



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

WEBOVÁ DEMONSTRACE NÁVRHU LINEÁRNÍCH FILTRŮ

WEB-BASED DEMONSTRATION OF LINEAR FILTERS DESIGN

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ONDŘEJ POSPÍŠIL

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ONDŘEJ NOVOTNÝ

BRNO 2017

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2016/2017

Zadání bakalářské práce

Řešitel: **Pospíšil Ondřej**

Obor: Informační technologie

Téma: **Webová demonstrace návrhu lineárních filtrů**
Web-Based Demonstration of Linear Filters Design

Kategorie: Web

Pokyny:

1. Prostudujte si teorii zpracování signálů, s důrazem na lineární číslicovou filtraci.
2. Prostudujte metody návrhů lineárních číslicových filtrů (FIR i IIR).
3. Seznamte se s demonstracemi, které jsou k dispozici na webu a kriticky je zhodnoťte.
4. Navrhněte vlastní demonstraci zabývající se návrhem filtrů a prokonzultujte její teoretickou i grafickou podobu s vedoucím.
5. Zvolte implementační jazyk a demonstraci implementujte.
6. Umístěte na www stránkách, získejte a shrňte základní reakce uživatelů.

Literatura:

- <http://www.fit.vutbr.cz/study/courses/ISS/public/>
- Jan, Jiří: Číslicová filtrace, analýza a restaurace signálů, VUTIUM, 2002.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 4 zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Novotný Ondřej, Ing.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2016

Datum odevzdání: 17. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
611 65 Brno, Sečtetěchova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Cílem této práce je vytvořit demonstrační aplikaci zaměřenou na návrh číslicových filtrů. V prvních kapitolách se věnujeme teorii číslicové filtrace a návrhu číslicových filtrů. V následující kapitole si shrneme a ohodnotíme existující řešení. V poslední části je popsán návrh a implementace vlastní demonstrační aplikace.

Abstract

The aim of this work is to create a demonstration application focused on the design of digital filters. In the first chapters we outline the theory of digital filtration and design of digital filters. Then we summarize and evaluate the existing solutions. The following chapters describe the design and implementation of the actual application.

Klíčová slova

Signál, zpracování signálu, filtr, číslicový filtr, FIR filtr, IIR filtr, návrh filtru

Keywords

Signal, signal processing, filter, digital filter, FIR filter, IIR filter, filter design

Citace

POSPÍŠIL, Ondřej. *Webová demonstrace návrhu lineárních filtrů*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Ondřej Novotný

Webová demonstrace návrhu lineárních filtrů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Ondřeje Novotného. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Ondřej Pospíšil
16. května 2017

Poděkování

Rád bych poděkoval vedoucímu práce Ing. Ondřeji Novotnému za cenné rady, vstřícnost při konzultacích a odborný dohled. Dále rodině a přítelkyni za morální podporu.

Obsah

1	Úvod	4
2	Teorie	5
2.1	Diskrétní systémy	5
2.2	Charakteristiky systémů	6
2.2.1	Impulsní charakteristika	6
2.2.2	Kmitočtová charakteristika	6
2.2.3	Přenosová funkce	8
3	Návrh číslicových filtrů	9
3.1	Návrh číslicových filtrů typu FIR	9
3.1.1	Metoda vzorkování frekvenční charakteristiky	9
3.1.2	Metoda váhování impulsní charakteristiky	10
3.2	Návrh číslicových filtrů typu IIR	13
3.2.1	Metody využívající podobnost s analogovými filtry	13
3.3	Metody přímého návrhu	15
3.3.1	Modifikovaná Yule-Walkerova metoda	15
4	Existující řešení	17
4.1	Způsob hodnocení	17
4.2	Vybraná řešení	17
4.2.1	MicroModeler DSP	17
4.2.2	TFilter	18
4.2.3	Kaiser-Bessel filter generator	19
4.2.4	Bakalářská práce Ondřeje Novotného	19
4.3	Shrnutí	19
5	Popis implementace	21
5.1	Implementační prostředí	21
5.2	Návrh aplikace	21
5.3	Implementace	22
5.3.1	Loader, Locale, Charts	22
5.3.2	Controller, CookiesHandler	22
5.3.3	Filter, DspUtils	23
5.3.4	Použité knihovny	23
5.3.5	Problémy	24
5.4	Výsledná aplikace a její použití	24
5.4.1	Ovládání	24

5.4.2	Výstup	25
5.4.3	Scénář použití	25
5.4.4	Dostupnost	25
6	Hodnocení aplikace	27
6.1	Formulář	27
6.2	Vyhodnocení formuláře	27
7	Závěr	29
7.1	Budoucí práce	29
	Literatura	30
	A Obsah CD	31
	B Výsledky formuláře	32
B.1	Co se Vám na aplikaci líbí?	32
B.2	Co se Vám na aplikaci nelíbí?	32
B.3	Připomínky	32

Seznam obrázků

2.1	Schéma obecně rekurzivního filtru.	6
2.2	Charakteristiky horní propusti	7
3.1	Frekvenční charakteristika dolní propusti, zobrazující parametry návrhu (vztah mezi ideálním a navrhovaným filtrem).	10
3.2	Okenní funkce	12
3.3	Vzájemná souvislost roviny s a roviny z	14
4.1	MicroModeler DSP	18
4.2	TFilter	18
4.3	Kaiser–Bessel filter generator	19
4.4	Bakalářská práce Ondřeje Novotného	20
5.1	Návrhový vzor Model-View-Presenter	22
5.2	Zjednodušený diagram tříd.	23
5.3	Charakteristiky a nastavení filtru	25
5.4	Koeficienty filtru, kód v jazyce C	26

Kapitola 1

Úvod

Číslicové zpracování signálu má dosah v různých sférách běžného života, jako například zpracování audiovizuálních dat. Za zmínku stojí především rozpoznávání lidské řeči. V dnešní době je číslicové zpracování signálu v popředí oproti analogovému řešení. V této práci se zaměříme pouze na úzkou část tohoto oboru, a to je *filtrace signálů*.

Filtr můžeme chápat jako systém, do kterého vstupuje signál a vystupuje upravený signál. Filtry můžeme rozdělit do dvou skupin: s nekonečnou a konečnou impulsní odezvou. V prvním případě se při zpracování signálu používají i staré vzorky výstupu, v druhém případě nikoliv. Popsat filtr se dá nemalým množstvím charakteristik. V této práci se budeme zabývat především návrhem těchto filtrů. Návrh filtrů se dá rozdělit podle toho, jakou bude mít výsledný filtr impulsní odezvu. Dají se také charakterizovat dle přístupu, který používají při samotném návrhu. Tedy zda-li používají analogový vzor filtru či nikoliv.

Cílem této práce bude shrnout teorii z číslicové lineární filtrace a návrhu filtrů. Prozkoumat existující řešení a vytvořit webovou demonstraci interaktivního návrhu lineárních filtrů. V kapitolách 2 a 3 se budeme zabývat teorií o filtrech a jejich charakteristik, plynule navážeme na metody návrhu filtru. Poté si v kapitole 4 probereme dostupná existující řešení. Dále si přiblížíme návrh a implementační detaily demonstrace (kapitola 5). V závěrečné kapitole 6 se zaměříme na testování s uživateli.

Kapitola 2

Teorie

V této kapitole si vysvětlíme základy diskretních systémů. Popíšeme jejich vlastnosti a základní rozdělení filtrů. Dále projdeme různé charakteristiky, díky kterým se dozvíme chování filtru. Teorie diskretních systémů je sepsána podle [15], charakteristiky jsou popsány podle [8] a [14].

2.1 Diskretní systémy

Diskretní systém si můžeme představit jako lineární funkci, vstupem jsou signály a výstupem jsou upravené signály. Základním matematickým popisem vlastností diskretních systémů jsou diferenční rovnice.

V této práci budeme pojednávat o **lineárních časově invariantních číslicových systémech s pamětí**. Tyto systémy splňují podmínky linearity, tedy aditivitu 2.1a a scaling 2.1b.

$$x_1[n] + x_2[n] \rightarrow y_1[n] + y_2[n] \quad (2.1a)$$

$$ax[n] \rightarrow ay[n] \quad (2.1b)$$

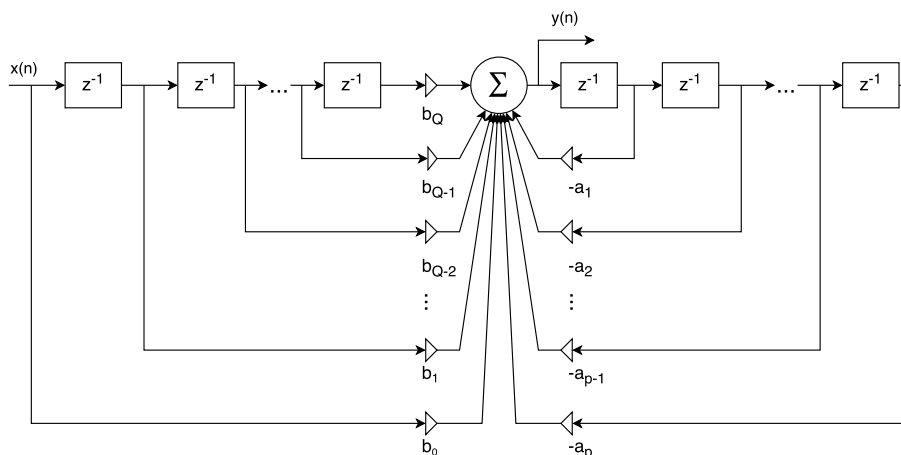
Dále jsou neměnné v čase, jejich vlastnosti zůstávají po celou dobu stejné. Posledním atributem je paměť, pomocí které jsou schopny si uchovat předešlé hodnoty, a dělí se na dva typy:

- *nerekurzivní*, které využívají pro výpočet výstupu jen vstupních hodnot, tyto filtry označujeme jako FIR - Finite Impulse Response,
- *rekurzivní*, využívající jak vstupních, tak zpožděných výstupních hodnot. Jde tedy o systémy se zpětnými vazbami, tyto filtry označujeme jako IIR - Infinite Impulse Response.

