

# 1 ČÍSLICOVÉ FILTRE A FFT (OPAKOVANIE)

## 1.1 ÚVOD

**Číslicové filtre** a efektívne algoritmy výpočtu **DFT**, ktoré využívajú rýchle algoritmy výpočtu založené na algoritme **FFT** patria medzi základné algoritmy číslicového spracovania signálov (ČSS). Všetky prakticky využívané číslicové signálové procesory (DSP) majú architektúru optimalizovanú práve pre tieto dva základné typy algoritmov ČSS. Pomocou týchto algoritmov je možné demonštrovať prakticky všetky podstatné vlastnosti jadra klasických DSP, čo bude náplňou ďalších cvičení.

Vzhľadom na to, že problematika číslicových filtrov a FFT je podrobne preberaná v iných špecializovaných predmetoch, cieľom tohto cvičenia je zopakovanie základných pojmov z tejto oblasti a opísanie **niektorých praktických metód návrhu** číslicových filtrov v prostredí Matlab (v ďalších častiach budú opisované príkazy Matlabu v. 4.21c s nainštalovaným Signal Processing Toolboxom v. 3.0b, prípadne grafického rozhrania tzv. Digital Filter Package [1]) resp. iných špecializovaných programov (program FilterExpress od firmy Systolix [2]). Aj keď balíky [1] a [2] sú voľne dostupné, umožňujú navrhnuť a analyzovať mnohé praktické číslicové filtre. V prípade extrémnych nárokov na parametre navrhovaných filtrov resp. ich analýzu je možné použiť komerčne dostupné balíky – napr. [3] (tento toolbox Matlabu však nie je na KEMT dostupný).

## 1.2 ČÍSLICOVÉ FILTRE

### 1.2.1 FILTRE S KONEČNOU IMPULZOVOU ODPOVEĎOU

**Filtre s konečnou impulzovou odpoveďou** (FIR- Finite Impulse Response) patria medzi najjednoduchšie číslicové filtre ako z teoretického, tak aj z realizačného hľadiska. FIR filter  $N$ -tého rádu je možné opísať **prenosovou funkciou**

$$H_{FIR}(z) = \sum_{k=0}^N h[k]z^{-k} \quad (1.1)$$

pričom  $h[k]$  sú **koefficienty** FIR filtra. **Frekvenčná charakteristika**  $H^f(\theta)$  FIR filtra (**komplexná funkcia** reálnej premennej  $\theta$ ) je určená vzťahom

$$H^f(\theta) = H_{FIR}(z) \Big|_{z=e^{j\theta}} \quad (1.2)$$

pričom  $\theta \in \langle -\pi, \pi \rangle$  (alebo  $\theta \in \langle 0, 2\pi \rangle$ ) je normalizovaná frekvencia

$$\theta = 2\pi \frac{F}{F_{\text{samp}}} \quad (1.3)$$

a  $F_{\text{samp}}$  je **frekvencia vzorkovania**.

Keďže frekvenčná charakteristika je komplexnou funkciou, v praxi sa využívajú amplitúdové a fázové frekvenčné charakteristiky, ktoré sú reálnymi funkciami reálnej premennej  $\theta$ . V prostredí Matlab je možné zobrazit' obidve charakteristiky pomocou príkazu *freqz*.

### Príklad

Zobrazte v prostredí Matlab amplitúdovú a fázovú charakteristiku FIR filtra s koeficientmi  $h_i = 1.0$ ,  $i = 0, 1, \dots, 10$ .

