

Vzorkovač-rekonstrukce průběhu a FFT analýza

Vzorkování signálu

Z měřeného signálu se odebírají v pravidelných intervalech délky T_s (index je z angl. sampling, vzorkování) vzorky, čili hodnoty signálu v okamžicích $n \cdot T_s$, kde n je celé kladné číslo. Tyto hodnoty se následně převedou na čísla, uloží do paměti a jsou určeny k číslicovému zpracování nebo k zobrazení. Původní analogový signál z nich můžeme rekonstruovat, byla-li dodržena tzv. vzorkovací věta: Je-li vzorkován frekvenčně omezený signál s horní mezní frekvencí f_M v ekvidistantních okamžicích s periodou $T_s = 1/f_s$, pak můžeme původní signál získat zpět bez ztráty informace, jeli splněna podmínka $f_s \geq 2 \cdot f_M$. Pokud dojde k porušení podmínky nemůže rekonstrukce proběhnout správně a vznikne tzv. aliasing (zdeformovaný průběh).

Kvantování signálu

Analogově číslicový převodník (analog-to-digital converter) je obvod převádějící hodnotu napětí na svém vstupu na odpovídající číslo. Výstup z analogově-číslcového převodníku je číselný kód (zpravidla n -bitový binární nebo binárně-dekadický). Je ukládán do paměti a následně využit pro zobrazení v grafické podobě (číslcový osciloskop), případně před zobrazením zpracován (spektrální analyzátor, analyzátor výkonu). Pokud je použit přímo k číslicovému zobrazení výsledku měření na zobrazovací přístroje (číslcový voltmetr), je číslo zobrazeno v dekadické číselné soustavě a je mu přiřazena desetinná tečka odpovídající použitému vstupnímu rozsahu voltmetru.

Měření základních parametrů periodických signálů

Digitalizací periodického napětí získáme posloupnosti u_n nebo i_n pro $n \in \langle 0, n-1 \rangle$. Pro posloupnost vzorků měřeného signálu použijeme společné označení x_n . Vzorkovací frekvenci upravíme tak, že sejmeme právě N vzorků za periodu signálu T . Upravíme definice efektivní hodnoty, stejnosměrné složky a střední aritmetické hodnoty tak, že integrály nahradíme součty dílčích integrálů přes sousední vzorkovací intervaly T_s . Pro efektivní hodnotu tak dostaneme výraz

$$X_{\text{ef}} = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} \frac{1}{T_s} \int_{t_n}^{t_{n+1}} x^2(t) dt}$$

pokud nahradíme měřenou veličinu $x(t)$ tzv. schodovitou aproximací, čili předpokládáme konstantní hodnotu signálu rovnou amplitudě vzorku až do okamžiku odebrání následujícího vzorku, platí pro efektivní hodnotu výraz

$$X_{\text{ef}} = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} x_n^2}$$

Pro stejnosměrnou složku platí

$$X_0 = \frac{1}{N} \sum_{n=0}^{N-1} x_n$$

a pro střední aritmetickou hodnotu

$$X_{\text{str}} = \frac{1}{N} \sum_{n=0}^{N-1} |x_n|$$

Přesnější určení uvedených parametrů umožňuje použití lineární interpolace, při které spojíme sousední vzorky úsečkami.

Měření frekvenčního spektra

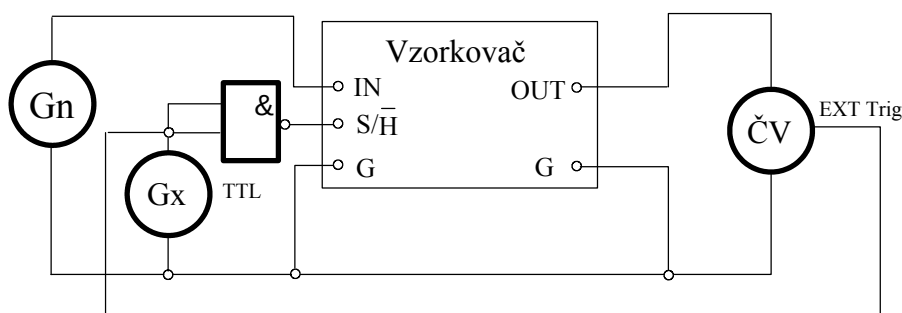
Ovzorkujeme-li periodický měřený signál N vzorků na periodu, můžeme vypočítat koeficienty Fourierovy řady, čili spektrum signálu pomocí N -bodové diskretní Fourierovy transformace (DFT).