Výstup obecně rekurzivního filtru se dá zapsat vztahem:

$$y[n] = \sum_{k=0}^Q b_k x[n-k] - \sum_{k=1}^P a_k y[n-k], \quad (2.2)$$

kde $x[n-k]$ jsou aktuální a zpožděné verze vstupu o k vzorků a $y[n-k]$ jsou zpožděné verze výstupu taktéž o k vzorků.



Obrázek 2.1: Schéma obecně rekurzivního filtru.

2.2 Charakteristiky systémů

Chování a vlastnosti diskrétního systému můžeme popsat pomocí různých charakteristik. Tou první je impulsní charakteristika nebo také impulsní odezva (viz obr. 2.2b). Rozebereme si také frekvenční charakteristiku (viz. obr. 2.2a), přenosovou funkci filtru a s ní související nuly a póly (viz. obr. 2.2c, kde nuly n_k jsou zaznačeny jako kroužky a póly p_k jako křížky).

2.2.1 Impulsní charakteristika

Jedná se o odezvu systému na jednotkový impuls. Impulsní odezva nám dá představu o reakci systému na vstup. Vstupní signál se dá rozložit na jednotlivé vzorky, každý z těchto vzorků spouští vlastní impulsní odezvu. Konvolucí vstupního signálu s impulsní odezvou lze získat výstup filtru:

$$y[n] = x[n] \star h[n] = \sum_{k=-\infty}^{+\infty} x[k]h[n-k]. \quad (2.3)$$

2.2.2 Kmitočtová charakteristika

Pro studium frekvenčního chování systému mu na vstup předložíme komplexní exponenciálu s normovanou kruhovou frekvencí ω :

$$x[n] = e^{-j\omega_1 n}. \quad (2.4)$$

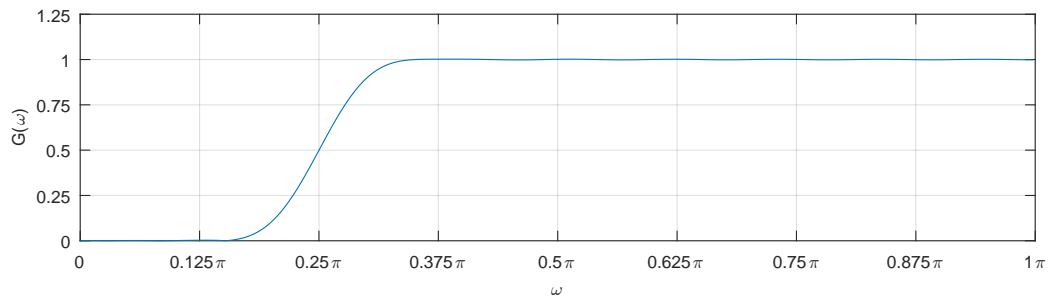
Zjistíme, že se nám na výstupu objevil stejný signál, ovšem násobený funkcí své kruhové frekvence a impulsní odezvy. Označíme tuto funkci:

$$H(e^{j\omega}) = \sum_{k=0}^{\infty} h[k]e^{-j\omega k}. \quad (2.5)$$

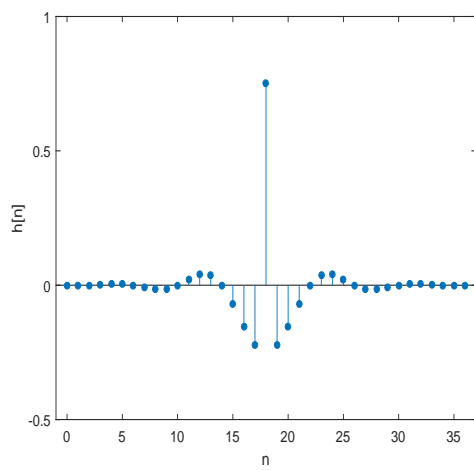
Zjišťujeme, že tato kmitočtová charakteristika je obrazem Fourierovy transformace s diskrétním časem nad impulsní odezvou:

$$h[n] \xrightarrow{DTFT} H(e^{j\omega}). \quad (2.6)$$

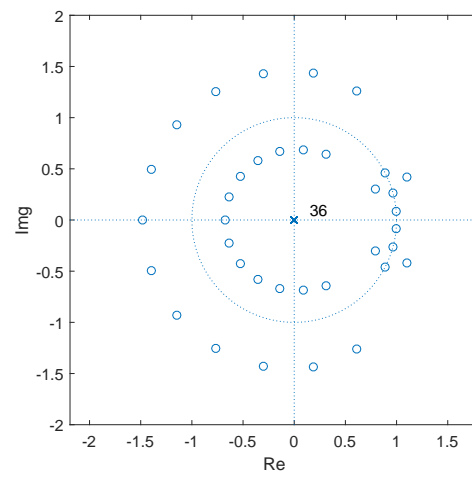
Musí tedy mít její vlastnosti, jako periodicitu spektra a symetrii.



(a) Frekvenční charakteristika



(b) Impulsní charakteristika



(c) Nuly a póly

Obrázek 2.2: Charakteristiky horní propusti

2.2.3 Přenosová funkce

Pro systém se vstupem $x[n]$ a výstupem $y[n]$ budeme přenosovou funkcí definovat jako:

$$H(z) = \frac{Y(z)}{X(z)}. \quad (2.7)$$

Pro dosažení do této rovnice musíme z -transformovat diferenční rovnici 2.2 následovně:

$$y[n] = \sum_{k=0}^Q b_k x[n-k] - \sum_{k=1}^P a_k y[n-k] \longrightarrow Y(z) = \sum_{k=0}^Q b_k X(z) z^{-k} - \sum_{k=1}^P a_k Y(z) z^{-k}. \quad (2.8)$$

Takto transformovanou rovnici dosadíme do vzorce 2.7 a můžeme vypočítat přenosovou funkci:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^Q b_k z^{-k}}{1 + \sum_{k=1}^P a_k z^{-k}} = \frac{B(z)}{A(z)}, \quad (2.9)$$

kde $B(z)$ a $A(z)$ jsou dva polynomy.

Nuly a póly přenosové funkce

Přenosovou funkci 2.7 můžeme zapsat také pomocí součinů:

$$H(z) = \frac{B(z)}{A(z)} = b_0 z^{P-Q} \frac{\prod_{k=1}^Q (z - n_k)}{\prod_{k=1}^P (z - p_k)}, \quad (2.10)$$

kde n_k jsou body v rovině z , pro tyto hodnoty bude funkce $H(z)$ nulová, nazývají se tedy nulové body. A p_k jsou body v rovině z , pro tyto hodnoty bude funkce $H(z)$ nekonečná, nazývají se póly.

Zde je nutné podotknout další vlastnost filtrů, a to je stabilita filtru. Systém je stabilní, pokud všechny póly leží uvnitř jednotkové kružnice. Musí tedy platit tato podmínka $|p_k| < 1$.

Kapitola 3

Návrh číslicových filtrů

V této kapitole si popíšeme návrhové metody filtrů. Tyto metody jsou pro každý typ filtru různé. Nejprve projdeme návrhy FIR filtrů a poté IIR filtrů.

3.1 Návrh číslicových filtrů typu FIR

Pro návrh číslicových filtrů existuje mnoho různých metod. V této kapitole se budeme soustředit na dvě metody návrhu, které byly použity v implementaci a jsou obecně nejrozšířenější. Konkrétně tyto dvě metody: metoda vzorkování frekvenční charakteristiky a metoda váhování impulsní charakteristiky. Tato kapitola byla zpracována podle [8], pro pevná okna a metodu váhování impulsní charakteristiky byl také využit zdroj [5].