### Riešenie

Príklad vyriešime v prostredí Matlab zapísaním nasledujúcich príkazov

```
h=[1 1 1 1 1 1 1 1 1 1 1];
freqz(h);
```

Medzi charakteristické vlastnosti FIR filtrov patrí ich **stabilita** (t.j. kvantovaním koeficientov sa FIR filter nemôže stať nestabilným) a možnosť realizácie FIR **filtra s lineárnou fázovou charakteristikou**. Nutnou a postačujúcou podmienkou aby FIR filter mal lineárnu fázovú charakteristiku je symetria jeho koeficientov, ktorú je možné vyjadriť v tvare

$$h[k] = h[N - k], \quad k = 0, 1, \dots, N \quad (1.4)$$

## 1.2.2 FILTRE S NEKONEČNOU IMPULZOVOU ODPOVEĎOU

Pre **filtre s nekonečnou impulzovou odpoveďou** (IIR – Infinite Impulse Response) je charakteristická prítomnosť spätnej väzby<sup>1</sup>, čo sa v prípade prenosovej funkcie prejaví existenciou menovateľa prenosovej funkcie  $H_{\text{IIR}}(z)$ , ktorý je na rozdiel od  $H_{\text{FIR}}(z)$  rôzny od 1 (pre jednoduchosť uvažujme, že rád čitateľa a menovateľa sú rovnaké):

$$H_{\text{IIR}}(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_N z^{-N}}{1 + a_1 z^{-1} + \dots + a_N z^{-N}} \quad (1.5)$$

Implementáciu IIR filtra opísaného prenosovou funkciou (1.5) je možné realizovať viacerými spôsobmi. Najznámejšie a v praxi najčastejšie využívané sú realizácie založené na rozklade (1.5) do sekcií druhého rádu, tzv. **bikvadov** podľa vzťahu (za predpokladu, že  $N$  je párne)

<sup>1</sup> Existuje však aj možnosť realizácie FIR filtrov pomocou štruktúr so spätnou väzbou. Typickým príkladom je realizácia FIR filtra v tvare  $y(n) = y(n-1) + x(n) - x(n-N)$ , ktorý za predpokladu vhodných počiatočných podmienok realizuje FIR filter  $y(n) = x(n) + x(n-1) + \dots + x(n-N+1)$ .

$$H_{IIR}(z) = \prod_{k=1}^{N/2} \left( \frac{b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2}}{1 + a_{1k}z^{-1} + a_{2k}z^{-2}} \right) = \prod_{k=1}^{N/2} H_k(z) \quad (1.6)$$

pričom prenosová funkcia  $k$ -tého bikvadu  $H_k(z)$  je určená vo všeobecnom prípade piatimi koeficientmi  $b_{0k}, b_{1k}, b_{2k}, a_{1k}, a_{2k}$ . IIR filtre sú charakteristické predovšetkým podmienenou stabilitou, pričom po kvantovaní koeficientov sa IIR filter môže stať nestabilným. Jedným z hlavných dôvodov rozkladu IIR filtra na kaskádne zapojenie bikvadov je predovšetkým podstatne nižšia citlivosť kaskády bikvadov na kvantovanie koeficientov. Navyiac v prípade bikvadu je možné jednoznačne definovať oblasť v ktorej sa musia nachádzať koeficienty, aby bol bikvad stabilný.

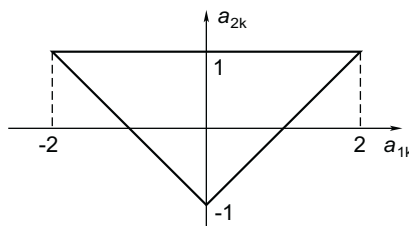
Nutnou podmienkou stability bikvadu je, aby koeficienty menovateľa  $H_k(z)$  spĺňali podmienku

$$|a_{jk}| < 2.0 \quad j=1,2 \quad k=1,2,\dots,N/2 \quad (1.7)$$

Postačujúcou podmienkou stability je splnenie vzťahov

$$-1 < a_{2k} < 1, \quad 1 + a_{1k} + a_{2k} > 0, \quad 1 - a_{1k} + a_{2k} > 0 \quad (1.8)$$

ktoré definujú tzv. trojuholník stability zobrazený na Obr.1.



Obr.1 Trojuholník stability bikvadu, body vo vnútri trojuholníka zodpovedajú stabilnému bikvadu

Z pohľadu implementácie pomocou DSP s pevnou rádovou čiarkou je hodnota 2.0 vo vzťahu (1.7) určitým problémom<sup>2</sup> (pretože interval čísel je u typických DSP obmedzený na  $(-1,1)$ ). V prípade koeficientov  $a_i \quad i=1,2,\dots,N$  vo vzťahu (1.5) nie je možné sformulovať na kontrolu stability podmienky, ktoré by boli také jednoduché ako podmienka (1.7).