$$x(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \cdot e^{-j \frac{2\pi}{N} \cdot n \cdot k}$$

kde k je řád harmonické (tj. $f_k = f_1 \cdot k$), n je celé číslo (pořadí vzorku), N je počet vzorků za periodu signálu $x(t)$. N musí být vyjádřitelné jako celočíselná mocnina dvěma. Pro zvýšení rozlišitelnosti zobrazení FFT lze doplnit naměřené vzorky řadou nulových hodnot (zero padding). Výpočtem se získá N spektrálních čar představujících vzorky Fourierovy transformace vzorkovaného signálu ležícího v pásmu od 0 až po vzorkovací frekvenci.

Výpočet DFT se v praxi provádí pomocí algoritmů nazývaných souhrnně rychlá Fourierova transformace (FFT), které umožňují podstatně zkrátit dobu výpočtu. Tento způsob určení frekvenčního spektra se používá v přístrojích nazývaných FFT spektrální analyzátoři.

Schéma zapojení :



Navrhněte program dle následujícího zadání :

- 1) Na generátoru G_x nastavte vzorkovací frekvenci 10Hz.
- 2) Vyvolejte ovladač generátoru a nastavte zatěžovací impedanci(infinity), typ průběhu(sinus), frekvenci (1Hz) a amplitudu ($1V_{RMS}$)
- 3) Vyvolejte ovladač multimetru a nastavte funkci (DC), integrační dobu (aperture) na 0.02 PLC, přiřadíte výstupy (Readings a Trig_count) a v menu Trigger options nastavte Trigger source na externí, Trigger count (počet spuštění) na 100 a sample count na 1.
- 4) Zadejte vzorkovací frekvenci a určete vzorkovací interval a pomocí objektu Build Waveform sestavte data pro zobrazení navzorkovaného průběhu, který zobrazíte v objektu Waveform (Time).
- 5) Ve formulích vypočtete z navzorkovaných hodnot stejnosměrnou složku, střední hodnotu a efektivní hodnotu vzorkovaného signálu.
- 6) Využijte funkci FFT pro získání frekvenčního spektra vzorkovaného signálu, které zobrazíte v objektu XvsY plot. Data pro Y-ovou osu získáme ve formulích, do kterých vepíšeme $fft(x)/N$ kde N je počet vzorků, $concat(A[0], 2 * A[1:N/2])$ a $mag(A)$. Data pro X-ovou osu získáme v cyklu (od 0 do $N/2$). Hodnoty z cyklu podělíme vzorkovací frekvencí a shromáždíme v Collectoru.

&1. Ověřte určení efektivní hodnoty (nastaveno na generátoru $1V_{rms}$) a z ní vypočtené střední hodnoty (přes činitel tvaru) s hodnotami zjištěnými z navzorkovaných hodnot. pro sinusový trojúhelníkový a obdélníkový signál. Výsledky zpracujte tabelárně

&2. Ověřte správnost určení stejnosměrné složky.

&3. Nyní porušte vzorkovací větu (nastavte frekvenci generátoru G_n na 10Hz a pak na 9Hz). Porovnejte parametry a vyvod'te závěry.

&4. Analyzujte obdélníkový signál frekvence 0.5Hz a napětí $1V_{RMS}$. Vyvod'te závěry.

&5. Pokuste se vymyslet postup určení frekvence a periody z navzorkovaného průběhu.

PROGRAMOVÁ REALIZACE ČÍSLICOVÝCH FILTRŮ

Filtrace signálu je jedním ze základních operací při zpracování signálu. Pomocí filtrace lze ze signálu potřebné složky vybrat nebo odstranit některé nežádoucí složky, jakými jsou např. šumy. S filtry se setkáte zejména v oblasti zpracování zvuku (hi-fi věž, discman, mobilní telefon, televize atd.). Základní myšlenku znázorňuje obr. č. 1.



obr.1 Schéma filtrace

Filtry obecně dělíme na analogové a číslicové. Toto základní rozdělení rozlišuje nejen princip činnosti filtru, ale také fyzickou realizaci. Filtry v oboru zpracování signálů jsou zařízení k úpravě frekvenčního spektra signálu.