3.1.1 Metoda vzorkování frekvenční charakteristiky

Máme-li FIR filtr specifikován vektorem koeficientů h , a tedy i impulsní odezvou $h[n]$, můžeme díky inverzibilitě diskrétní Fourierovy transformace vypočítat jeho diskrétní frekvenční charakteristiku

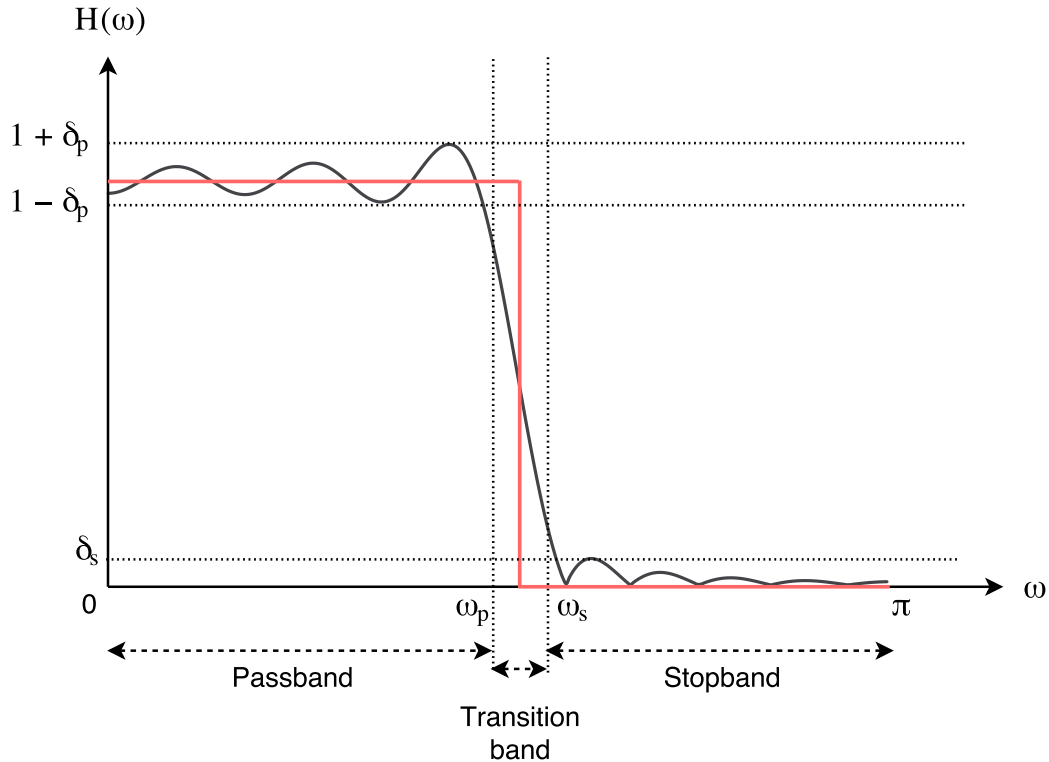
$$F_k = \sum_{n=0}^{N-1} h(n)e^{-j\frac{2\pi}{N}kn}. \quad (3.1)$$

Porovnáme-li tento výraz s 2.4, vidíme, že se jedná o vzorky spojité frekvenční charakteristiky. Na základě požadované frekvenční charakteristiky můžeme určit vektor h koeficientů filtru s konečnou impulsní odezvou. Je-li tedy zadána požadovaná frekvenční charakteristika v intervalu $\langle 0, \omega_s \rangle$, postupujeme takto:

1. Rozdělíme tento interval na zvolený počet N subintervalů a z nich odečteme N hodnot požadované frekvenční charakteristiky F_k .
2. Zpětnou diskrétní Fourierovou transformací na F_k získáme impulsní odezvu $h[n]$ a tedy i koeficienty filtru h .

Takto vypočtený filtr bude svou frekvenční charakteristikou přesně procházet v odečtených bodech F_k původní požadované charakteristiky. Při hustším vzorkování, tedy při více zvolených subintervalech, lze očekávat zlepšení aproximace.

V našem případě požadujeme, aby navrhovaný filtr byl s reálnými koeficienty. K tomu je třeba zajistit sudou symetrii reálné složky a lichou symetrii imaginární složky žádané frekvenční charakteristiky. Pak je tedy možné zadat žádoucí vzorky F_k jen pro $k = 0, 1, \dots, (N/2)$



Obrázek 3.1: Frekvenční charakteristika dolní propusti, zobrazující parametry návrhu (vztah mezi ideálním a navrhovaným filtrem).

a zbývající hodnoty dopočítat podle vztahu $F_{N-k} = F_k^*$. Pro kauzální filtr je třeba fázi jednotlivých prvků F_k zadat tak, aby její hodnoty odpovídaly žádanému filtru:

$$\arg(F_k) = -k\pi(N-1) + a\frac{\pi}{2}, \quad (3.2)$$

kde $a = 0$ pro filtry s požadovanou symetrickou impulsní charakteristikou a $a = 1$ pro antisymetrické charakteristiky.

Jednodušším postupem je zadat fázi nulovou, což vede na filtr nekauzální, a ten pak kauzalizovat zpožděním impulsní charakteristiky o $0.5T(N-1)$.

3.1.2 Metoda váhování impulsní charakteristiky

Vychází ze znalosti obecně neomezené impulsní charakteristiky požadovaného filtru. Vzhledem k tomu, že zpravidla jsou požadavky na filtr specifikovány frekvenční charakteristikou (viz obr. 3.1), tvoří tak první krok návrhu výpočet impulsní charakteristiky. Koeficienty impulsní charakteristiky můžeme přesně stanovit podle vztahu pro výpočet koeficientů Fourierovy řady

$$h_d[n] = \int_{-\frac{\pi}{T}}^{\frac{\pi}{T}} G_d(\omega) e^{j\omega n T} d\omega. \quad (3.3)$$

Vypočtená impulsní charakteristika je ovšem obecně nekonečná. Proto musíme její délku zkrátit na zvolený rozsah N členů a vynásobit oknem w

$$h[n] = h_d[n]w[n], \quad n \in \langle 0, N \rangle. \quad (3.4)$$

Výsledkem je frekvenční charakteristika $G(\omega)$, která se bude lišit od požadované frekvenční charakteristiky. Jde o to, aby rozdíly mezi $G(\omega)$ a $G_d(\omega)$ byly přijatelné. Toho lze dosáhnout různými tvary oken, tedy velikostí koeficientů w .

Pevná okna

Jedná se okna s předepsaným tvarem, které nelze nijak parametricky měnit. V této práci budeme používat pět základních oken:

1. Obdélníkové okno (3.2a)

$$\begin{aligned} w[n] &= 1, & n \in \langle 0, N-1 \rangle \\ w[n] &= 0, & n \in (-\infty, 0) \cup (N-1, \infty) \end{aligned} \quad (3.5)$$

2. Bartlettovo okno (3.2b)

$$\begin{aligned} w[n] &= \frac{2n}{N-1}, & n \in \left\langle 0, \frac{N-1}{2} \right\rangle \\ w[n] &= 2 - \frac{2n}{N-1}, & n \in \left\langle \frac{N-1}{2}, N-1 \right\rangle \end{aligned} \quad (3.6)$$

3. Hannovo okno (3.2c)

$$w[n] = 0.5 \left[1 - \cos \frac{2\pi n}{N-1} \right], \quad n \in \langle 0, N-1 \rangle \quad (3.7)$$

4. Hammingovo okno (3.2d)

$$w[n] = 0.54 - 0.46 \cos \frac{2\pi n}{N-1}, \quad n \in \langle 0, N-1 \rangle \quad (3.8)$$

5. Blackmanovo okno (3.2e)

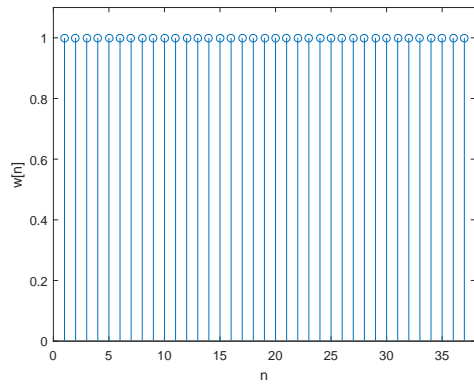
$$w[n] = 0.42 - 0.5 \cos \frac{2\pi n}{N-1} + 0.08 \cos \frac{4\pi n}{N-1}, \quad n \in \langle 0, N-1 \rangle \quad (3.9)$$

Parametrická okna

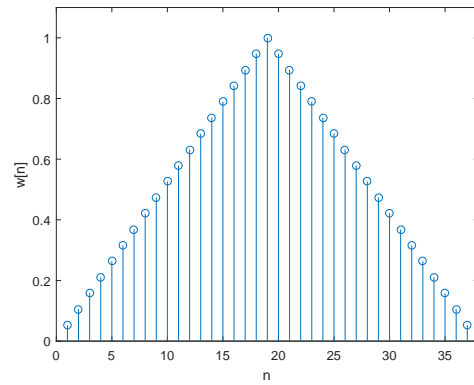
Další možností, jak získat požadovanou impulsní charakteristiku, jsou parametrická okna. Tato okna se mění podle zadaného parametru, tak abychom získali co nejpřesnější výsledek. Jedním z nich je Kaiserovo okno (3.2f), které si zde popíšeme. Pro výpočet jednotlivých koeficientů tohoto okna se používá vzorec:

$$w[n] = \frac{I_0(\beta \sqrt{1 - (\frac{2n-N}{N})^2})}{I_0(\beta)}, \quad n \in \langle 0, N-1 \rangle, \quad (3.10)$$

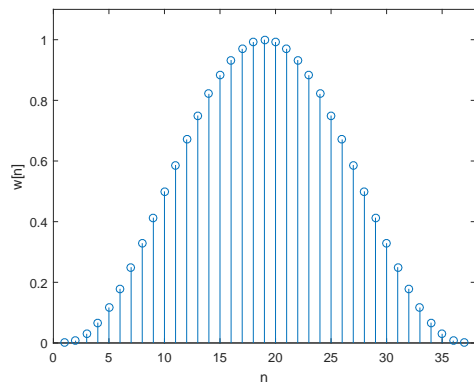
kde β je parametr umožňující manipulovat s vlnitostí propustného a závěrného pásma (viz. parametry δ_s a δ_p na obr. 3.1), I_0 je Besselova funkce nultého řádu. V této práci budeme používat pro výpočet Kaiserova okna Besselovu funkci prvního typu



