, příklad s Hammingovým oknem

5.2.3 Hann

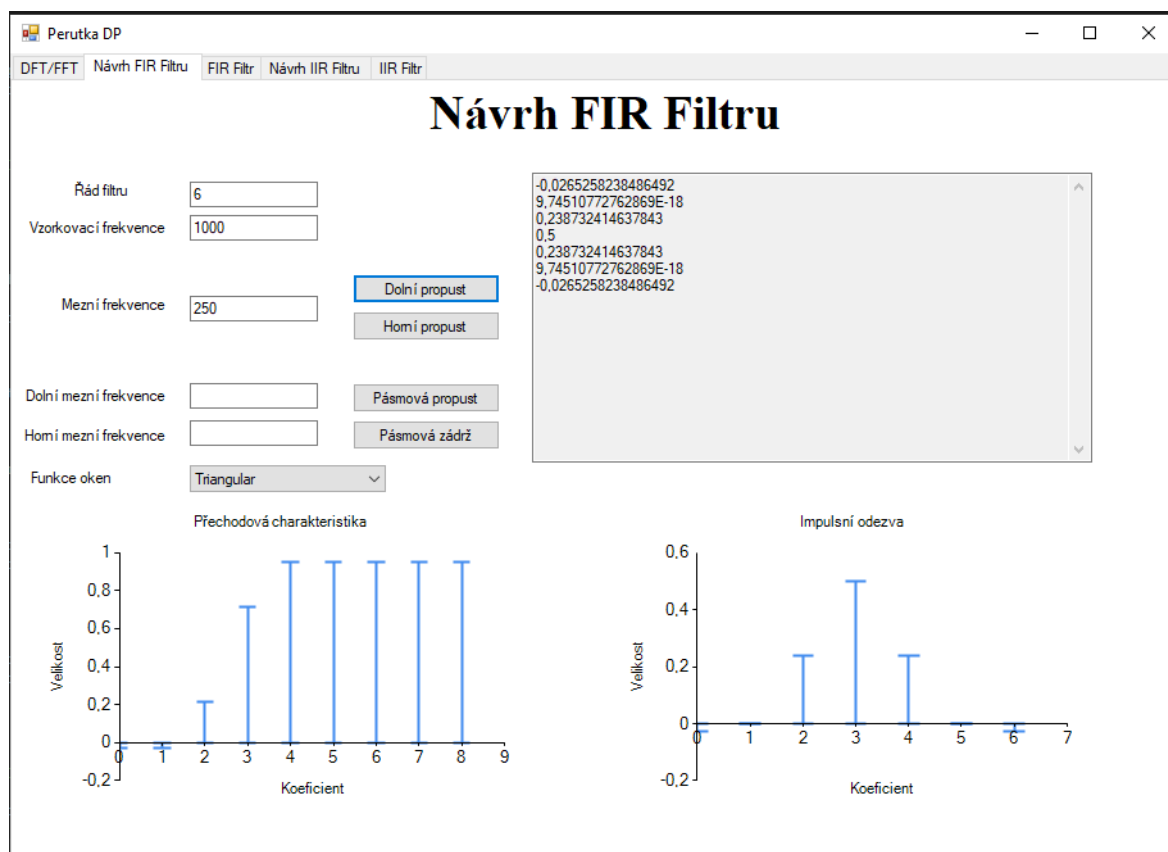
```
for (int n = 0; n < Count; n++)  
{  
    double value = 0.5 * (1 - Math.Cos((2 * Math.PI * n) / (Count -  
    1)));  
    weightList.Add(value);  
}
```



Obrázek 29: Program, příklad s Hannovým oknem

5.2.4 Trojúhelníkové

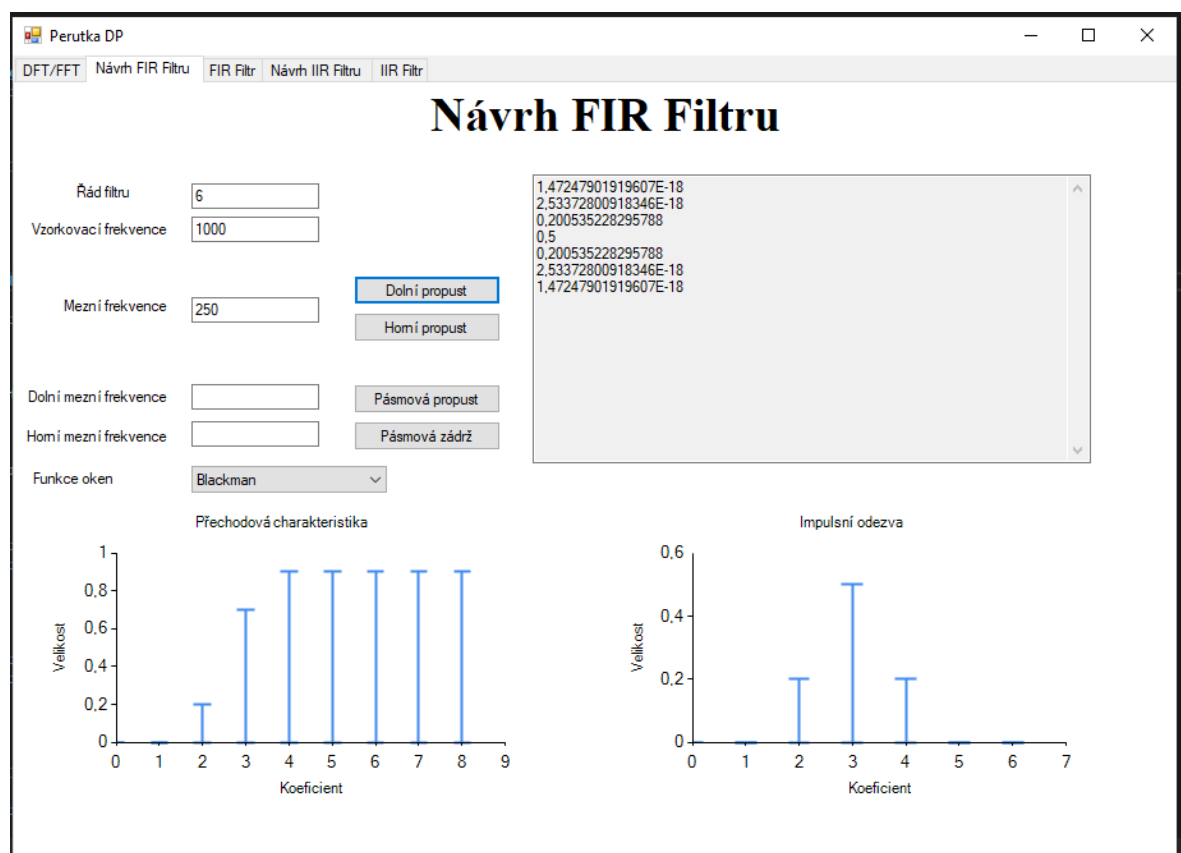
```
for (int n = 0; n < Count; n++)
{
    double value = 1.0 - Math.Abs((n - ((Count - 1.0) / 2.0)) / ((Count + 1) / 2.0));
    weightList.Add(value);
}
```