Filtrovat lze ve frekvenční nebo časové oblasti. Ve frekvenční oblasti se filtruje násobením frekvenčního přenosu filtru obrazem vstupního signálu, který je třeba v průběhu měření opakovaně počítat. Potom následuje inverzní transformace zpět do časové oblasti. Tento způsob filtrace je vhodný pro dodatečnou úpravu signálů. Filtrace v časové oblasti je naproti tomu teoreticky bez zpoždění.

Analogový filtr pracuje na principu elektronických obvodů, které jsou tvořeny základními elektronickými součástkami, jakými jsou rezistory, kondenzátory a cívky účelově zapojenými tak, aby vznikl požadovaný filtrační efekt. Elektrické signály (proud a napětí) jsou v těchto obvodech v čase spojitě. Jejich vzájemnou závislost popisují diferenciální rovnice.

Číslicový (digitální) filtr je výsledkem numerických operací prováděných digitálním procesorem. Takovým procesorem může být např. každý osobní počítač řady PC nebo specializovaný digitální signálový procesor (DSP). Číslicové (digitální) filtry zpracovávají vzorkované signály. To znamená, že spojitý čas je nahrazen posloupností ekvidistantně rozdělených časových okamžiků $t = nT_s$, $n = 0, 1, 2, \dots$

Číslicový filtr má oproti analogovému následující výhody :

- Číslicový filtr je programovatelný. Jeho práce je určena programem uloženým uvnitř paměti procesoru. To znamená, že lze měnit jeho funkci a parametry beze změny fyzického zapojení jednotlivých součástek (hardwaru).

- Lze jej jednoduše navrhnout, zkoušet, implementovat na každé běžné počítačové pracovní stanici.

- Parametry číslicového filtru nejsou závislé na čase ani na teplotě, protože nejsou tvořeny součástkami jejichž vlastnosti se s časem a teplotou mění (drift u analogových obvodů). Jejich charakteristiky jsou extrémně stabilní.

ZÁKLADNÍ PARAMETRY FILTRŮ

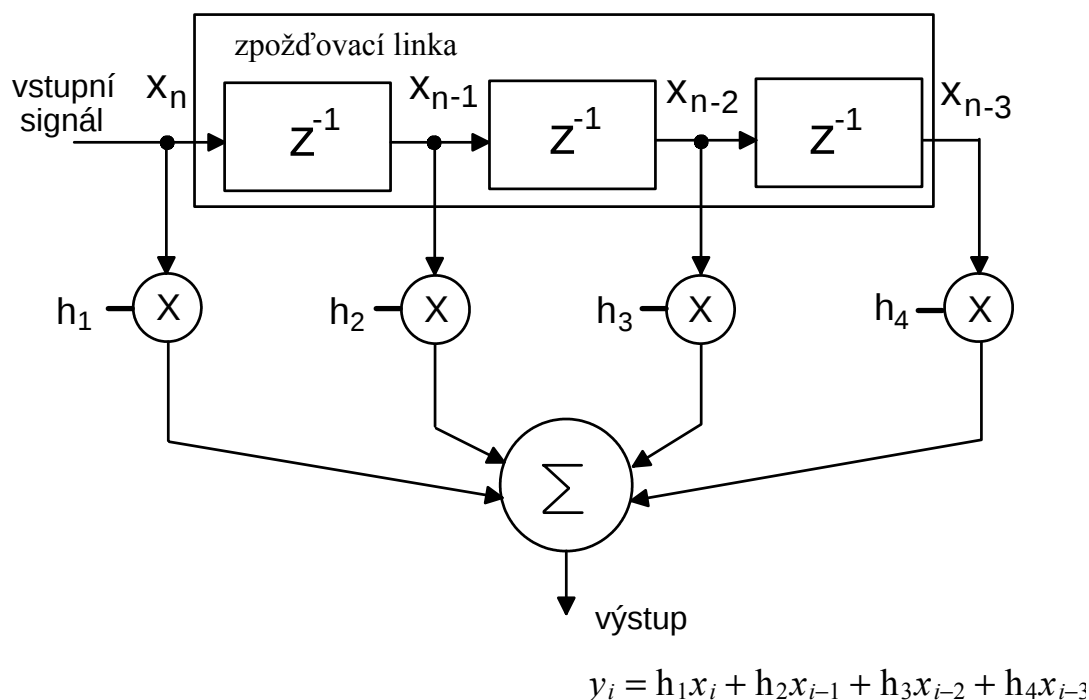
Obecná přenosová funkce filtru má následující tvar