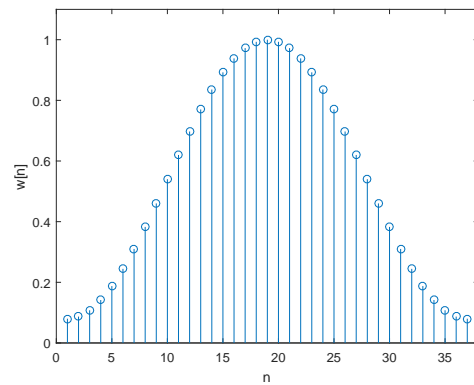
(a) Obdélíkové okno



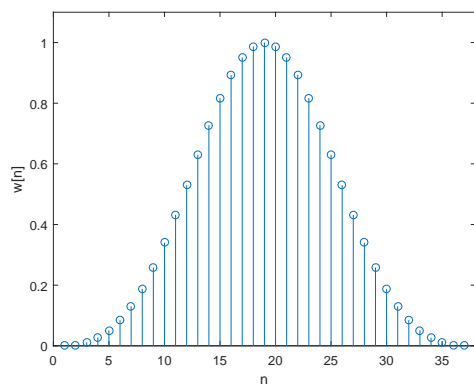
(b) Bartlettovo okno



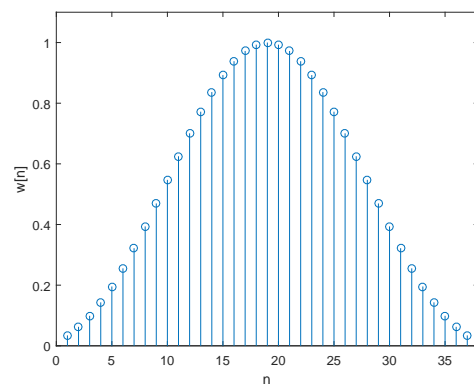
(c) Hannovo okno



(d) Hammingovo okno



(e) Blackmanovo okno



(f) Kaiserovo okno s parametrem $\alpha = 35dB$

Obrázek 3.2: Okenní funkce

$$I_0(x) = \sum_{k=0}^{\infty} \left[\frac{\left(\frac{x}{2}\right)^k}{k!} \right]^2. \quad (3.11)$$

Parametr β získáme transformací vstupního parametru α , dle následujícího vzorce 3.12.

$$\beta = \begin{cases} 0.1102(\alpha - 8.7) & \alpha > 50 \\ 0.5842(\alpha - 21)^{0.4} + 0.07886(\alpha - 21) & 21 \leq \alpha \leq 50 \\ 0 & \alpha < 21 \end{cases} \quad (3.12)$$

Jak již bylo řečeno, vstupem je parametr α , který uvádí míru zvlnění v dB , daného předpisem

$$\alpha = 20 \log(\min(\delta_s, \delta_p)). \quad (3.13)$$

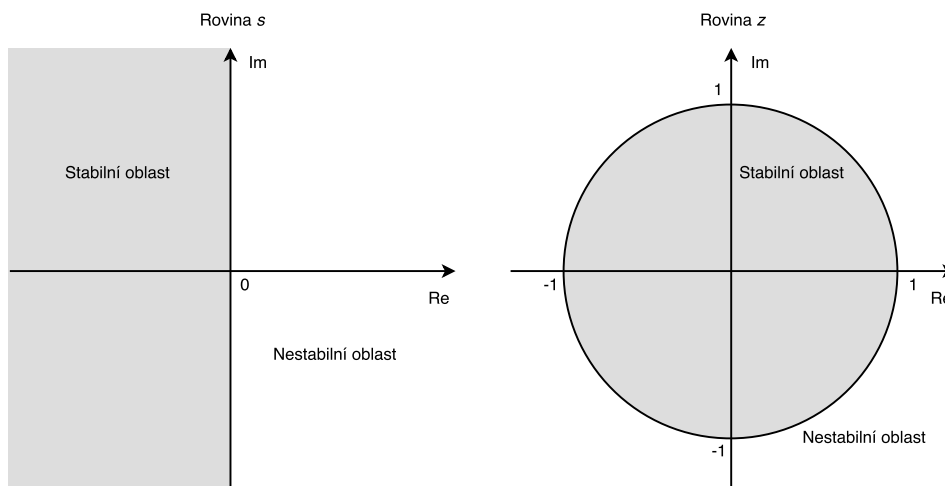
3.2 Návrh číslicových filtrů typu IIR

Stejně, jak pro filtry s konečnou impulsní odezvou, tak i zde je mnoho možných metod pro návrh filtru. Tyto metody se rozdělují do dvou skupin: metody využívající podobnost s analogovými filtry a metody přímého návrhu. Tato kapitola byla zpracována podle [14], [8], [2].

3.2.1 Metody využívající podobnost s analogovými filtry

Přenosová funkce analogových filtrů má, stejně jako v případě číslicových filtrů, tvar racionálně lomené funkce. Na základě této podobnosti je možné vhodnou transformací analogové přenosové funkce, definované v rovině s , získat přenosovou funkci IIR filtru v rovině z . Tato transformace má dvě základní podmínky:

1. Imaginární osa roviny $s = j\omega$ se musí transformovat na jednotkovou kružnici v rovině $z = e^{j\omega}$. Díky této podmínce se zachovají frekvenční charakteristiky analogového filtru.
2. Levá polorovina $Re(s) < 0$ roviny s se musí transformovat dovnitř jednotkové kružnice $|z| < 1$ roviny z . Tato podmínka zaručuje, že stabilní analogový filtr se bude transformovat na stabilní číslicový filtr.



Obrázek 3.3: Vzájemná souvislost roviny s a roviny z .

Butterworthova aproximace

Abychom mohli používat metody na základě podobnosti s analogovými filtry, musíme si vytvořit analogový filtr. Mezi ně například patří Butterworthův filtr. Pro výpočet polynomů přenosové funkce slouží následující vzorce.

- Pro sudý řád požadovaného filtru:

$$B_n(s) = \prod_{k=1}^{\frac{n}{2}} \left[s^2 - 2s \cos\left(\frac{2k+n-1}{2n}\pi\right) + 1 \right] \quad (3.14)$$

- Pro lichý řád požadovaného filtru:

$$B_n(s) = (s+1) \prod_{k=1}^{\frac{n-1}{2}} \left[s^2 - 2s \cos\left(\frac{2k+n-1}{2n}\pi\right) + 1 \right] \quad (3.15)$$

Těmito vztahy získáme přenosovou funkci normované dolní propusti. Provedeme frekvenční transformaci podle tabulky 3.1. Poté již můžeme tento analogový filtr použít pro výpočet číslicového IIR filtru.

Metoda invariance impulsní charakteristiky

Impulsní invariance je intuitivně nejpřijatelnější definice podobnosti mezi analogovým a diskrétním systémem v časové oblasti, která požaduje, aby impulsní charakteristika diskrétního systému $h_d[n]$ byla tvořena vzorky impulsní charakteristiky $h_a(nT)$ odpovídajícího systému analogového

$$h_d[n] = h_a[nT]. \quad (3.16)$$

Přenosovou funkci analogového filtru je možné rozložit na parciální zlomky

$$H_a(s) = \sum_{i=1}^N \frac{k_i}{s - p_i}. \quad (3.17)$$

Tabulka 3.1: Frekvenční transformace v analogové oblasti roviny s , viz. obr. 3.1

Dolní propust	$s \Rightarrow \frac{s}{\omega_p}$	ω_p - mezní úhlová frekvence propustného pásma dolní propusti
Horní propust	$s \Rightarrow \frac{\omega_p}{s}$	ω_p - mezní úhlová frekvence propustného pásma horní propusti
Pásmová propust	$s \Rightarrow \frac{s^2 + \omega_0^2}{s\omega_\Delta}$	$\omega_0^2 = \omega_{p1}\omega_{p2}$, $\omega_\Delta = \omega_{p2} - \omega_{p1}$, ω_{p1}, ω_{p2} - hranice propustného pásma
Pásmová zádrž	$s \Rightarrow \frac{s\omega_\Delta}{s^2 + \omega_0^2}$	$\omega_0^2 = \omega_{n1}\omega_{n2}$, $\omega_\Delta = \omega_{n2} - \omega_{n1}$, ω_{n1}, ω_{n2} - hranice nepropustného pásma

Na takto vyjádřenou přenosovou funkci použijeme zpětnou Laplaceovou transformaci

$$h_d[n] = \mathcal{L}^{-1}\{H_a(s)\} = \sum_{i=1}^N k_i e^{p_i n T}. \quad (3.18)$$

Podrobíme-li tento vztah transformaci \mathcal{Z} , dostaneme požadovanou impulsně invariantní přenosovou funkci číslicového filtru

$$H_d(z) = \mathcal{Z}\{h_d[n]\} = \sum_{i=1}^N k_i \frac{1}{1 - e^{p_i T} z^{-1}} = \sum_{i=1}^N k_i \frac{z}{z - e^{p_i T}}. \quad (3.19)$$

3.3 Metody přímého návrhu

Tyto metody realizují filtr přímo na základě specifikace ideální frekvenční charakteristiky. Nevyužívají podobnosti s analogovými filtry, proto nejsou omezeny na standardní typy filtrů, jako je například dolní propust, horní propust atd. Můžeme tedy navrhnout filtr s libovolnou frekvenční odezvou.

3.3.1 Modifikovaná Yule-Walkerova metoda

Tato metoda má dvě logické části, tou první je výpočet jmenovatele $A(z)$ a tou druhou je výpočet čitatele $B(z)$ přenosové funkce filtru definované v 2.7.