Obrázek 30: Program, příklad s trojúhelníkovým oknem

5.2.5 Blackman

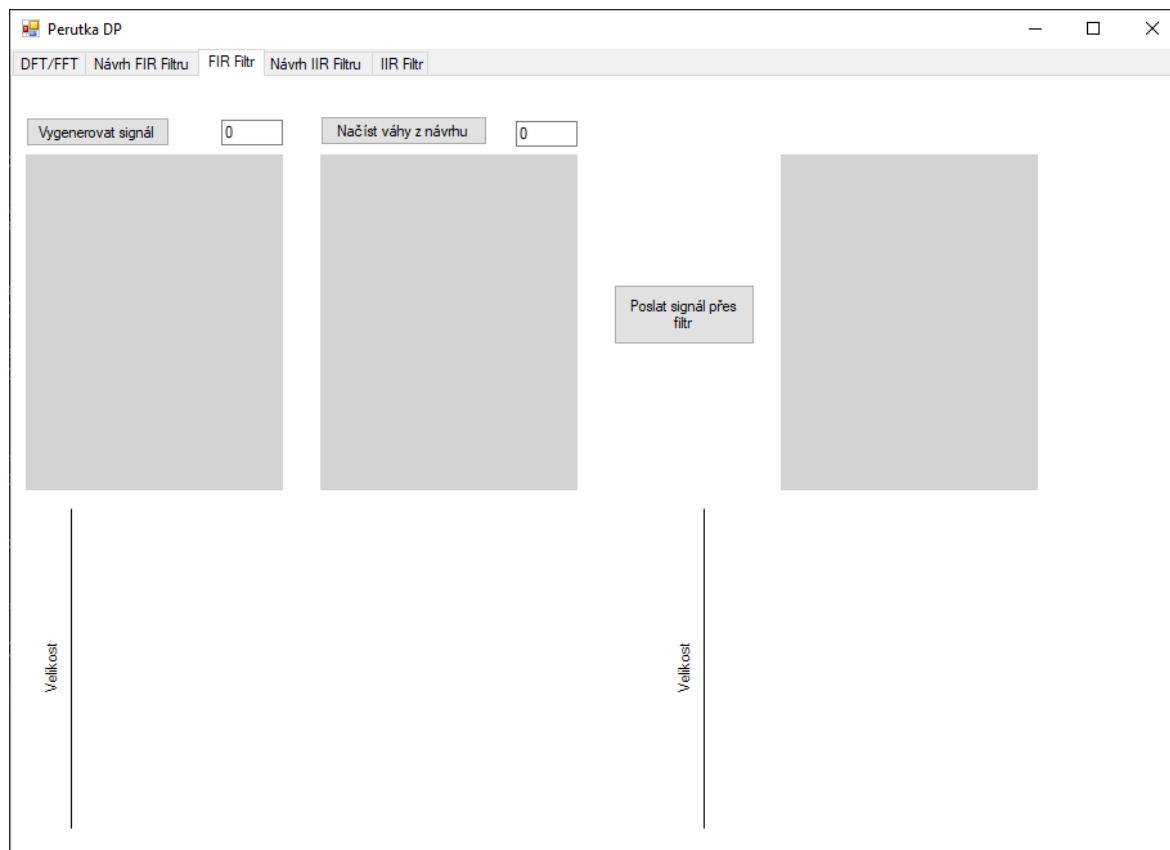
```
for (int n = 0; n < Count; n++)
{
    double cos1 = 0.5 * Math.Cos((2.0 * Math.PI * n) / (Count - 1));
    double cos2 = 0.08 * Math.Cos((4.0 * Math.PI * n) / (Count - 1));
    double value = 0.42 - cos1 + cos2;
    weightList.Add(value);
}
```



Obrázek 31: Program, příklad s Blackmanovým oknem

6 TAB 3 FIR FILTR

Tato část se zabývá samotným použitím FIR filtru pro vstupní signál (filtrací vstupního signálu).

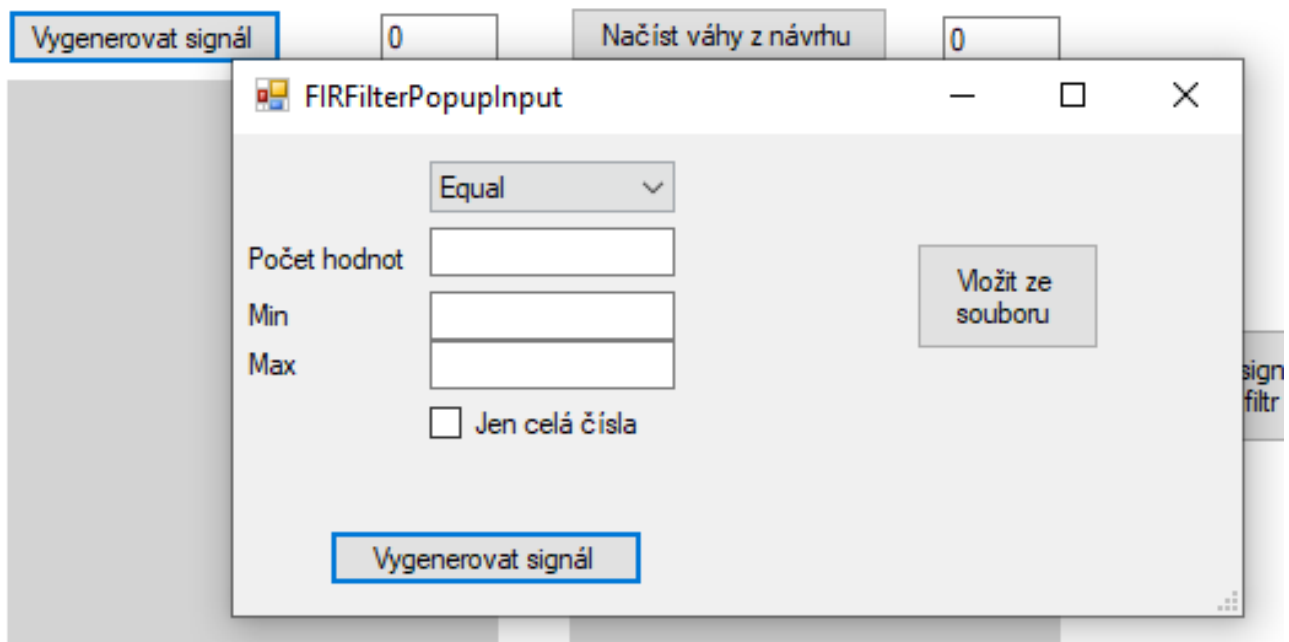


Obrázek 32: Program, FIR filtr

Pro zadání vstupního signálu máme tři možnosti:

1. Zadání počtu a vypsat jednotlivé hodnoty signálu
2. Vygenerovat signál
3. Vložit signál z textového souboru

Váhy je možno buď ručně vyplnit, nebo nainportovat z předchozí části pomocí tlačítka „Načíst váhy z návrhu“.



Obrázek 33: Program, FIR filtr, okno pro generování signálu

Při generování signálu máme na výběr buď normální (Gaussovo) nebo rovnoměrné rozdělení signálu. Také si můžeme vybrat, jestli chceme generovat desetinná čísla nebo jen celá čísla.

Hodnoty výstupního signálu jsou počítány podle následujícího kódu:

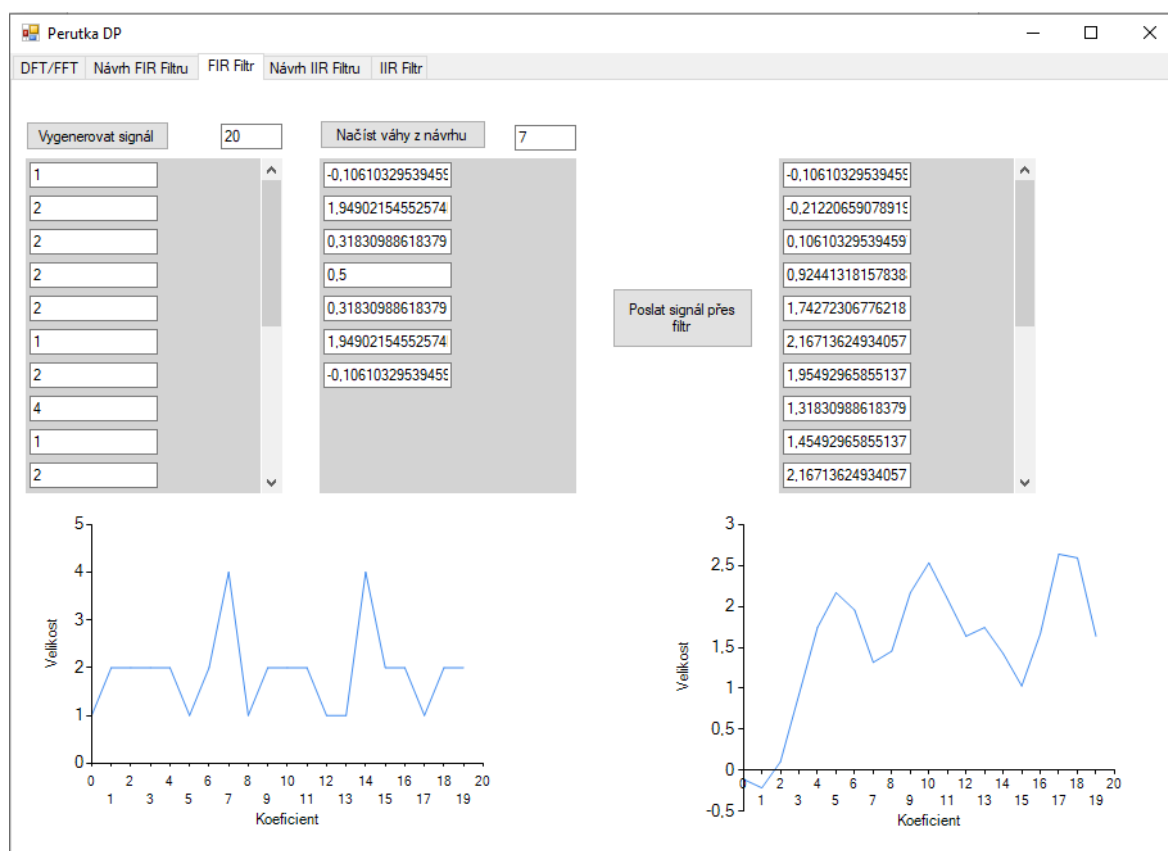
```
double result = 0;
for (int i = 0; i <= filterOrder; i++)
{
    if (index - i < 0)
    { }
    else
    {
        result += WeightsList[i] * InputList[index - i];
    }
}
```