Na implementáciu sekcie druhého rádu je možné využiť niekoľko realizácií. Najznámejšia je **priama forma I**, ktorú je možné opísať diferencnou rovnicou (index  $k$  je pre jednoduchosť vynechaný)

$$y(n) = \sum_{j=0}^2 b_j x(n-j) - \sum_{j=1}^2 a_j y(n-j) \quad (1.9)$$

<sup>2</sup> Štandardné DSP tento problém riešia s využitím dodatočných módov aritmetickej jednotky. V ďalších cvičeniach bude riešenie tohto problému opísané pre ADSP 218x/219x od firmy Analog Devices.

Pri technickej realizácii sú často využívané realizácie (tzv. **kanonické formy**), ktoré využívajú minimálny počet stavebných prvkov (napr. oneskorovacích členov, násobičiek, ...). Medzi najčastejšie využívané patria **kanonická forma I** (tzv. **transponovaná priama forma**) opísaná diferenčnou rovnicou

$$y(n) = b_0x(n) + w_1(n-1) \quad (1.10)$$

$$w_1(n) = b_1x(n) - a_1y(n) + w_2(n-1) \quad (1.11)$$

$$w_2(n) = b_2x(n) - a_2y(n) \quad (1.12)$$

a **kanonická forma II** (tzv. **priama forma II**) opísaná diferenčnou rovnicou

$$w(n) = x(n) - a_1w(n-1) - a_2w(n-2) \quad (1.13)$$

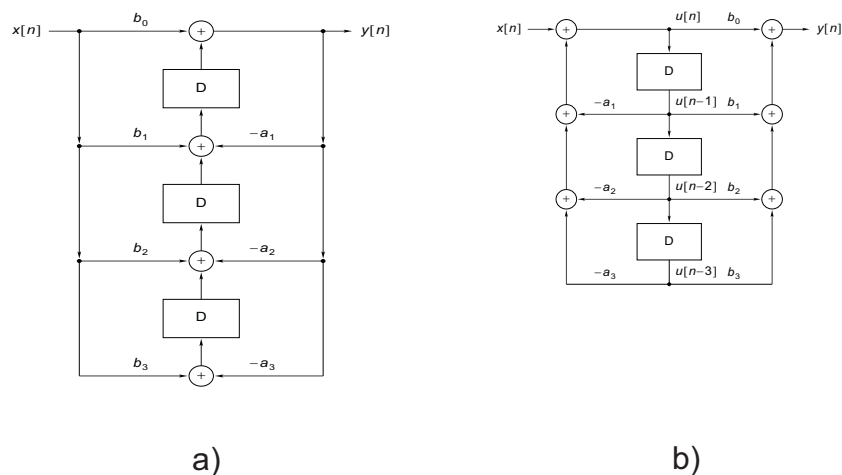
$$y(n) = \sum_{j=0}^2 b_j w(n-j) \quad (1.14)$$

Výhodnou vlastnosťou IIR filtrov je menší počet koeficientov v porovnaní s FIR filtermi, ktoré sú potrebné na realizáciu filtra s určitými vlastnosťami (napr. požadovanou strmost'ou medzi priepustným a nepriepustným pásmom).

### Príklad

Na základe rovníc (1.10) - (1.14) nakreslite realizáciu kanonických foriem I a II pre IIR filter rádu  $N = 3$  (realizáciu bikvadu dostaneme pre  $N = 2$  ).