Máme-li definovanou ideální frekvenční charakteristiku, postupujeme obdobně jako při návrhu FIR filtru, viz. 3.1.1. Prvním krokem je tedy navzorkování této frekvenční charakteristiky. Čím hustější bude navzorkování, tím přesnější bude aproximace. Na tyto vzorky aplikujeme zpětnou Fourierovu transformaci. Po tomto kroku jsme získali impulsní odezvu $h[n]$. Posléze použijeme metodu váhování impulsní charakteristiky, viz. 3.1.2, kde impulsní odezvu $h[n]$ vynásobíme oknem $w[n]$. Takto získané koeficienty h vložíme do Toeplitzovy matice, která má obecný tvar

$$A = \begin{pmatrix} a_0 & a_{-1} & \cdots & \cdots & a_{-(n-1)} \\ a_1 & a_0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & a_0 & a_{-1} \\ a_{n-1} & \cdots & \cdots & a_1 & a_0 \end{pmatrix}. \quad (3.20)$$

Tuto matici vyplníme pomocí koeficientů h následovně

$$\begin{pmatrix} h[p] & h[p-1] & \cdots & \cdots & h[1] \\ h[p+1] & h[p] & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & \vdots \\ \vdots & & & & \ddots & \vdots \\ h[n] & h[n-1] & \cdots & \cdots & h[n-p] \end{pmatrix} = \begin{pmatrix} -h[p] \\ -h[p+1] \\ \vdots \\ \vdots \\ -h[n] \end{pmatrix}, \quad (3.21)$$

kde p je řád požadovaného filtru. Touto úpravou jsme získali soustavu tzv. Yule-Walkerových rovnic. Pomocí metody nejmenších čtverců vypočítáme jednotlivé koeficienty polynommické rovnice $A(z)$. Posledním krokem je případné překlopení těchto koeficientů tak, aby kořeny této rovnice ležely uvnitř jednotkové kružnice a tedy splňovaly podmínku stability filtru, viz. 2.2.3.

Nyní musíme získat čitatele přenosové funkce. Jeho výpočet probíhá pomocí těchto zjednodušených kroků:

- Výpočet polynomu čitatele, který odpovídá aditivní dekompozici frekvenční odezvy.
- Vyhodnocení kompletní frekvenční odezvy, která odpovídá polynomům čitatele a jmenovatele.
- Použití techniky spektrální faktorizace k získání impulsní odezvy filtru (více o spektrální faktorizaci viz. [9]).
- Získání polynomu čitatele pomocí metody nejmenších čtverců nad impulsní odezvou.

Výsledkem jsou tedy dva polynomy $B(z)$ a $A(z)$ přenosové funkce navrhovaného filtru. Podrobnější popis tohoto postupu můžete nalézt v [11].

Kapitola 4

Existující řešení

Nyní si představíme různá existující řešení, která sloužila jako vzor pro tuto práci. Vyjmenujeme si prvky, které by aplikace měla obsahovat. Objektivně je zhodnotíme a poté uvedeme, jaký vliv měly na výslednou aplikaci této práce.

4.1 Způsob hodnocení

Vyjmenujeme si požadované vlastnosti, podle kterých bude tvořeno hodnocení:

- Přehlednost aplikace.
- Intuitivní ovládání.
- Možnost vytvořit libovolný typ filtru.
- Vykreslení všech potřebných charakteristik.
- Interaktivní charakteristiky (možnost vytvořit filtry několika způsoby).

Dále popíšeme u každého příkladu jiné nedostatky, nebo naopak prvky nad rámec hodnocení.

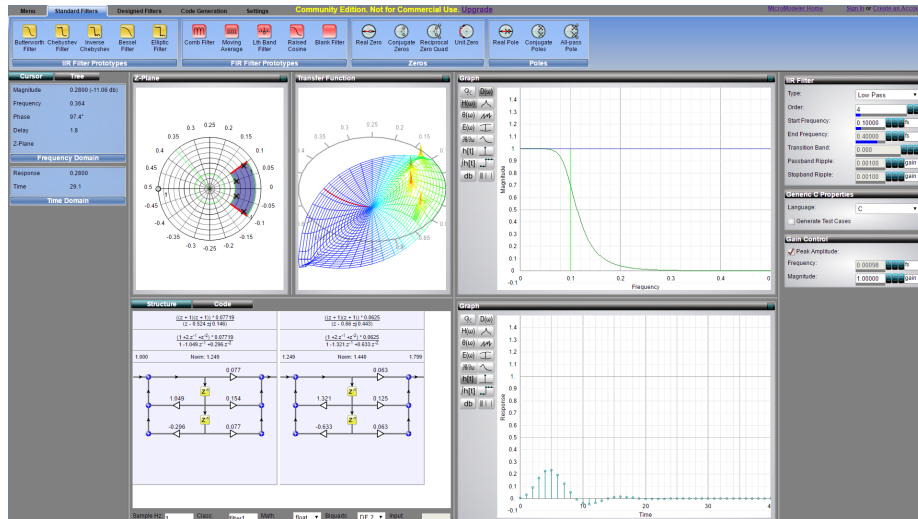
4.2 Vybraná řešení

Níže vyjmenovaná řešení byla vybrána podle toho, zda měla přínosnou hodnotu pro tuto práci. Do hodnocení nebyly zahrnuty desktopové aplikace, či moduly do programů typu Matlab.

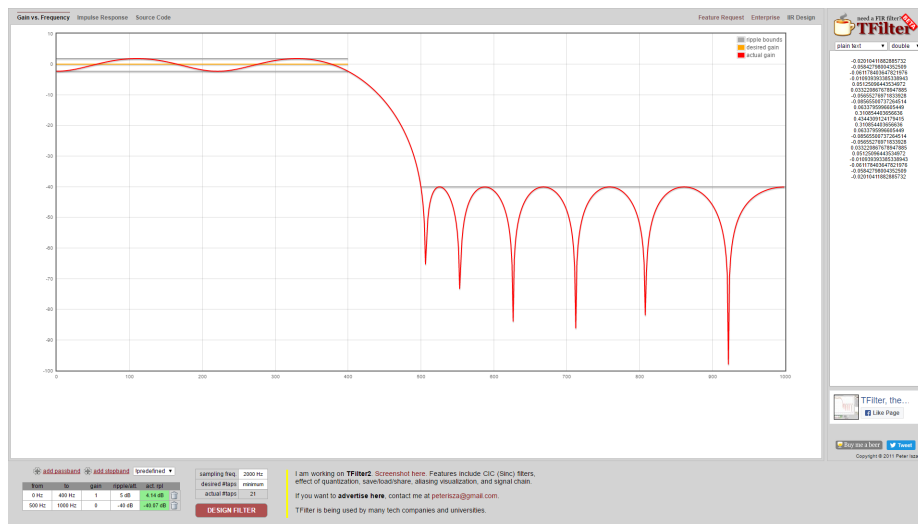
4.2.1 MicroModeler DSP

Pravděpodobně nejkomplexnější nástroj pro návrh filtrů. Výstup aplikace je velmi přehledný, zobrazuje či umožňuje zobrazit potřebné charakteristiky. Obsahuje nepřehledné množství příkladů i popis funkcionality, dále také obsahuje nemalý počet metod pro výpočet filtru. Filtr je možné vytvořit mnoha způsoby, od vložení nul a pólů po nakreslení ideální frekvenční charakteristiky. Nepřehlednost a složité ovládání aplikace jsou jedinými výtkami, není lehké najít potřebnou techniku či rychle vložit do grafu body.

Viz. obr. 4.1, dostupné na: <http://www.micromodeler.com/dsp/>.



Obrázek 4.1: MicroModeler DSP

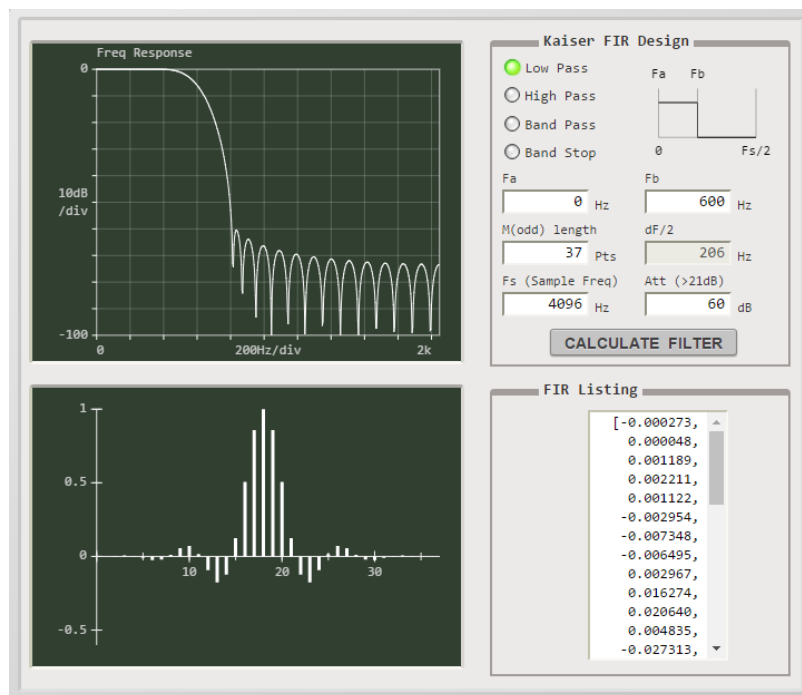


Obrázek 4.2: TFilter

4.2.2 TFilter

Zde se opět setkáváme s nástrojem tvořící pouze FIR filtr, používá Parks–McClellan algoritmus pro návrh. Aplikace je relativně přehledná, nastavení filtru je trochu schované, vykreslované grafy zabírají přespříliš místa. Ovládání je jednoduché, najdeme zde předdefinované filtry, ale také možnost vytvořit si vlastní. Při vytváření filtru je obtížné zadávání pásem, krajní hodnoty se musejí zadávat jako frekvence do tabulky, nelze je zadat přímo do grafu. Tím byla i nastíněna interaktivita grafů, která se zde neobjevuje. Také tu chybí vykreslení nul a pólů.