6.1.1 Příklad FIR Filtrace

Bylo vygenerováno 20 hodnot od 0 do 5. Filtr byl sestaven s parametry:

- Dolní propust
- Vzorkovací frekvence 1000 Hz
- Mezní frekvence 250 Hz
- Funkce okna **Rectangular**

Grafy vizualizují vstupní a výstupní signál při průchodu filtrem.



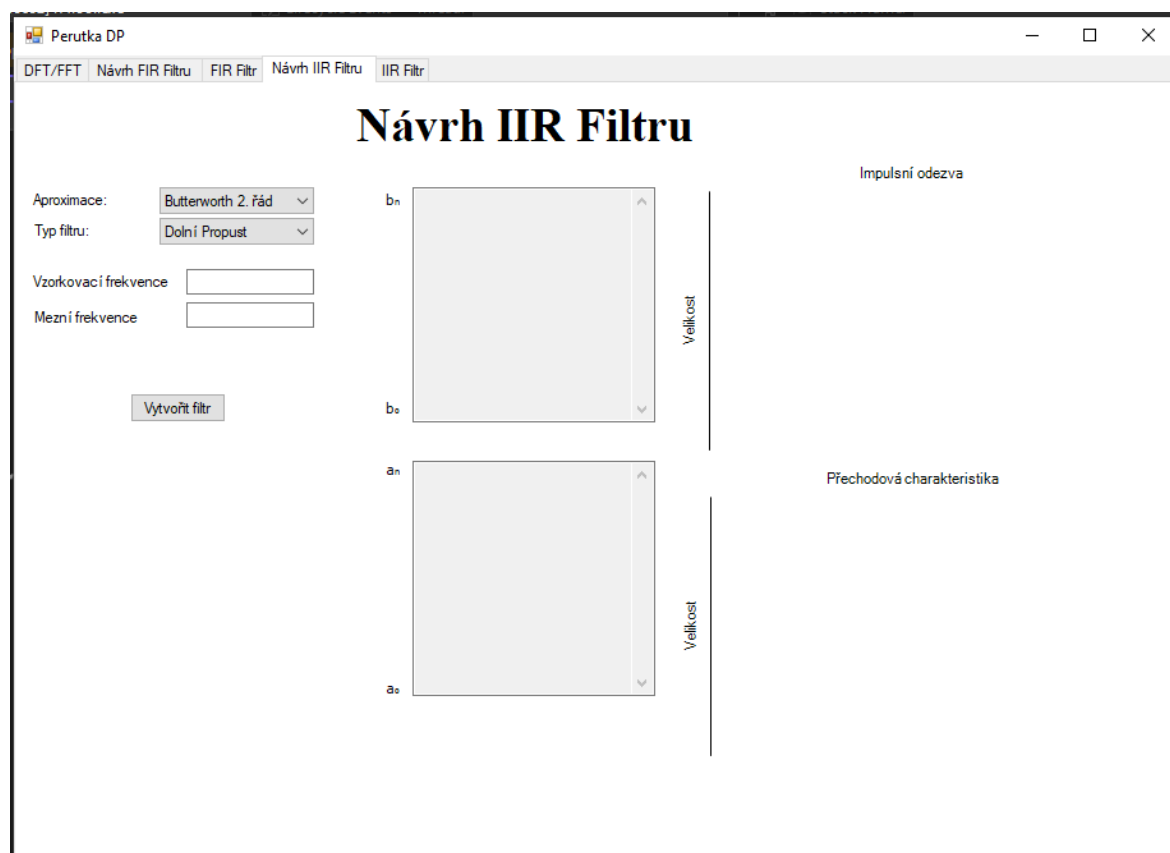
Obrázek 34: Program, příklad FIR filtrace

7 TAB 4 NÁVRH IIR FILTRU

Pro návrh IIR filtru byl realizován analogový prototyp Butterworth 2. řádu, ze kterého byly odvozeny vzorce pro všechny čtyři základní typy filtru – dolní propust, horní propust, pásmovou propust a pásmovou zádrž.

Grafy:

- Přechodová charakteristika – vyjadřuje reakci filtru na posloupnost jedniček
- Impulsní odezva – vyjadřuje reakci filtru na jednu jedničku



Obrázek 35: Program, návrh IIR filtru

7.1 Analogový prototyp Butterworth 2. řád

Normovaná dolní propust je definována následující přenosovou funkcí:

$$\frac{1}{(s^2 + \sqrt{2}s + 1)}$$

Podle následujících vztahů nahradíme za s a spočítáme analogové a digitální koeficienty pro vzorce pro všechny čtyři typy filtru.

Typ frekvenční transformace	Převodní vztahy	Parametry
$LP_N \rightarrow LP$	$s \rightarrow \frac{s}{\omega_p}$	ω_p - mezní úhlová frekvence dolní propusti
$LP_N \rightarrow HP$	$s \rightarrow \frac{\omega_p}{s}$	ω_p - mezní úhlová frekvence horní propusti
$LP_N \rightarrow BP$	$s \rightarrow \frac{s^2 + \omega_0^2}{s\omega_\Delta}$	$\omega_0^2 = \omega_{n1}\omega_{n2}$, $\omega_\Delta = \omega_{n2} - \omega_{n1}$ ω_{n1}, ω_{n2} mezní úhlové frekvence pásmové propusti
$LP_N \rightarrow BS$	$s \rightarrow \frac{s\omega_\Delta}{s^2 + \omega_0^2}$	$\omega_0^2 = \omega_{n1}\omega_{n2}$, $\omega_\Delta = \omega_{n2} - \omega_{n1}$ ω_{n1}, ω_{n2} mezní úhlové frekvence pásmové zadržky

Obrázek 36: Převodní vztahy z normované dolní propusti

7.2 Dolní propust

$$G_{a(s)} = \frac{1}{s^2 + \sqrt{2}s + 1}$$

$$G_{(s)} = \frac{1}{\frac{s^2}{\omega_p^2} + \sqrt{2} \frac{s}{\omega_p} + 1} = \frac{1}{\frac{1}{\omega_p^2} * (s^2 + \sqrt{2}\omega_p s + \omega_p^2)} = \frac{\omega_p^2}{s^2 + \sqrt{2}\omega_p s + \omega_p^2}$$

Analogové koeficienty:

Obecný tvar:

$$\frac{a_2 s^2 + a_1 s + a_0}{b_2 s^2 + b_1 s + a_0}$$

Z upraveného výrazu výše získáváme následující koeficienty:

$$a_0 = \omega_p * \omega_p$$

$$a_1 = 0$$

$$a_2 = 0$$

$$b_0 = \omega_p * \omega_p$$

$$b_1 = \sqrt{2} * \omega_p$$

$$b_2 = 1$$

Bilineární transformace pro převod z roviny s do roviny z podle vztahu:

$$s = \frac{2(z-1)}{T(z+1)}$$

Digitální koeficienty:

$$\begin{aligned} \frac{a_0}{b_2 \frac{4}{T^2} \left(\frac{z-1}{z+1}\right)^2 + b_1 \frac{2}{T} \frac{z-1}{z+1} + b_0} &= \frac{a_0}{b_2 \frac{4}{T^2} \frac{z^2 - z + 1}{(z+1)^2} + b_1 \frac{2}{T} \frac{z-1}{z+1} + b_0} \\ &= \frac{1}{\frac{1}{T^2(z+1)^2} * (4b_2 * (z^2 - z + 1) + 2b_1 * (z-1) + b_0 T^2 (z+1)^2)} \\ &= \frac{a_0 T^2 (z+1)^2}{4b_2 * (z^2 - z + 1) + 2b_1 * (z-1) + b_0 T^2 (z+1)^2} \\ &= \frac{a_0 T^2 z^2 + 2a_0 T^2 z + a_0 T^2}{z^2 * (4b_2 + 2Tb_1 + b_0 T^2) + z * (-8b_2 + 2b_0 T^2) + 4b_2 - 2Tb_1 + T^2 b_0} \end{aligned}$$

Obecný tvar:

$$\frac{c_2 z^2 + c_1 z + c_0}{d_2 z^2 + d_1 z + d_0}$$

Po bilineární transformaci a úpravě jsou výsledné digitální koeficienty:

$$c_0 = a_0 * T^2$$

$$c_1 = 2a_0 * T^2$$

$$c_2 = a_0 * T^2$$

$$d_0 = 4b_2 - 2b_1 T + b_0 T^2$$

$$d_1 = -8b_2 + 2b_0 T^2$$

$$d_2 = 4b_2 + 2b_1 T + b_0 T^2$$

Tyto koeficienty jsou v programu jako vzorce při výpočtu charakteristiky konkrétního filtru.

Pro výpočty je potřeba dopočítat zkreslenou úhlovou frekvenci ω_{an} , která nahrazuje ω_p ve výpočtech podle vztahu:

$$\omega_{an} = \frac{2}{T} * \operatorname{tg}\left(\frac{2 * \pi * f_c * T}{2}\right)$$

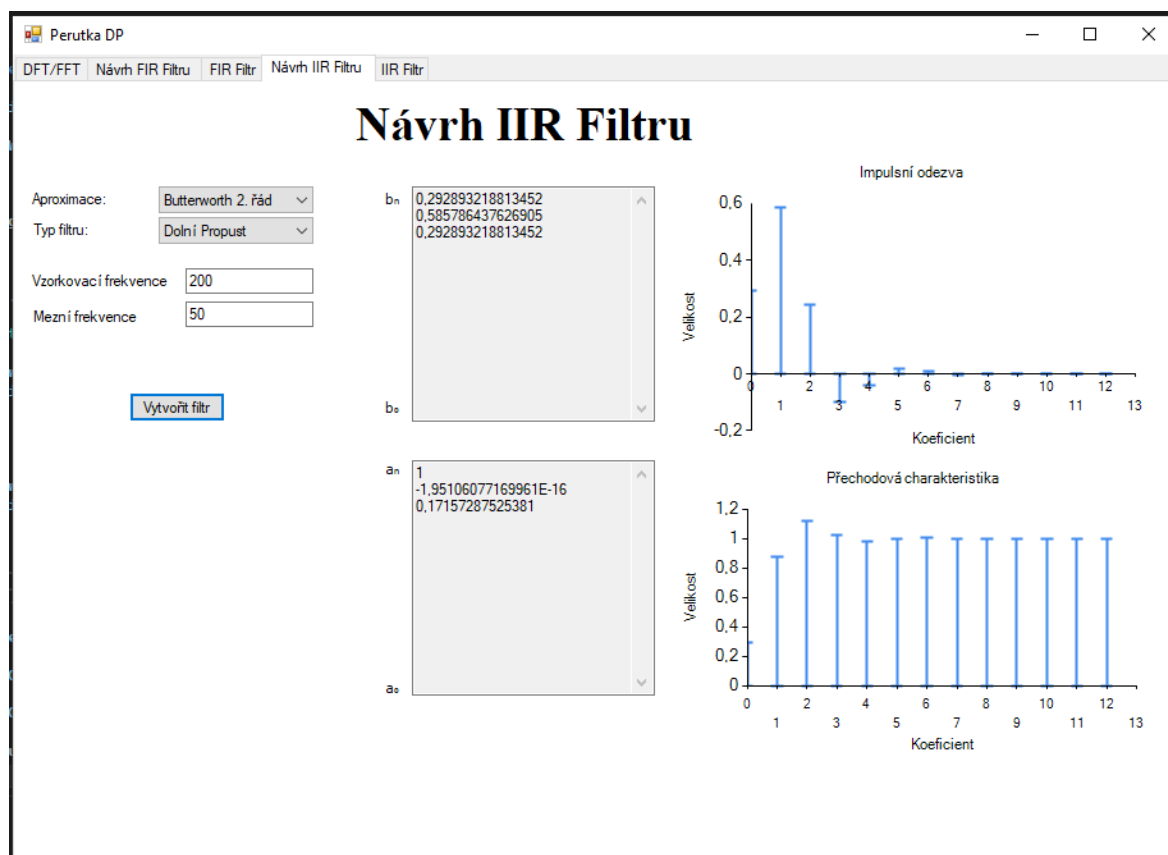
Kde:

ω_{an} je zkreslená úhlová rychlost

T je perioda signálu $\frac{1}{f_{vz}}$

f_c je mezní frekvence

f_{vz} je vzorkovací frekvence



Obrázek 37: Program, návrh IIR dolní propusti

7.3 Horní propust

Pro horní propust použijeme souměrnost filtru, kde koeficienty horní propusti mají stejnou hodnotu, jako u dolní propusti, jen prostřední člen c_1 je záporný:

$$c_0 = a_0 * T^2$$

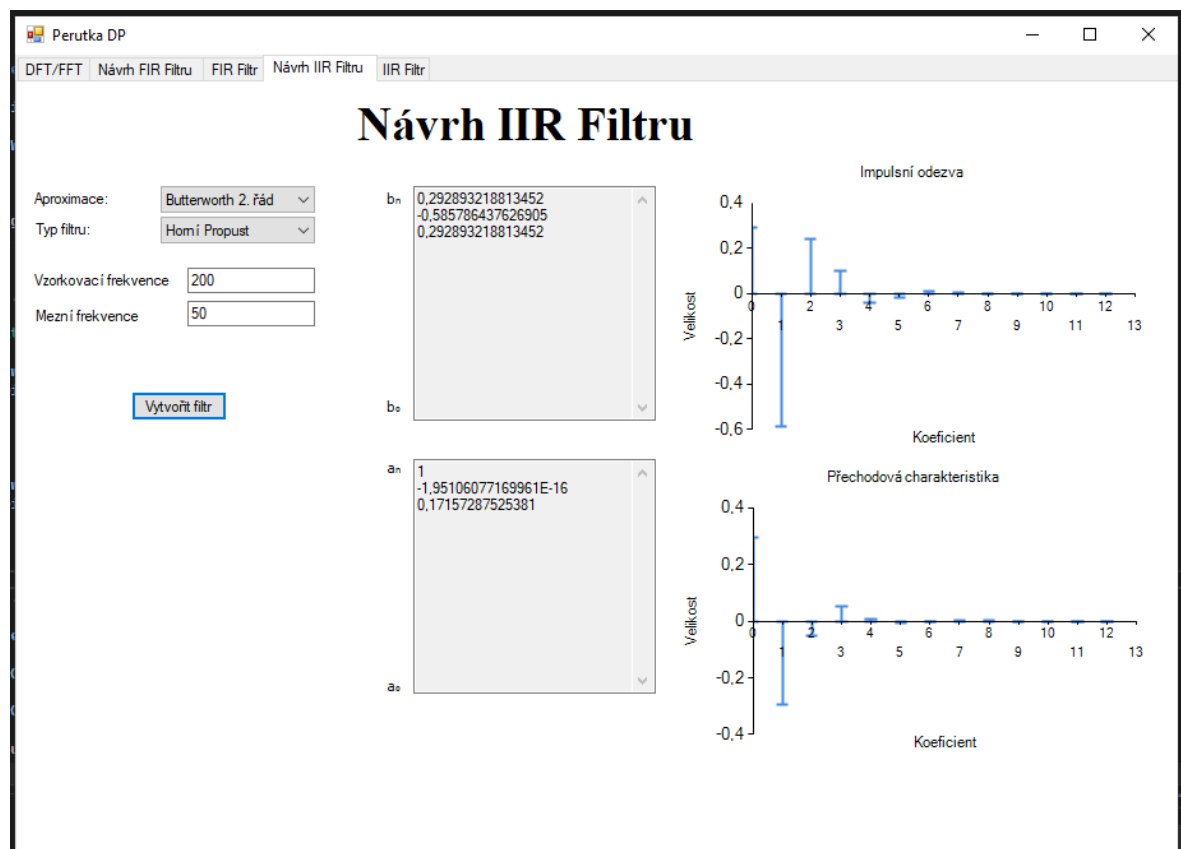
$$-c_1 = 2a_0 * T^2$$

$$c_2 = a_0 * T^2$$

$$d_0 = 4b_2 - 2b_1T + b_0T^2$$

$$d_1 = -8b_2 + 2b_0T^2$$

$$d_2 = 4b_2 + 2b_1T + b_0T^2$$



Obrázek 38: Program, návrh IIR horní propusti

7.4 Pásmová zadrž

Pro výpočty je potřeba dopočítat zkreslené hraniční úhlové rychlosti podle vztahu:

$$\omega_{an1} = \frac{2}{T} * tg\left(\frac{2 * \pi * f_{c1} * T}{2}\right)$$

$$\omega_{an2} = \frac{2}{T} * tg\left(\frac{2 * \pi * f_{c2} * T}{2}\right)$$

Kde:

ω_{an1} je zkreslená úhlová rychlost dolní meze

ω_{an2} je zkreslená úhlová rychlost horní meze

T je perioda signálu $\frac{1}{f_{vz}}$

f_{c1} je mezní frekvence dolní meze

f_{c2} je mezní frekvence horní meze

f_{vz} je vzorkovací frekvence

Z těchto úhlových rychlostí spočítáme:

$$\omega_{\Delta} = \omega_{an2} - \omega_{an1}$$

$$\omega_0^2 = \omega_{an2} * \omega_{an1}$$

Tyto hodnoty použijeme ve výpočtech koeficientů.

Analogové koeficienty:

Obecný tvar:

$$\frac{a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0}{b_4 s^4 + b_3 s^3 + b_2 s^2 + b_1 s + a_0}$$

Po výpočtech jsou výsledné analogové koeficienty:

$$a_0 = \omega_0^4$$

$$a_1 = 0$$

$$a_2 = 2\omega_0^2$$

$$a_3 = 0$$

$$a_4 = 1$$

$$b_0 = \omega_0^4$$

$$b_1 = \sqrt{2}\omega_\Delta\omega_0^2$$

$$b_2 = \omega_\Delta^2 + 2\omega_0^2$$

$$b_3 = \sqrt{2}\omega_\Delta$$

$$b_4 = 1$$

Digitální koeficienty:

Obecný tvar:

$$\frac{c_4z^4 + c_3z^3 + c_2z^2 + c_1z + c_0}{d_4z^4 + d_3z^3 + d_2z^2 + d_1z + d_0}$$

Po bilineární transformaci a úpravě jsou výsledné digitální koeficienty:

$$c_0 = 16 + 4a_2T^2 + a_0T^4$$

$$c_1 = -64 + 4a_0T^4$$

$$c_2 = 96 - 8a_2T^2 + 6a_0T^4$$

$$c_3 = -64 + 4a_0T^4$$

$$c_4 = 16 + 4a_2T^2 + a_0T^4$$

$$d_0 = 16 - 8b_3T + 4b_2T^2 - 2b_1T^3 + b_0T^4$$

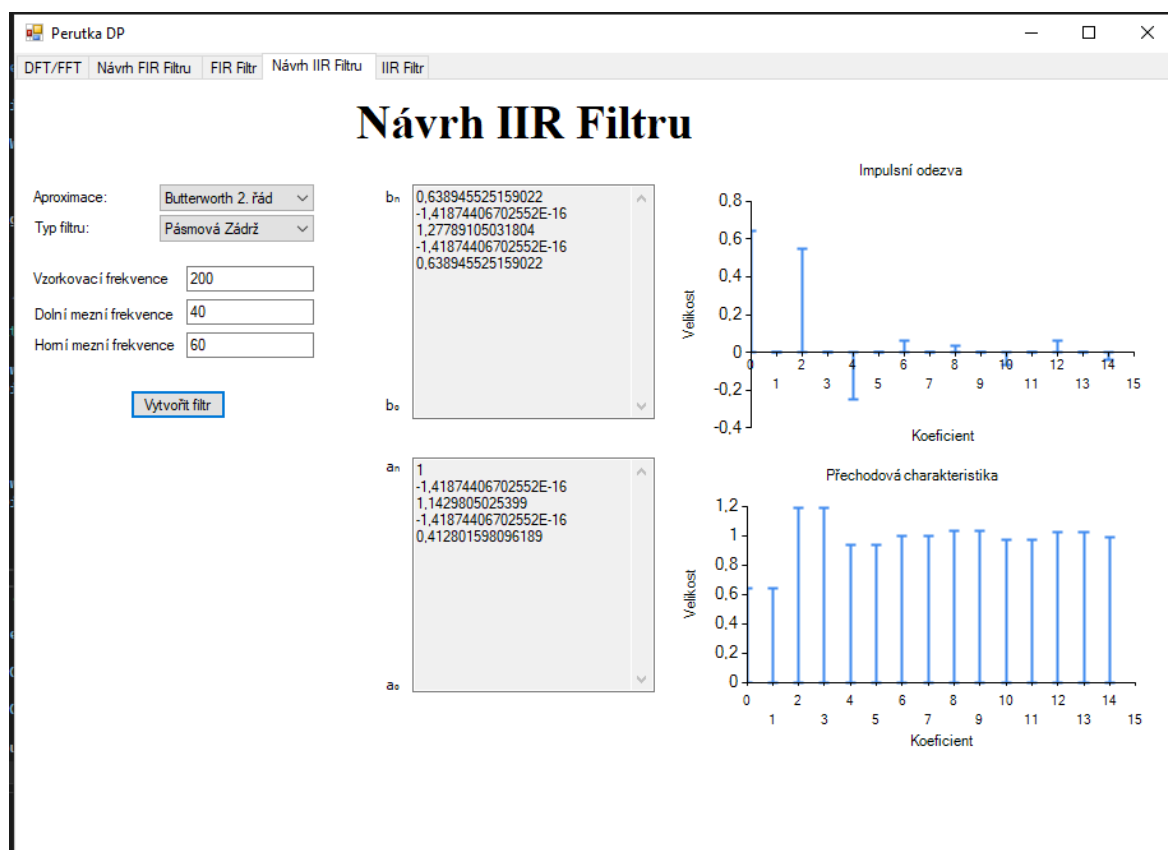
$$d_1 = -64 + 16b_3T - 4b_1T^3 + 4b_0T^4$$

$$d_2 = 96 - 8b_2T^2 + 6b_0T^4$$

$$d_3 = -64 - 16b_3T + 4b_1T^3 + 4b_0T^4$$

$$d_4 = 16 + 8b_3T + 4b_2T^2 + 2b_1T^3 + b_0T^4$$

Tyto koeficienty jsou v programu jako vzorce při výpočtu charakteristiky konkrétního filtru.