### Riešenie



Obr.2 Kanonické formy – a) forma I, b) forma II

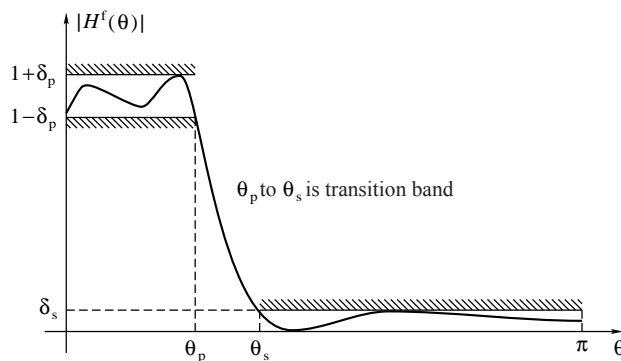
**Príklad**

Na základe rovnice (1.9) nakreslite realizáciu priamej formy<sup>3</sup>.

**1.2.3 NÁVRH ČÍSLICOVÝCH FILTROV**

V procese návrhu je potrebné určiť predovšetkým **koeficienty** filtra, prípadne aj **rád** filtra. Existujú rôzne **návrhových algoritmy**, ktoré sa líšia **kritériom a metódami optimalizácie**. Tieto algoritmy sú preberané v iných špecializovaných predmetoch a preto sa ich opisom nebudeme podrobnejšie zaoberať. Sústreďíme sa predovšetkým na praktické zvládnutie niektorých vybraných návrhových algoritmov s cieľom navrhnuť číslicový filter (typu FIR aj IIR) s požadovanými parametrami.

Medzi najčastejšie požiadavky<sup>4</sup> v prípade návrhu číslicového filtra patrí **tolerančná schéma**, ktorá je v prípade DP filtra naznačená na Obr.3, pričom  $\delta_p$  je zvlnenie v priepustnom,  $\delta_s$  je zvlnenie v nepriepustnom pásme a  $\theta_p, \theta_s$  sú normované frekvencie priepustného resp. nepriepustného pásma. Podobne je možné zakresliť tolerančné schémy pre HP, PZ a PP.



Obr.3 Tolerančná schéma DP filtra

Návrh číslicových filtrov FIR a IIR v prostredí Matlab budeme demonštrovať na príkladoch. Podrobnejší opis všetkých príkazov Matlab-u je možné získať príkazom

help prikaz

**Príklad**

Navrhnite DP FIR s pásmom priepustnosti do 3000 Hz, pásmom tlmenia od 5000 Hz, a maximálnym zvlnením v priepustnom a nepriepustnom pásme 0.01 ( $20 \log_{10}(1.01) \approx 0.09 \text{ dB}$ ) resp. 0.001 ( $20 \log_{10}(0.001) = -60 \text{ dB}$ ). Pre návrh využite metódu oknových funkcií a program Matlab, pričom frekvencia vzorkovania je 48000 Hz.

<sup>3</sup> Aj keď uvedená priama forma vyžaduje viac stavebných blokov ako kanonické formy, je často využívaná pri praktickej implementácii napr. pomocou ADSP 218x/9x.

<sup>4</sup> V praxi sa samozrejme číslicové filtre navrhujú aj podľa iných špecifikácií.

## Riešenie

Pre návrh filtrov PP, PZ, DP, HP klasickou metódou oknových funkcií je možné využiť príkaz ***fir1***, v prípade požiadavky na zložitejšie priebehy amplitúdovej frekvenčnej charakteristiky je možné využiť príkaz ***fir2***. Pri návrhu je potrebné zvoliť typ použitého okna. Metóda oknových funkcií však nie je optimálna (z hľadiska zložitosti navrhnutého filtra, t.j. počtu koeficientov<sup>5</sup> FIR filtra, ktoré sú potrebné na splnenie požiadaviek zadania) a v praxi je často výhodnejšie využiť zložitejšie návrhové algoritmy (hlavne pokiaľ ich počíta počítač), ktoré umožňujú navrhnuť FIR filter s menším počtom koeficientov. V ďalšej časti využijeme tzv. **Remezov výmenný algoritmus**, ktorý patrí do kategórie pomerne výkonných návrhových algoritmov. Najskôr určíme rád FIR filtra, ktorý by mal splniť zadané požiadavky:

```
[n,fo,mo,w] = remezord( [3000 5000], [1 0], [0.01 0.001], 48000 );
```