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + \dots + b_m z^{-m}}{1 + a_1 z^{-1} + \dots + a_n z^{-n}}$$

Podle implementace *konvolucí* nebo *rekurzí* dělíme filtry na filtry s konečnou impulsní odezvou (FIR) a filtry s nekonečnou impulsní odezvou (IIR). Filtry realizované konvolucí mají ve jmenovateli pouze 1. Jsou proto obecně mnohem rychlejší. K výpočtu filtru konvolucí je využito pouze členů čitatele přenosové funkce a proto je aktuální hodnota výstupu závislá pouze na velikostech historie vstupů násobených odpovídajícími koeficienty podle vzorce

$$Y_i = \sum_{k=0}^N h_k \cdot x_{i-k}$$

kde h_k jsou koeficienty filtru a x_k jsou vzorky vstupního signálu. Tento filtr je vždy stabilní. Schematicky je výpočet výstupní hodnoty filtru FIR uveden na obr.2 a filtru IIR na obr.3.



obr.2 Schéma filtru s konečnou impulsní odezvou (FIR)

U filtrů IIR je navíc pro výpočet aktuálního výstupu nutné do výpočtu zahrnout předchozí hodnoty výstupu. Při návrhu je nutné dbát i na jeho stabilitu, jelikož kromě nul v čitateli má přenosová funkce také komplexní kořeny, které musí mít zápornou reálnou část.

Nejdůležitější vlastností filtru (vycházející z přenosové funkce) je jeho *frekvenční charakteristika* $H(j\omega)$. Tato charakteristika určuje změnu autospektra $S_{YY}(j\omega)$ signálu na výstupu filtru oproti autospektru $S_{XX}(j\omega)$ signálu na jeho vstupu, což je popsáno rovnicí

$$S_{YY} = H(j\omega) * S_{XX}(j\omega).$$

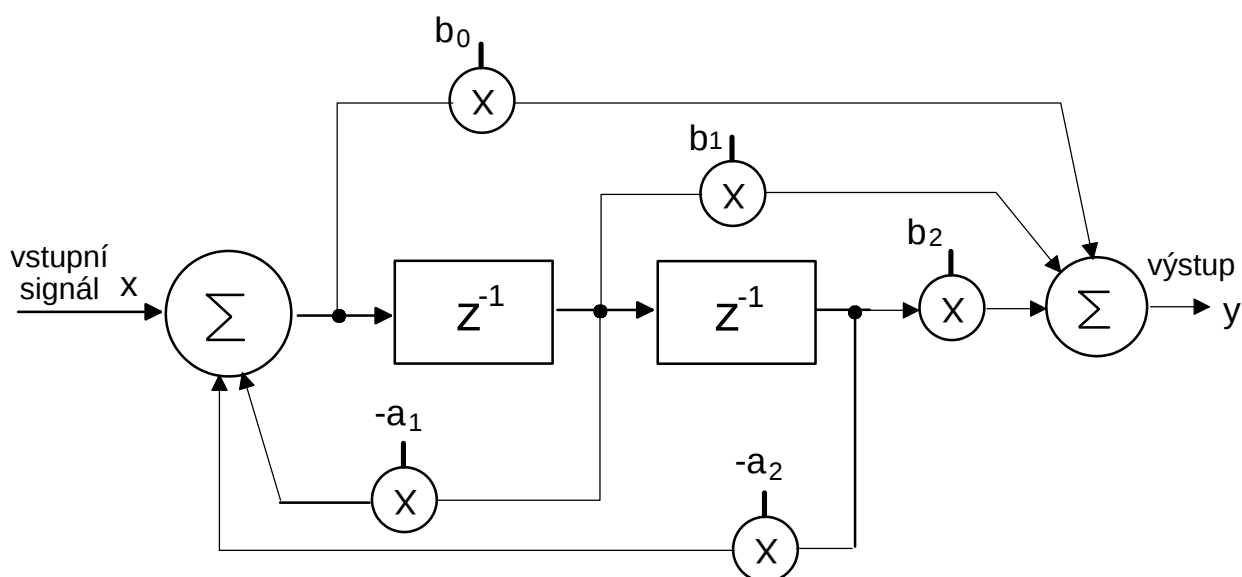
Spektrum signálu je průchodem filtru tvarováno.