Viz. obr. 4.2, dostupné na: <http://t-filter.engineerjs.com/>.



Obrázek 4.3: Kaiser-Bessel filter generator

4.2.3 Kaiser-Bessel filter generator

Tato aplikace demonstruje návrh FIR filtru. Používá kombinaci metod vzorkování frekvenční charakteristiky a váhování impulsní charakteristiky. Grafy jsou přehledné, ale chybí zobrazení nul a pólů, také chybí graf s tvarem okna. Není možné pracovat s grafy. Neumožňuje výběr jiného okna než Kaiserova. Velkým kladem je detailní popis algoritmu a úryvky kódu.

Viz. obr. 4.3, dostupné na: <http://www.arc.id.au/FilterDesign.html>.

4.2.4 Bakalářská práce Ondřeje Novotného

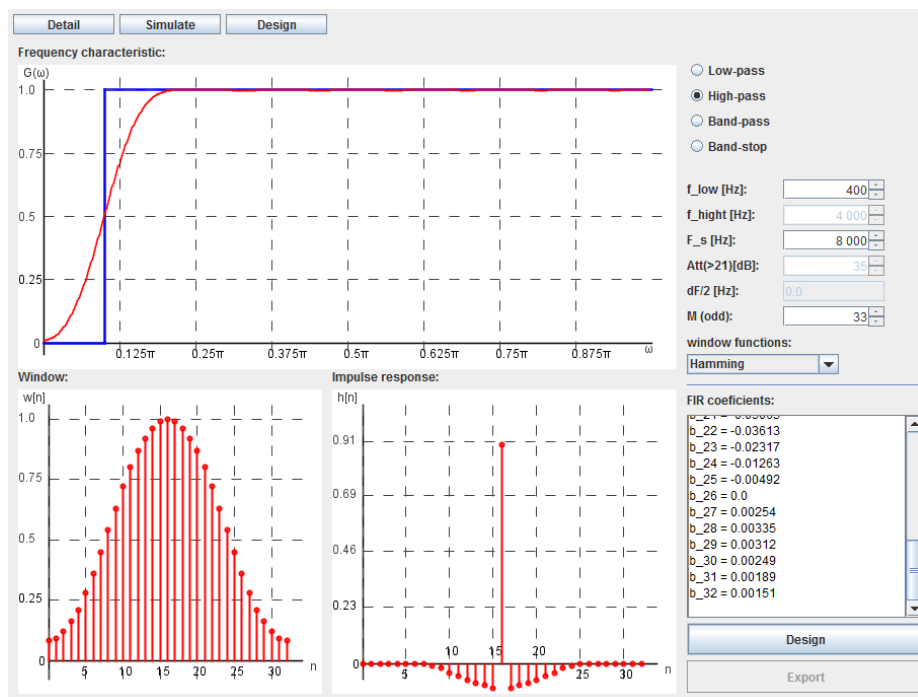
Stejně jako předchozí řešení i tato aplikace se zaměřuje pouze na návrh FIR filtru. Rozhraní je přehledné, ovládání je také rychle osvojitelné. V samotném návrhu nezobrazuje nuly a póly, ostatní charakteristiky jsou precizně vykreslené. Obsahuje několik pevných a parametrických oken. Grafy nejsou interaktivní. Největším záporem je aplikační prostředí, nejedná se o webové řešení.

Z důvodů nekompatibility s nejnovější verzí jazyka Java a bezpečnostní politikou prohlížečů již tato demonstrace není dostupná jako webový applet. Jedinou možností je stáhnutí a používání desktopové aplikace. Tato práce by tuto aplikaci měla nahradit a rozšířit opravdovou webovou aplikací.

Viz. obr. 4.4, dostupné na: <http://www.fit.vutbr.cz/study/courses/ISS/public/>.

4.3 Shrnutí

Prvně jmenovaný MicroModeler DSP splňuje naprosto veškeré stanovené požadavky. Na rozdíl od ostatních řešení obsahuje velké množství návrhů IIR filtru, zobrazuje veškeré



Obrázek 4.4: Bakalářská práce Ondřeje Novotného

charakteristiky a obsahuje interaktivitu grafů. V ostatních řešeních jsou také různé nápady, kterými by se tato práce mohla řídit. V žádném řešení nebyly objeveny prvky, kterých by se výsledná aplikace měla absolutně vyvarovat.

Kapitola 5

Popis implementace

V této kapitole si popíšeme implementační detaily aplikace z pohledu návrhu, samotné implementace a výsledného programu. Aplikace má sloužit k demonstraci dříve popisovaných návrhových metod filtrů. Počítá se základními znalostmi o signálech a filtraci. Tato demonstrace je určena k prohloubení těchto znalostí na základě interaktivního návrhu filtru. Musí být možné měnit různá nastavení a navrhovat filtr více způsoby, především představit veškeré možné charakteristiky a dynamicky zobrazovat jejich vztah k filtru samotnému. Pro pochopení výpočtů, které aplikace provádí, jsou dodány příklady se vzorci a výpočty.

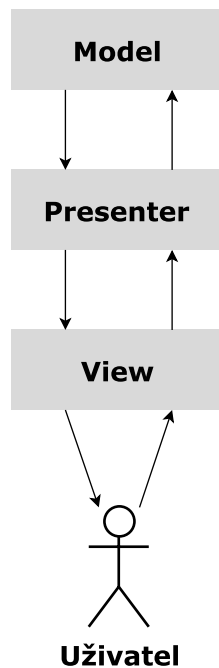
5.1 Implementační prostředí

Zadáním bylo vytvořit webovou aplikaci. Vybral jsem tedy jazyk JavaScript, a to z mnoha důvodů. Především je to podpora prohlížečů, aby tedy bylo možné pustit aplikaci v jakémkoliv moderním prohlížeči. Dále je také výhodný v tom, že není potřeba stahovat žádné dodatečné soubory. Veškeré výpočty provádí samotný prohlížeč, není tedy potřebné řešit výkonnost či vytížení serverové strany. Výhodou také může být transparentnost zdrojových kódů, kdokoliv by se zajímal o tuto tematiku konkrétněji, může do nich nahlédnout k lepšímu porozumění problematiky. Pro grafické uživatelské rozhraní byl použit jazyk HTML v kombinaci se stylovým předpisem v jazyce CSS.

5.2 Návrh aplikace

Prvním krokem před implementací bylo vytvoření mockupu, zobrazující detailní rozložení grafických prvků v aplikaci. Posléze bylo nutné zvolit návrhový vzor. Logicky se nabízel návrhový vzor MVP, který aplikaci rozděluje do tří logických celků a to: Model, View, Presenter. Model je jádro aplikace, zde se provádí veškeré výpočty. View slouží k převodu dat modelu do podoby vhodné k interaktivní prezentaci uživateli. Presenter reaguje na události od uživatele a provádí potřebné úkony ve View či v Modelu.

Program bude obsahovat nabídku s možnými návrhovými metodami a různými příklady. V návrhu filtru budou vykresleny potřebné charakteristiky, nastavení filtru, výpis koeficientů a přenosové funkce filtru. Návrh se bude ovládat pomocí charakteristik, tedy vykreslení ideální frekvenční charakteristiky, nebo umístění nul a pólů přenosové funkce na jednotkové kružnici. Bude možné zvolit přednastavený filtr, řád filtru, případně u FIR filtru okenní funkci. Veškeré prvky návrhu filtru bude možné zvětšit na celou obrazovku pro detailnější pohled.



Obrázek 5.1: Návrhový vzor Model-View-Presenter

5.3 Implementace

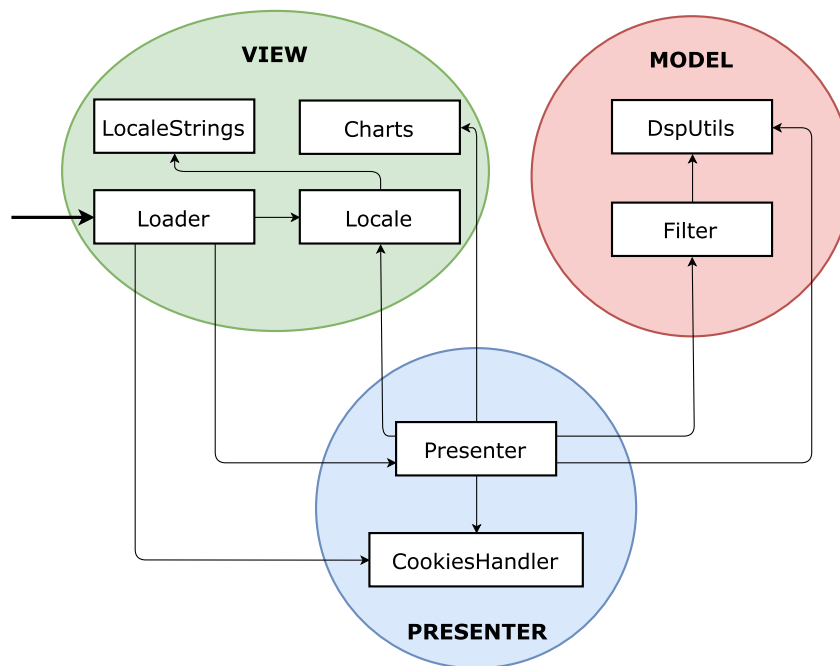
Nyní si rozebereme nejdůležitější objekty, které umožňují chod aplikace. Tyto objekty si logicky rozdělíme podle návrhového vzoru, který byl popsán v předchozí sekci. Graficky znázorněná závislost implementačních prvků je na obr. 5.2

5.3.1 Loader, Locale, Charts

Loader je prvotní objekt, který pomocí šablon nahraje veškeré prvky grafického rozhraní. Šablony jsou definovány jako HTML soubory pro jednotlivé části webové stránky. Data se do těchto šablon nahrají podle zvoleného jazyka, funkce pro získání potřebných dat jsou obsaženy v objektu **Locale** a veškerá textová data jsou uložena v souboru **LocaleStrings**. Dále je zde obsažen objekt **Charts**, který uchovává předpis pro vykreslení jednotlivých grafů. Grafy je nutné od sebe rozlišovat, každá charakteristika má jiný vzhled či popis os. Vstupem je pole číselných hodnot, tedy vzorky jednotlivých charakteristik.