Obrázek 39: Program, návrh IIR pásmové zádrže

7.5 Pásmová propust

Pro výpočty je potřeba dopočítat zkreslené hraniční úhlové rychlosti podle vztahu:

$$\omega_{an1} = \frac{2}{T} * tg\left(\frac{2 * \pi * f_{c1} * T}{2}\right)$$

$$\omega_{an2} = \frac{2}{T} * tg\left(\frac{2 * \pi * f_{c2} * T}{2}\right)$$

Kde:

ω_{an1} je zkreslená úhlová rychlost dolní meze

ω_{an2} je zkreslená úhlová rychlost horní meze

T je perioda signálu $\frac{1}{f_{vz}}$

f_{c1} je mezní frekvence dolní meze

f_{c2} je mezní frekvence horní meze

f_{vz} je vzorkovací frekvence

Z těchto úhlových rychlostí spočítáme:

$$\omega_{\Delta} = \omega_{an2} - \omega_{an1}$$

$$\omega_0^2 = \omega_{an2} * \omega_{an1}$$

Tyto hodnoty použijeme ve výpočtech koeficientů

Analogové koeficienty:

Obecný tvar:

$$\frac{a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0}{b_4 s^4 + b_3 s^3 + b_2 s^2 + b_1 s + a_0}$$

Po výpočtech jsou výsledné analogové koeficienty:

$$a_0 = 0$$

$$a_1 = 0$$

$$a_2 = \omega_{\Delta}^2$$

$$a_3 = 0$$

$$a_4 = 0$$

$$b_0 = \omega_0^4$$

$$b_1 = \sqrt{2}\omega_{\Delta}\omega_0^2$$

$$b_2 = \omega_{\Delta}^2 + 2\omega_0^2$$

$$b_3 = \sqrt{2}\omega_{\Delta}$$

$$b_4 = 1$$

Digitální koeficienty:

Obecný tvar:

$$\frac{c_4z^4 + c_3z^3 + c_2z^2 + c_1z + c_0}{d_4z^4 + d_3z^3 + d_2z^2 + d_1z + d_0}$$

Po bilineární transformaci a úpravě jsou výsledné digitální koeficienty:

$$c_0 = 4a_2T^2$$

$$c_1 = 0$$

$$c_2 = -8a_2T^2$$

$$c_3 = 0$$

$$c_4 = 4a_2T^2$$

$$d_0 = 16 - 8b_3T + 4b_2T^2 - 2b_1T^3 + b_0T^4$$

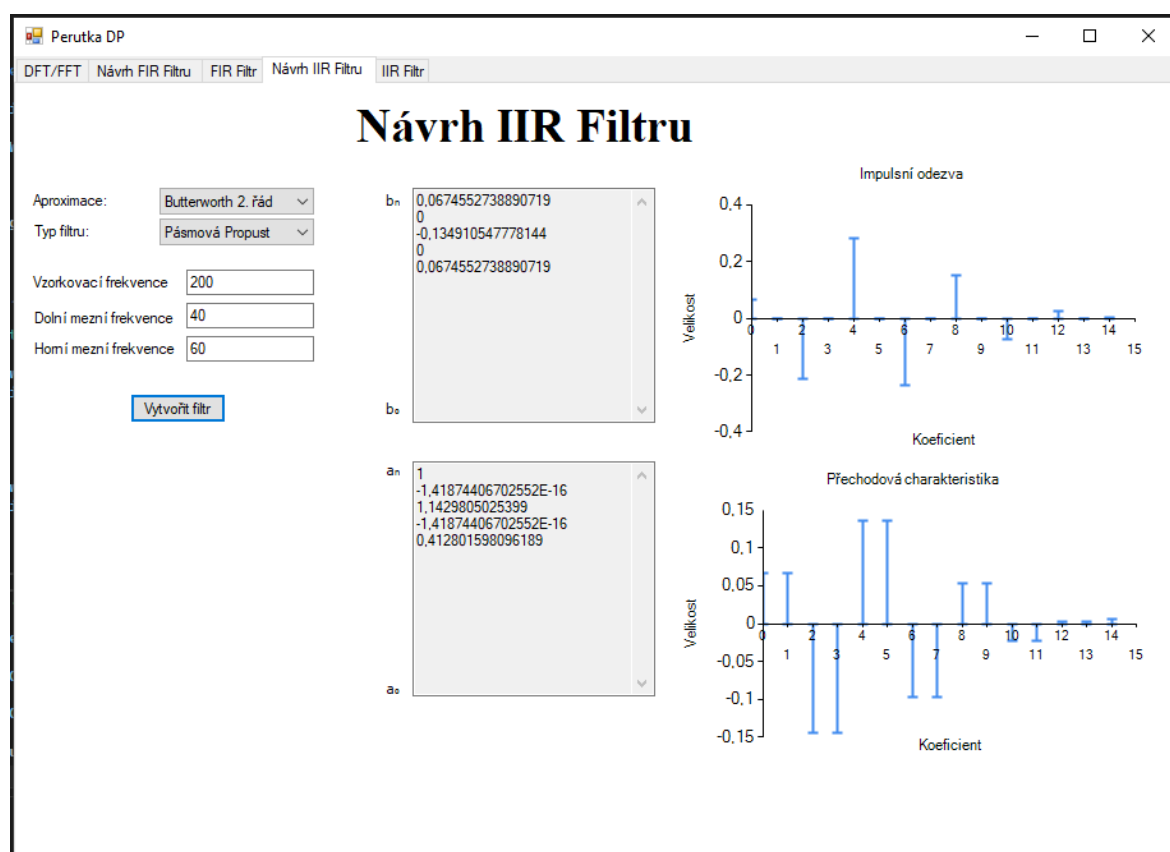
$$d_1 = -64 + 16b_3T - 4b_1T^3 + 4b_0T^4$$

$$d_2 = 96 - 8b_2T^2 + 6b_0T^4$$

$$d_3 = -64 - 16b_3T + 4b_1T^3 + 4b_0T^4$$

$$d_4 = 16 + 8b_3T + 4b_2T^2 + 2b_1T^3 + b_0T^4$$

Tyto koeficienty jsou v programu jako vzorce při výpočtu charakteristiky konkrétního filtru.



Obrázek 40: Program, návrh IIR pásmové propusti

8 TAB 5 IIR FILTR

Poslední část demonstruje samotné použití IIR filtru pro vstupní signál (filtrací vstupního signálu). Na obrázku je kromě vah i prostor pro výslednou IIR rovnici filtru.

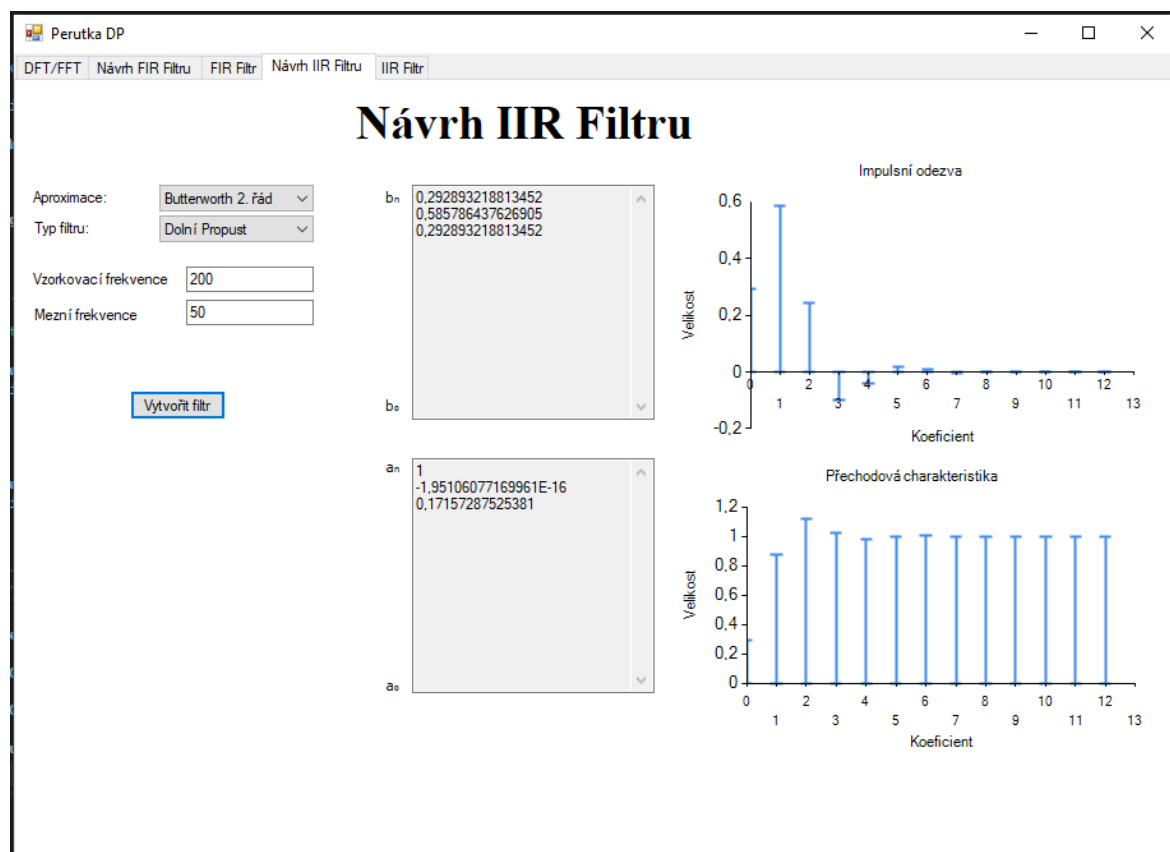


Obrázek 41: Program, IIR filtr

8.1.1 Příklad IIR Filtrace

Bylo vygenerováno 20 hodnot od 0 do 5. Filtr byl sestaven s parametry:

- Dolní propust
- Prototyp Butterworth 2. řád
- Vzorkovací frekvence 200 Hz
- Mezní frekvence 50 Hz



Obrázek 42: Návrh IIR filtru pro příklad

Načtením této charakteristiky filtru se nám vyplní váhy pro vstupy a výstupy a zobrazí se rovnice odpovídající našemu filtru. Grafy vizualizují vstupní a výstupní signál při průchodu filtrem.