Řád filtru FIR je počet předchozích vstupů (uložených v paměti procesoru) použitých pro výpočet aktuálního výstupu. Jeho velikost je o jedna nižší, než je počet koeficientů filtru.

Hlavní nevýhodou filtrů FIR je, že pro splnění požadavků na daný filtr potřebují daleko vyšší řád filtru než filtry IIR. Naproti tomu filtry IIR nemají nikdy lineární fázi (konstantní skupinové zpoždění). U filtru FIR lze vhodnou volbou koeficientů přenosové funkce dosáhnout lineární fázové charakteristiky v celém kmitočtovém pásmu. (Toho se využívá i při návrhu analogových filtrů).

FIR - finite impulse response

IIR - infinite impulse response



$$y_i = b_0x_i + b_1x_{i-1} + b_2x_{i-2} - a_1y_{i-1} - a_2y_{i-2}$$

obr. 3: Schéma filtru s nekonečnou impulsní odezvou (IIR)

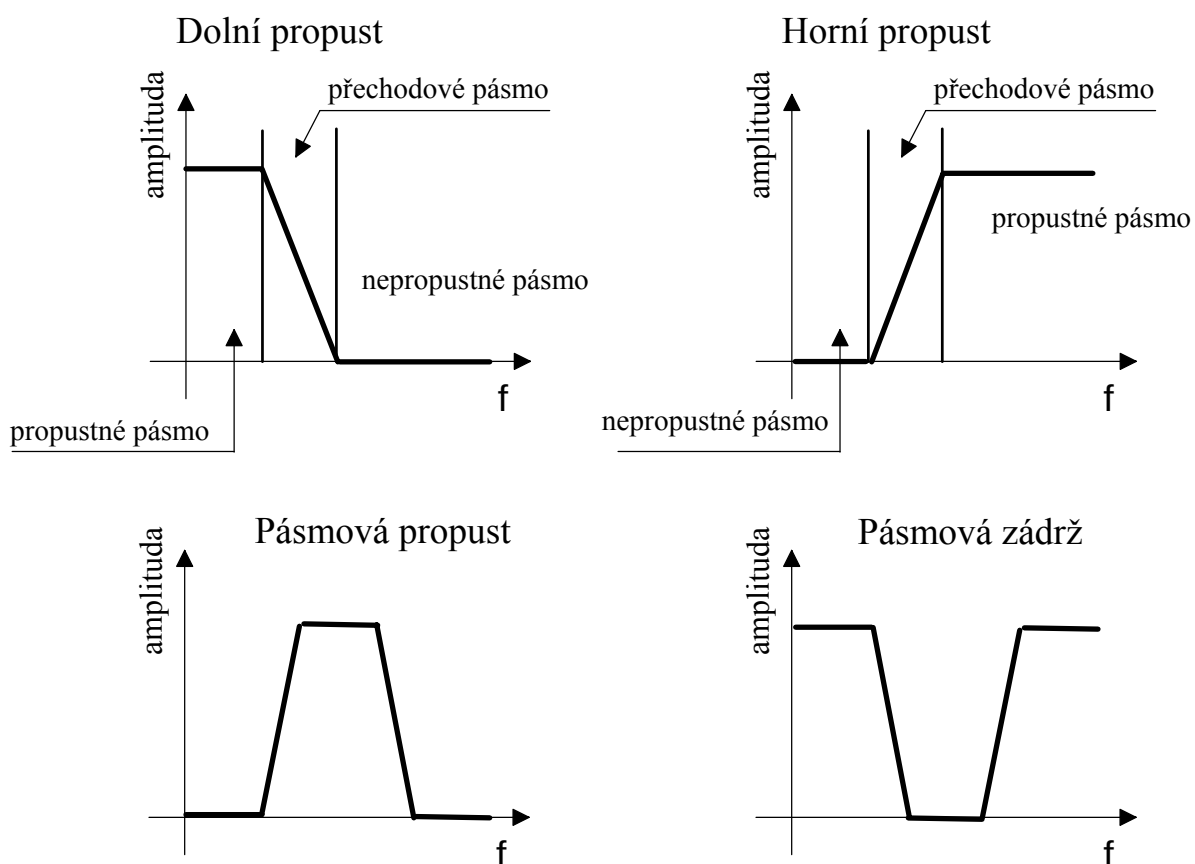
Podle způsobu ovlivnění spektra signálu se filtry rozdělují na tyto základní typy:

- filtr typu dolní propust (low pass filter)
- filtr typu horní propust (high pass filter)
- filtr typu pásmová propust (band pass filter)
- filtr typu pásmová zadrž (band stop filter)

Průběh frekvenčních charakteristik podle pásma propustnosti je znázorněn na obr.4.