a potom učíme koeficienty navrhnutého filtra:

```
h = remez(n,fo,mo,w);
```

navrhnutý FIR filter má

```
size(h)
```

koeficientov a má symetrické koeficienty podľa vzťahu (1.4) (je to filter s lineárnou fázou). Na spracovanie jednej vzorky navrhnutý FIR filter vyžaduje 62 **MAC**<sup>6</sup> operácií. Koeficienty navrhnutého filtra je možné zobrazit príkazom

```
plot(h)
```

a amplitúdovú a fázovú frekvenčnú charakteristiku príkazom

```
freqz(h)
```

## Príklad

*Navrhnite DP IIR z predchádzajúceho príkladu a porovnajte zložitosť navrhnutého filtra s FIR filtrom navrhnutým v predchádzajúcom príklade.*

## Riešenie

Pre návrh IIR je možné opäť využiť niekoľko rôznych návrhových algoritmov. V ďalšej časti využijeme návrh tzv. **eliptických filtrov** (niekedy tiež nazývaných Cauerove). Tieto filtre využívajú Čebyševovskú aproximáciu v priepustnom aj nepriepustnom pásme a pri návrhu vyžadujú pomerne zložité výpočty (Jakobiho eliptické funkcie, eliptické integrály). S využitím Matlabu je však ich návrh veľmi jednoduchý. Najskôr určíme rád filtra:

```
Rp=20*log10(1.01) % prepocet na decibely
Rs=-20*log10(0.001) % prepocet na decibely
[n,w]=ellipord(3000/(48000/2), 5000/(48000/2), Rp, Rs);
```

a teraz určíme koeficienty filtra

<sup>5</sup> Počet koeficientov je často kritický predovšetkým pri realizácii filtrov v reálnom čase a preto sa snažíme minimalizovať počet koeficientov navrhovaného filtra.

<sup>6</sup> MAC operácia (**M**ultiply **A**nd **a**ccumulate) je operácia typu  $A=A+X*Y$ .

`[b,a]=ellip(n,Rp,Rs,w);`

a jeho amplitúdové a frekvenčné charakteristiky

`freqz(b,a)`

Na spracovanie jednej vzorky navrhnutý IIR filter vyžaduje len 12 MAC operácií. Ak potrebujeme zobrazit' koeficienty na viac desiatinných miest ako 4, použijeme príkaz

```
format long
b
a
```

Pri rozklade na sekcie druhého rádu je potrebné zistiť korene polynómov, ktoré je možné nájsť pomocou príkazu **roots** využit' základné znalosti z algebry o rozklade polynómov.

### Príklad

*Rozložte navrhnutý filter na sekcie druhého rádu.*

Inou možnosťou (podstatne pohodlnejšou) ako navrhnuť číslicové filtra je využitie špecializovaných návrhových programov, ktoré umožňujú navrhnuť číslicové filtre dokonca bez znalosti akýchkoľvek príkazov. Umožňujú tiež rozklad navrhnutých IIR filtrov na sekcie druhého rádu. Na počítačoch v učebni sú dostupné programy [1]<sup>7</sup> a [2]<sup>8</sup>, ktoré umožňujú pohodlným spôsobom navrhnuť prakticky využiteľné FIR a IIR filtre a rozklad IIR filtrov na sekcie druhého rádu. Ich ovládanie je intuitívne a nevyžaduje podrobnejší opis.

## 1.3 SPEKTRÁLNA ANALÝZA A FFT

Diskrétna Fourierova transformácia (DFT) je jednou z najdôležitejších operácií v číslicovom spracovaní signálov. Základným predpokladom využívania DFT je jednak znalosť jej vlastností a jednak efektívnych algoritmov pre jej výpočet. Rýchla Fourierova transformácia (FFT – Fast Fourier Transform) je typickým predstaviteľom rýchleho algoritmu pre výpočet DFT.