Obrázek 43: Příklad IIR filtrace

ZÁVĚR

V této diplomové práci byla zpracována problematika frekvenční analýzy a návrhu číslicových filtrů s důrazem na praktické využití metod DSP v prostředí Windows Forms. Cílem bylo vytvořit nástroj, který umožní efektivní analýzu a úpravu číslicových signálů pomocí metod diskrétní Fourierovy transformace (DFT, FFT) a návrhu FIR a IIR filtrů.

V teoretické části byly shrnuty klíčové principy, které tvoří základ číslicového zpracování signálů – od samotné Fourierovy analýzy po metody návrhu číslicových filtrů. V praktické části pak byl navržen a implementován software, který uživateli umožňuje nejen analyzovat frekvenční složení signálů, ale také navrhnout vhodný filtr a sledovat jeho účinky na konkrétních datech.

Výsledná aplikace prokázala svou funkčnost na zadaných testovacích scénářích a může sloužit jako výukový nástroj pro studenty nebo jako základní platforma pro další vývoj nástrojů v oblasti DSP. Zpřístupnění zdrojového kódu zároveň otevírá prostor pro budoucí úpravy, například rozšíření o pokročilejší metody filtrace nebo integraci s externími zařízeními pro záznam a zpracování dat v reálném čase.

Tato práce tak propojuje teoretické znalosti s jejich praktickým uplatněním v oblasti efektivního zpracování číslicových signálů.

SEZNAM POUŽITÉ LITERATURY

- [1] BANSAL, Mohan. *Analog and Digital Signal | Continuous and Discrete Time Signal*. Online. YouTube. 2020. Dostupné z: <https://www.youtube.com/watch?v=gRU4snny14s>. [cit. 2025-05-12].
- [2] OPPENHEIM, Alan V.; WILLSKY, Alan S. a NAWAB, S. Hamid. Continuous-time and Discrete-time signals. In: *Signals & Systems*. 2. vydání. Prentice-Hall, 1996, s. 1-5. ISBN 9780136511755.
- [3] DOŇAR, Bohuslav a ZAPLATÍLEK, Karel. *MATLAB – začínáme se signály*. BEN, 2006. ISBN 8073002000.
- [4] HLAVÁČ, Václav a SEDLÁČEK, Miloš. *Zpracování signálů a obrazů*. ČVUT, 2009. ISBN 978-80-01-04442-1.
- [5] *Vzorkování*. Online. In: WIKIPEDIA. Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2005, 3. 6. 2024. Dostupné z: <https://cs.wikipedia.org/wiki/Vzorkov%C3%A1n%C3%AD>. [cit. 2025-05-12].
- [6] *Vzorkovací frekvence*. Online. In: WIKIPEDIA. Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2007, 16. 3. 2023. Dostupné z: https://cs.wikipedia.org/wiki/Vzorkovac%C3%AD_frekvence. [cit. 2025-05-12].
- [7] PROGRAHAMING. *Sample And Hold Sampling Zero-order Hold First-order Hold Digital-to-analog Converter PNG*. Online, obrázek. FAVPNG.COM. FAVPNG. © 2025. Dostupné z: https://favpng.com/png_view/signal-sample-and-hold-sampling-zero-order-hold-first-order-hold-digital-to-analog-converter-png/mMn6aTHB. [cit. 2025-05-12].
- [8] *Kvantování (signál)*. Online. In: WIKIPEDIA. Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2007, 21. 5. 2022. Dostupné z: [https://cs.wikipedia.org/wiki/Kvantov%C3%A1n%C3%AD_\(sign%C3%A1l\)](https://cs.wikipedia.org/wiki/Kvantov%C3%A1n%C3%AD_(sign%C3%A1l)). [cit. 2025-05-12].
- [9] *A/D převodník*. Online. In: WIKIPEDIA. Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2011, 28. 4. 2025. Dostupné z:

- https://cs.wikipedia.org/wiki/A/D_p%C5%99evodn%C3%ADk. [cit. 2025-05-12].
- [10] *Quantized.signal*. Online, obrázek. In: WIKIPEDIA. Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2011, 28. 4. 2025. Dostupné z: https://cs.wikipedia.org/wiki/Kvantov%C3%A1n%C3%AD_%28sign%C3%A1l%29#/media/Soubor:Quantized.signal.svg. [cit. 2025-05-12].
- [11] *Fourierova řada v goniometrickém tvaru*. Online. VUT FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ. Ústav telekomunikací. © 2025. Dostupné z: https://www.utko.fekt.vut.cz/~rajmic/applets/vliv_na_FR_gonio/. [cit. 2025-05-14].
- [12] OPPENHEIM, Alan V.; WILLISKY, Alan S. a NAWAB, S. Hamid. Fourier series representation of periodic signals. In: *Signals & Systems*. 2. vydání. Prentice-Hall, 1996, s. 177-283. ISBN 9780136511755.
- [13] *Conditions for Existence of Fourier Series (Dirichlet Conditions)*. Online. NESO ACADEMY. YouTube. 2017. Dostupné z: https://www.youtube.com/watch?v=bIS_OImUJ-c. [cit. 2025-05-14].
- [14] ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ [CVUT]. *Vlastnosti Fourierových řad*. Online. Katedra matematiky. Dostupné z: <https://math.fel.cvut.cz/cz/mt/txt/3/txc3ea3g.htm>. [cit. 2025-05-14].
- [15] *Fourier Transform*. Online. GEEKSFORGEEEKS, SANCHHAYA EDUCATION PRIVATE LIMITED. Geeksforgeeks. 2024. Dostupné z: <https://www.geeksforgeeks.org/fourier-transform/>. [cit. 2025-05-12].
- [16] OPPENHEIM, Alan V.; WILLISKY, Alan S. a NAWAB, S. Hamid. The continuous-time Fourier transform. In: *Signals & Systems*. 2. vydání. Prentice-Hall, 1996, s. 284-357. ISBN 9780136511755.
- [17] *Discrete Fourier Transform*. Online. Sciencedirect. © 2025. Dostupné z: <https://www.sciencedirect.com/topics/engineering/discrete-fourier-transform>. [cit. 2025-05-12].

- [18] XU, Simon. *The FFT Algorithm – Simple Step by Step*. Online. YouTube. 2015. Dostupné z: <https://www.youtube.com/watch?v=htCj9exbGo0>. [cit. 2025-05-12].
- [19] *Cooley–Tukey FFT algorithm*. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2001-. Dostupné z: https://en.wikipedia.org/wiki/Cooley%E2%80%93Tukey_FFT_algorithm. [cit. 2025-05-14].
- [20] VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ [VUT]. *Kmitočtové filtry*. Online. Kmitočtové filtry. Dostupné z: https://www.radio.feec.vutbr.cz/raida/mcvt/textbook/01_kmitoctove_filtry.pdf. [cit. 2025-05-14].
- [21] VÍCH, Robert a SMÉKAL, Zdeněk. *Číslicové filtry*. Academia, 2001. ISBN 978-80-200-0761-2.
- [22] DAVÍDEK, Vratislav; VLČEK, Jaroslav a LAIPERT, Miloš. *Analogové a číslicové filtry*. 2. vydání. ČVUT, 2006. ISBN 80-01-03026-1.
- [23] INSTITUT BIOSTATISTIKY A ANALÝZ LÉKAŘSKÉ FAKULTY MASARYKOVY UNIVERZITY. *Lineární filtrace*. Online, obrázek. Matematická biologie učebnice. Dostupné z: <https://portal.matematickabiologie.cz/index.php?pg=analiza-a-modelovani-dynamickych-biologickych-dat--linearni-a-adaptivni-zpracovani-dat--linearni-filtrace-princip--linearni-filtrace>. [cit. 2025-05-17].
- [24] SANJEEV, Sarpal. *Difference between IIR and FIR filters: a practical design guide*. Online. Advanced Solutions Nederland. 2020. Dostupné z: <https://www.advsolned.com/difference-between-iir-and-fir-filters-a-practical-design-guide/>. [cit. 2025-05-14].
- [25] DINIZ, Paulo S. R.; DA SILVA, Eduardo A. B. a NETTO, Sergio L. *FIR filter approximation*. In: *Digital Signal Processing*. 2. vydání. Cambridge University Press, 2010, s. 277-348. ISBN 978-0-521-88775-5.
- [26] UNIVERSITY OF NEWCASTLE UPON TYNE. *Design of FIR Filters*. Online. Newcastle University. © 2021. Dostupné z:

- <https://www.staff.ncl.ac.uk/oliver.hinton/eee305/Chapter4.pdf>. [cit. 2025-05-14].
- [27] TAN, Li. Finite Impulse Response Filter Design. In: *Digital Signal Processing*. Elsevier, 2008, s. 215-302. ISBN 978-0-12-374090-8.
- [28] *Triangular window*. Online. RecordingBlogs. Dostupné z: <https://www.recordingblogs.com/wiki/triangular-window>. [cit. 2025-05-14].
- [29] *Window function*. Online, obrázek. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2001-2025. Dostupné z: https://en.wikipedia.org/wiki/Window_function#/media/File:Window_function_and_frequency_response_-_Rectangular.svg. [cit. 2025-05-14].
- [30] *Window function*. Online, obrázek. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2001-2025. Dostupné z: [https://en.wikipedia.org/wiki/Window_function#/media/File:Window_function_and_frequency_response_-_Hamming_\(alpha_=_0.53836,_n_=_0...N\).svg](https://en.wikipedia.org/wiki/Window_function#/media/File:Window_function_and_frequency_response_-_Hamming_(alpha_=_0.53836,_n_=_0...N).svg). [cit. 2025-05-14].
- [31] *Window function*. Online, obrázek. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2001-2025. Dostupné z: [https://en.wikipedia.org/wiki/Window_function#/media/File:Window_function_and_its_Fourier_transform_%E2%80%93_Hann_\(n_=_0...N\).svg](https://en.wikipedia.org/wiki/Window_function#/media/File:Window_function_and_its_Fourier_transform_%E2%80%93_Hann_(n_=_0...N).svg). [cit. 2025-05-14].
- [32] *Window function*. Online, obrázek. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2001-2025. Dostupné z: [https://en.wikipedia.org/wiki/Window_function#/media/File:Window_function_and_its_Fourier_transform_%E2%80%93_Triangular_\(n_=_0...N\).svg](https://en.wikipedia.org/wiki/Window_function#/media/File:Window_function_and_its_Fourier_transform_%E2%80%93_Triangular_(n_=_0...N).svg). [cit. 2025-05-14].
- [33] *Window function*. Online, obrázek. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2001-2025. Dostupné z: [https://en.wikipedia.org/wiki/Window_function#/media/File:Window_function_and_its_Fourier_transform_%E2%80%93_Blackman_\(n_=_0...N\).svg](https://en.wikipedia.org/wiki/Window_function#/media/File:Window_function_and_its_Fourier_transform_%E2%80%93_Blackman_(n_=_0...N).svg). [cit. 2025-05-14].

- [34] UNIVERSITY OF NEWCASTLE UPON TYNE. *Design of IIR Filters*. Online. Newcastle University. © 2021. Dostupné z: <https://www.staff.ncl.ac.uk/oliver.hinton/eee305/Chapter5.pdf>. [cit. 2025-05-14].
- [35] TAN, Li. Infinite Impulse Response Filter Design. In: *Digital Signal Processing*. Elsevier, 2008, s. 303-412. ISBN 978-0-12-374090-8.
- [36] DINIZ, Paulo S. R.; DA SILVA, Eduardo A. B. a NETTO, Sergio L. IIR filter approximation. In: *Digital Signal Processing*. 2. vydání. Cambridge University Press, 2010, s. 349-408. ISBN 978-0-521-88775-5.
- [37] *Butterworth filter*. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2001-2025. Dostupné z: https://en.wikipedia.org/wiki/Butterworth_filter. [cit. 2025-05-14].
- [38] PŘIKRYL, Jan a VLČEK, Miroslav. *Bilineární transformace Shrnutí*. Online. CVUT. Zolotarev.fd.cvut.cz. 2015. Dostupné z: <https://zolotarev.fd.cvut.cz/static/msap/msp-prg-2015-12-handouts.pdf>. [cit. 2025-05-18].
- [39] *Graf Fungsi Sinus, Kosinus dan Tangen*. Online, graf. SPM Matematik. © 2025. Dostupné z: <https://spmmatematik.blog.onlinetuition.com.my/2021/06/92-graf-sinus-kosinus-dan-tangen.html>. [cit. 2025-05-14].

SEZNAM OBRÁZKŮ

Obrázek 1: Spojitý signál.....	13
Obrázek 2: Diskrétní signál	14
Obrázek 3: Vzorkování signálu [7].....	15
Obrázek 4: Kvantování signálu [10].....	16
Obrázek 5: Ideální dolní propust [23].....	21
Obrázek 6: Ideální horní propust [23].....	22
Obrázek 7: Ideální pásmová propust [23].....	22
Obrázek 8: Ideální pásmová zadrž [23]	23
Obrázek 9: Obdélníkové okno [29]	25
Obrázek 10: Hammingovo okno [30]	26
Obrázek 11: Hannovo okno [31]	26
Obrázek 12: Trojúhelníkové okno [32]	27
Obrázek 13: Blackmanovo okno [33].....	27
Obrázek 14: Srovnání oken.....	28
Obrázek 15: Schéma návrhu IIR filtru.....	29
Obrázek 16: Butterworth prototypy pro jednotlivé řády [37].....	30
Obrázek 17: Převodní vztahy z normované dolní propusti.....	30
Obrázek 18: Program, Fourierova transformace.....	33
Obrázek 19: Funkce sinus [39]	34
Obrázek 20: Program, příklad diskretní Fourierovy transformace	36
Obrázek 21: Program, příklad rychlé Fourierovy transformace	39
Obrázek 22: Program, Návrh FIR filtru.....	41
Obrázek 23: Program, FIR návrh dolní propusti	44
Obrázek 24: Program, FIR návrh horní propusti	44
Obrázek 25: Program, FIR pásmové propusti	46
Obrázek 26: Program, FIR pásmové zadrž	46
Obrázek 27: Program, příklad s obdélníkovým oknem	47
Obrázek 28: Program, příklad s Hammingovým oknem	48
Obrázek 29: Program, příklad s Hannovým oknem	49
Obrázek 30: Program, příklad s trojúhelníkovým oknem.....	50
Obrázek 31: Program, příklad s Blackmanovým oknem.....	51
Obrázek 32: Program, FIR filtr.....	52
Obrázek 33: Program, FIR filtr, okno pro generování signálu	53
Obrázek 34: Program, příklad FIR filtrace	54

Obrázek 35: Program, návrh IIR filtru.....	55
Obrázek 36: Převodní vztahy z normované dolní propusti.....	56
Obrázek 37: Program, návrh IIR dolní propusti	58
Obrázek 38: Program, návrh IIR horní propusti	59
Obrázek 39: Program, návrh IIR pásmové zadržky	62
Obrázek 40: Program, návrh IIR pásmové propusti	65
Obrázek 41: Program, IIR filtr.....	66
Obrázek 42: Návrh IIR filtru pro příklad	67
Obrázek 43: Příklad IIR filtrace.....	68

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

DFT Discrete Fourier transform

FFT Fast Fourier transform

FIR Finite impulse response

IIR Infinite impulse response

SEZNAM PŘÍLOH

Příloha P I: Odkaz na zdrojový kód

PŘÍLOHA P I: ODKAZ NA ZDROJOVÝ KÓD

`https://github.com/DeadArrow1/DPFATool`