5.3.2 Controller, CookiesHandler

Controller je hlavní ovládací prvkem této aplikace. Přemostuje komunikaci mezi modelem a pohledem. Především volá funkce pro výpočet filtru a poté funkce pro výpočet veškerých charakteristik. Dále také vkládá do grafického rozhraní většinu dynamických prvků a především grafickým prvkům přidává funkce na odposlech událostí. **CookiesHandler** je objekt pracující s cookies webového prohlížeče, například umožňuje nastavit jazyk aplikace tak, aby se i po opuštění stránky stále zobrazoval tento zvolený jazyk.



Obrázek 5.2: Zjednodušený diagram tříd.

5.3.3 Filter, DspUtils

V těchto komponentách je obsažena veškerá výpočtová logika. Objekt `Filter` obsahuje výpočetní algoritmy pro získání koeficientů filtru, vstupem je typ filtru a požadovaná metoda návrhu. V případě návrhu filtru pomocí nul a pólů mapuje kořeny na polynomicovou rovnici. `DspUtils` je neméně důležitý objekt, obsahuje velké množství pomocných funkcí pro výpočet filtru. Například vzorkování frekvence, výpočet impulsní odezvy, frekvenční odezvy, nul a pólů. Také obsahuje veškeré okenní funkce popsané v této práci, dále funkci pro filtrování signálu a stabilizování polynomické rovnice k jednotkové kružnici.

5.3.4 Použité knihovny

Nyní si popíšeme knihovny a jejich funkce, které byly v této práci využity.

- Highcharts [7] – tato knihovna slouží k vykreslení veškerých grafů, jsou zde také využity dva přídatné moduly. První modul umožňuje pohyb jednotlivých bodů, v druhém případě přidává funkci pro exportování grafů.
- Jmat [12] – matematická knihovna, která zjednodušuje práci s maticemi a vektory. Také je zde důležité zdůraznit, že obsahuje algoritmus pro výpočet polynomických rovnic pomocí nejmenších čtverců.
- jQuery-templating [4] – jak již název napovídá, tato knihovna je používána k šablonování, tedy naplnění grafického rozhraní textovými daty.
- jQuery-loading [3] – drobný doplněk pro knihovnu jQuery, obsahuje funkce pro zobrazování načítacích prvků.

- jQuery [6] – jedna z nejrozšířenějších knihoven pro jazyk JavaScript, obsahuje ne-přeberné množství funkcí a rozšíření, především zjednodušuje a optimalizuje přístup k DOM elementům internetových stránek.
- KaTeX [1] – interpretující knihovna pro jazyk LaTeX, výstupem je stylizovaný HTML kód, který se podobá výstupu LaTeXu.
- NumericJS [10] – v této knihovně lze nalézt velké množství matematických funkcí, v této práci je především využíván algoritmus pro výpočet rychlé Fourierovy transformace.
- Prism [13] – tato knihovna je používána pro stylizaci zdrojového kódu jazyka C, jako je například zvýraznění syntaxe nebo číslování řádků.

5.3.5 Problémy

Při implementaci se ukázalo, že jazyk Javascript není ideální variantou pro výpočet filtrů. Algoritmy jsou při návrhu vyšších řádů filtrů velmi pomalé. Nevýhodou je také neoptimalizovanost výpočtů přes vektory a matice. Z těchto důvodů nebylo možné implementovat návrhovou metodu popsanou v 3.2.1.

5.4 Výsledná aplikace a její použití

Nyní si popíšeme výslednou aplikaci, především se zaměříme na ovládání a použití. Nutno podotknout, že aplikace má mít především demonstrativní účel. Není tedy zahlcena množstvím textu, má interaktivně zobrazovat fungování filtrů a propojení jednotlivých charakteristik.

5.4.1 Ovládání

Výsledná aplikace je ovládána především myší, v několika případech se dá také použít vstup z klávesnice. Ovladatelné komponenty si zde vyjmenujeme tak, jak jsou vyobrazena na obr. 5.3:

1. tahatelné body ideální frekvenční charakteristiky
2. rozbalovací nabídka s přednastavenými filtry
3. číselník pro výběr řádu filtru
4. tlačítka sloužící k zvětšení a skrytí widgetu
5. tlačítko umožňující uložení grafu do souboru
6. interaktivní graf, kde je možné nastavit polohu nul a pólů
7. v případě návrhu FIR filtru, rozbalovací nabídka pro výběr okenní funkce

Číselné a rozbalovací nabídky se dají ovládat klávesnicí. Zbylé komponenty je potřebné ovládat myší. Po jakékoliv změně v nastavení filtru či po potáhnutí s jakýmkoliv bodem v grafu proběhne automatické vypočítání filtru a překreslení charakteristik. Dále je také možné v horní navigaci změnit jazyk demonstrace.



Obrázek 5.3: Charakteristiky a nastavení filtru

5.4.2 Výstup

Výstupem aplikace jsou vykreslené charakteristiky (viz. obr. 5.3), kterýkoliv z těchto grafů se dá uložit do souboru pro pozdější využití. Další výstupy jsou zobrazeny na obr. 5.4, především je to přenosová funkce navrhovaného filtru. Také zde jsou vypsané koeficienty filtru, které lze zkopírovat pro použití v programech typu MATLAB apod. Posledním výstupem je kód v jazyce C s koeficienty filtru a funkcí pro filtraci signálu.

5.4.3 Scénář použití

Po spuštění aplikace se zobrazí úvodní stránka s popisem ovládání a krátkým popisem o funkcionalitě demonstrace. V levé části je navigační nabídka, zde je možné si vybrat z návrhových metod pro FIR nebo IIR filtry. Po zvolení požadované návrhové metody se zobrazí stránka s návrhem filtru. Vždy se zobrazí přednastavený filtr. Filtr se dá specifikovat třemi způsoby. Tím prvním a pravděpodobně nejdůležitějším je nastavení bodů v grafu frekvenční charakteristiky, kterými nastavíme ideální frekvenční charakteristiku. Další možností je zvolení přednastavených filtrů. Poslední možností je poloha nul a pólů, kterými specifikujeme koeficienty přenosové funkce. Parametry filtru se dají nastavit pomocí řádu přenosové funkce a v případě FIR filtru výběrem okenní funkce.

Pro pochopení jednotlivých charakteristik nebo konkrétních návrhových metod jsou v demonstraci dostupné příklady. V nich jsou zobrazeny vzorce a jednotlivé výpočty tak, aby uživateli bylo jasné, jak se došlo k výsledku.

5.4.4 Dostupnost

Demonstrace je dostupná na školním serveru

<https://www.stud.fit.vutbr.cz/~xpospi87/IBP/>.

Přenosová funkce

$$H(z) = \frac{0.00085 - 0.00004z^{-1} - 0.0012z^{-2} + 0.00051z^{-3} + 0.00250z^{-4} - 0.00127z^{-5} - 0.00422z^{-6} + 0.00388z^{-7} + 0.00758z^{-8} - 0.00749z^{-9} - 0.01022z^{-10} + 0.01604z^{-11} + 0.01533z^{-12} - 0.02692z^{-13} - 0.01662z^{-14} + 0.05177z^{-15} + 0.02253z^{-16} - 0.09457z^{-17} - 0.01513z^{-18} + 0.52490z^{-19} + 0.30439z^{-20} - 0.01993z^{-21} + 0.00775z^{-22}}{1.00000z^{-22}}$$

Koeficienty filtru

B = [0.00085, 0.00004, -0.00120, 0.00051, 0.00250, -0.00127, -0.00422, 0.00388, 0.00758, -0.00749, -0.01022, 0.01604, 0.01533, -0.02692, -0.01662, 0.05177, 0.02253, -0.09457, -0.01513, 0.32537, 0.52490, 0.30439, -0.03495, -0.09927, 0.02223, 0.04709, -0.02150, -0.02794, 0.01491, 0.01390, -0.01993, -0.00775, 0.00724, 0.00302, -0.00472, -0.00134, 0.00234, 0.00025, -0.00132, -0.00007, 0.00077]

A = [1.00000]