Pre **komplexnú**<sup>9</sup> vstupnú postupnosť

$$x(n) = x_r(n) + jx_i(n) \quad n = 0, 1, \dots, N-1 \quad (1.15)$$

je algoritmus FFT definovaný vzťahom

$$X(k) = X_r(k) + jX_i(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi kn}{N}} \quad k = 0, 1, \dots, N-1 \quad (1.16)$$

t.j. výstupom FFT je tiež komplexná postupnosť. Na výpočet FFT v Matlabe sa využíva príkaz **fft**. Na výpočet inverznej FFT (IFFT)

<sup>7</sup> Pri inštalácii uvedeného balíka je potrebné postupovať presne podľa priloženého návodu a pred kopírovaním súborov z adresára UITOOLS je potrebné **skrátit'** mená m-súborov do formátu 8.3.

<sup>8</sup> Pre použitím je potrebné získať licenčný kód. Tento je možné získať bezplatne na príslušnej www stránke.

<sup>9</sup> Samozrejme toto je najvšeobecnejší prípad, postupnosť môže byť aj reálna.

$$x(n) = x_r(n) + jx(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi \frac{kn}{N}} \quad n = 0, 1, \dots, N-1 \quad (1.17)$$

je možné využiť príkaz **ifft**

FFT sa využíva predovšetkým na výpočet spektra signálu, prípadne na rýchly výpočet konvolúcie vo frekvenčnej oblasti. Keďže spektrum signálu je skoro vždy komplexné, je často potrebné využívať na zobrazenie absolútne hodnoty komplexného čísla (funkcia **abs**).

### Príklad

Zobrazte spektrum signálu, ktorý dostaneme ako súčet dvoch sínusoviek s frekvenciami 1500 Hz a 4000 Hz a amplitúdami 3.3, pričom frekvencia vzorkovania je 48000 Hz.

### Riešenie

```
n=0:1023; % diskretný čas
x=3.3*sin(2*pi*n*1500/48000)+3.3*sin(2*pi*n*4000/48000);
y=fft(x); % urcenie spektra
plot(abs(y)) % absolutna hodnota komplexneho vektora
```

### Príklad

Zobrazte spektrum signálu, na výstupe filtrov navrhnutých v predchádzajúcich príkladoch pre vstupný signál ktorý dostaneme ako súčet dvoch sínusoviek s frekvenciami 1500 Hz a 4000 Hz a amplitúdami 1.0, pričom frekvencia vzorkovania je 48000 Hz. Na výpočet výstupu filtra využite funkciu **filter**.

## LITERATÚRA

- [1] Digital Filter Package (DFP). Department of Electrical and Computer Engineering Ryerson Polytechnic University, Toronto, Ontario, CANADA [www.ee.ryerson.ca:8080/~mzeytin/dfp/index.html](http://www.ee.ryerson.ca:8080/~mzeytin/dfp/index.html) (február 2003).
- [2] FilterExpress - Digital Filter Synthesis v5.1. Freeware (potrebná registrácia), [www.systolix.co.uk/soldev.htm](http://www.systolix.co.uk/soldev.htm) (február 2003).
- [3] Filter Design Toolbox. The MathWorks Inc, [www.mathworks.com](http://www.mathworks.com) (február 2002).
- [4] Vích, R.: Transformace Z a některá její použití. SNTL, Praha 1983.
- [5] Magyar, A.: Číslíkové zpracování signálu II, Signálové procesory a jejich použití. Edičné stredisko TU v Košiciach, 1991, ISBN 80-7099-083-X.
- [6] Uhlíř, J. – Sovka, P.: Číslíkové zpracování signálu. Vydavatelství ČVUT, Praha 1995, ISBN 80-01-01303-0.
- [7] Porat, B.: A Course in Digital Signal Processing. John Wiley & Sons, Inc., New York, 1997, ISBN 0-471-14961-6.



- [8] Burrus, C.S. – McClellan, J.H. – Oppenheim, A.V. – Parks, T.W. – Schafer, R.W. – Schuessler, H.W.: Computer Based Exercises for Signal Processing using Matlab. Prentice Hall, Englewood Cliffs, New Jersey, 1994, ISBN 0-13-219825-8.