C zdrojový kód

```

1 #include <stdio.h>
2
3 #define COEFFS_COUNT 41
4
5 double B[COEFFS_COUNT] = { 0.0008530353849811413, -0.000036550883581556, -0.0011976769424509814, 0.0005071763133866572, -0.0025032213044179873, -0.0012711448749586829, -0.004215183690837971, 0.0038770152883704106, 0.007575892728914665, -0.007491911111111111, -0.01022222222222222, 0.01604444444444444, 0.01533333333333333, -0.02692222222222222, -0.01662222222222222, 0.05177222222222222, 0.02253333333333333, -0.09457222222222222, -0.01513333333333333, 0.3253722222222222, 0.5249022222222222, 0.3043922222222222, -0.03495222222222222, -0.09927222222222222, 0.02223222222222222, 0.04709222222222222, -0.02150222222222222, -0.02794222222222222, 0.01491222222222222, 0.01390222222222222, -0.01993222222222222, -0.00775222222222222, 0.00724222222222222, 0.00302222222222222, -0.00472222222222222, -0.00134222222222222, 0.00234222222222222, 0.00025222222222222, -0.00132222222222222, -0.00007222222222222, 0.00077222222222222 };
6 double A[COEFFS_COUNT] = { 1 };
7
8 double Memory_B[COEFFS_COUNT + 1];
9 double Memory_A[COEFFS_COUNT + 1];
10
11 double filter (double input) {
12
13     int i;
14     double output;
15
16     Memory_B[0] = input;
17     output = 0.0;
18
19     for (i = COEFFS_COUNT - 1; i >= 0; i--)
20     {
21         output += Memory_B[i] * B[i];
22         Memory_B[i + 1] = Memory_B[i];
23     }

```

Obrázek 5.4: Koeficienty filtru, kód v jazyce C

Kapitola 6

Hodnocení aplikace

Pro vyhodnocení aplikace byl vytvořen formulář, do kterého se vyplňovaly odpovědi. Tento formulář byl zveřejněn na sociální síti, ve skupině speciálně určené pro studenty VUT FIT.

6.1 Formulář

Formulář byl vytvořen pro získání zpětné vazby od uživatelů. Jeho cílem bylo zjistit, zda je aplikace uživatelsky přívětivá a má nějaký přínos k dané problematice. Otázky byly jednoduché, většinou založené pouze na číselném hodnocení (1 – nejlepší, 5 – nejhorší). Z hlediska kompatibility aplikace byla velmi důležitá první otázka, tedy zda se uživateli načetla aplikace korektně. V dnešní době se využívá široká škála internetových prohlížečů a každý implementuje podporu JavaScriptu rozdílně. Je důležité zjistit, v jakých prohlížečích a případně v jakých verzích prohlížeče aplikace nefunguje správně. Nyní si představíme jednotlivé otázky tohoto formuláře:

1. Zobrazila se Vám webová stránka korektně?
2. Jak hodnotíte vzhled aplikace? (1 – 5)
3. Jak hodnotíte ovládání aplikace? (1 – 5)
4. Jak hodnotíte výstup aplikace (charakteristiky, koeficienty apod.)? (1 – 5)
5. Jak hodnotíte přínos aplikace pro předmět ISS? (1 – 5)
6. Jak hodnotíte dodané příklady? (1 – 5)
7. Co se Vám na demonstraci líbí?
8. Co se Vám na demonstraci nelíbí?
9. Připomínky

6.2 Vyhodnocení formuláře

Formulář byl vyplněn 11 lidmi. Textové otázky nebyly povinné, z tohoto důvodu byly vyplněny pouze menšinou dotazujících. Hodnotící otázky jsou shrnuty v tabulce 6.1, kompletní odpovědi naleznete v příloze B.

Číslo otázky	Otázka	Průměrné hodnocení
2	Vzhled	1,5
3	Ovládání	1,7
4	Výstup	1,6
5	Přínos	1,7
6	Příklady	1,8

Tabulka 6.1: Průměrné výsledky hodnotících otázek

Aplikace byla přijata převážně kladně. Nejdůležitějším aspektem bylo, že všem zúčastněným fungovala aplikace korektně. Nejlépe byl hodnocen vzhled a výstup aplikace. Z textových odpovědí vyplývá, že aplikace nemá vyložený nedostatek, uživatelé pouze navrhovali drobné změny či rozšíření.

Hodnocení bylo mírným zklamáním, bylo vystaveno na sociální síti, na které to mohlo vidět několik stovek studentů. Očekával jsem větší zpětnou vazbu, včetně většího množství zodpovězených textových otázek.

Kapitola 7

Závěr

Primárním cílem této práce bylo vytvořit webovou demonstraci návrhu lineárních filtrů. Na počátku jsem si zopakoval teorii dané problematiky z předmětu ISS, dále jsem také prostudoval různé články a několik knih. Vyzkoušel jsem si dostupné demonstrace na internetu, ohodnotil je a z těchto hodnocení se inspiroval při samotné implementaci. Při návrhu aplikace jsem postupoval dle moderních návrhových metod. V implementační části jsem si prohloubil znalosti jazyka JavaScript.

7.1 Budoucí práce

Do budoucna by se v práci mohlo vytvořit několik rozšíření. Bylo by vhodné, kdyby bylo možné si vytvořené filtry uložit. S tím také souvisí možnost vložení přenosové funkce, tedy pouze zobrazení charakteristik pro již vytvořený filtr. Samozřejmě se také nabízí rozšíření počtu návrhových metod pro FIR i IIR filtry. Posledním, ale neméně zásadním vylepšením demonstrace by byla možnost ručního vykreslení ideální frekvenční charakteristiky.

Literatura

- [1] Academy, K.: KaTeX. <https://khan.github.io/KaTeX/>, [online], [cit. 2017-05-03].
- [2] Bičák, J.: *Návrh číslicových filtrů*. [online], [cit. 2017-05-03].
URL <http://hippo.feld.cvut.cz/vyuka/nds/pz.pdf>
- [3] Bonetti, C.: jQuery-loading. <http://carlosbonetti.github.io/jquery-loading/>, [online], [cit. 2017-05-03].
- [4] Burgess, P.: jQuery-templating. <http://codepb.github.io/jquery-template/>, [online], [cit. 2017-05-03].
- [5] Collins, A.: *Digital Filter Design*. [online], [cit. 2017-05-03].
URL <http://arc.id.au/FilterDesign.html>
- [6] Foundation, J.: jQuery. <https://jquery.com/>, [online], [cit. 2017-05-03].
- [7] Honsi, T.: Highcharts. <https://www.highcharts.com/>, [online], [cit. 2017-05-03].
- [8] Jiří, J.: *Číslicová filtrace, analýza a restaurace signálů*. VUTIUUM, 2002, ISBN 80-214-1558-4.
- [9] Lall, S.: *Spectral Factorization*. [online], [cit. 2017-05-03].
URL http://sepwww.stanford.edu/sep/prof/fgdp3/fgdp_03.pdf
- [10] Loisel, S.: Numeric Javascript. <http://www.numericjs.com/>, [online], [cit. 2017-05-03].
- [11] The MathWorks, I.: *yulewalk – Recursive digital filter design*. [online], [cit. 2017-05-03].
URL <https://www.mathworks.com/help/signal/ref/yulewalk.html>
- [12] Vandevenne, L.: Jmat.js. <https://github.com/lvandeve/jmat>, [online], [cit. 2017-05-03].
- [13] Verou, L.: Prism. <http://prismjs.com/>, [online], [cit. 2017-05-03].
- [14] Vích, R.; Smékal, Z.: *Číslicové filtry*. Academia, 2000, ISBN 80-200-0761-X.
- [15] Černocký, J.: *Systémy s diskrétním časem*. [online], [cit. 2017-05-03].
URL http://www.fit.vutbr.cz/study/courses/ISS/public/pred/11_disk_syst/disk_syst.pdf

Příloha A

Obsah CD

- BP.pdf – elektronická verze tohoto dokumentu
- doc – zdrojový kód této práce
- src – zdrojové kódy webové aplikace

Příloha B

Výsledky formuláře

V tabulce [B.1](#) jsou zaznamenaná veškerá hodnocení otázek. Dále také dokládám jednotlivé textové odpovědi.

B.1 Co se Vám na aplikaci líbí?

- Hezké grafy a závislost mezi nimi.
- prevedenie
- intuitivní, vzhled, interaktivní, výsledky v reálném čase, jednoduché, cz i en jazyk

B.2 Co se Vám na aplikaci nelíbí?

- Myslím si, že by mohla obsahovat více návrhových metod.
- ak to je volne dostupne fajna inak nevidim dovod to pouzít oproti matlabu
- drobnosti, možná výsledné koeficienty a transfer funkce by mohly být vypsané jinak než na jeden super dlouhý řádek

B.3 Připomínky

- dost dobrá demonstrace, bylo by fajna možná přidat ukázkou filtrace předdefinovaného signálu navrženým filtrem, časem třeba přidat 2D filtry atd.

Čas hodnocení	1. Funkce	2. Vzhled	3. Ovládání	4. Výstup	5. Přínos	6. Příklady
7.5.2017 15:56	Ano	1	2	2	1	2
7.5.2017 20:31	Ano	1	1	2	1	1
12.5.2017 10:46	Ano	1	1	2	1	1
14.5.2017 22:24	Ano	1	1	1	2	3
14.5.2017 22:26	Ano	1	1	1	1	1
14.5.2017 22:35	Ano	2	2	1	1	1
14.5.2017 22:41	Ano	2	3	2	3	2
14.5.2017 23:07	Ano	1	1	1	4	1
14.5.2017 23:17	Ano	1	1	1	1	4
15.5.2017 0:34	Ano	5	5	3	2	3
15.5.2017 15:32	Ano	1	1	2	2	1
	100%	1,5	1,7	1,6	1,7	1,8

Tabulka B.1: Výsledky hodnotících otázek