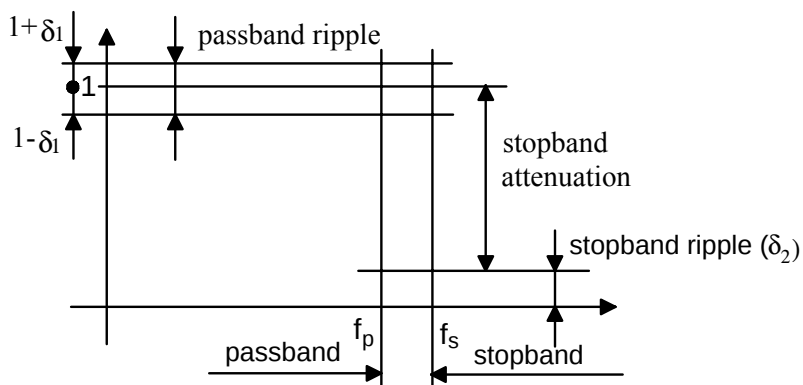
Podle zvolené metody návrhu filtru pro nás budou důležité různé jeho parametry. Důležitými parametry filtru, které lze vyčíst z obr.5, jsou zvlnění v propustném pásmu (passband ripple), útlum (stopband attenuation) a zvlnění v nepropustném pásmu (stopband ripple). Parametry, které jsou společné pro návrh všemi metodami jsou frekvence začátku a konce přechodových pásem. Tyto frekvence jsou dále normalizovány, což se provádí podílem dané frekvence a 1/2 vzorkovací frekvence. Získáme tak hodnoty frekvencí v rozsahu 0 až 1 v souladu se Shannon-Kotelnikovým vzorkovacím teorémem. Podle detailního průběhu frekvenčních charakteristik

a jejich matematického vyjádření existují hlavní druhy filtrů : Butterworthův, Besselův, Čebyševův prvního typu, Čebyševův druhého typu a Cauerův (eliptický) filtr.



obr.4 Ideální frekvenční charakteristiky filtrů

- Butterworthův filtr má maximálně ploché propustné i nepropustné pásmo
- Besselův filtr má menší selektivitu (široké přechodové pásmo) a konstantní skupinové zpoždění (lineární fázovou charakt.)
- Čebyševův filtr I. typu má zvlnění v propustném pásmu
- Čebyševův filtr II. typu má zvlnění v nepropustném pásmu. Čebyševovy filtry mají ostřejší přechod než Butterworthovy filtry.
- Eliptický filtr má zvlnění v propustném i nepropustném pásmu. Pro daný řád má nejostřejší přechod (nejušší přechodové pásmo) ze všech filtrů.



obr. 5 Důležité parametry filtru

Určení řádu IIR filtrů a mezních kmitočtů

Vstupní parametry :

- rp.....zvlnění v propustném pásmu (passband ripple) - (dB)
- rs.....odstup nepropustného pásma (stopband ripple) - (dB)
- wp....normovaný hraniční kmitočet specifikující propustné pásmo (v případě propustí vektor dvou kmitočtů) $wp = fp/(fv/2)$
- ws.....normovaný hraniční kmitočet specifikující nepropustné pásmo (v případě propustí a zadrží vektor dvou kmitočtů) $ws = fs/(fv/2)$

Výstupní parametry :

- N.....řád filtru
- wn....normovaný mezní (hraniční, kritický, zlomový) kmitočet - pokles o 3dB, v případě pásmových propustí vektor dvou normovaných mezních kmitočtů specifikující dané pásmo

$[N, wn] = \text{buttord}(wp, ws, rp, rs)$
 $[N, wn] = \text{cheb1ord}(wp, ws, rp, rs)$
 $[N, wn] = \text{cheb2ord}(wp, ws, rp, rs)$
 $[N, wn] = \text{ellipord}(wp, ws, rp, rs)$

Návrhy filtrů požadovaného řádu a mezních kmitočtů

Vstupní parametry :

- N.....řád filtru
- rp.....zvlnění v propustném pásmu (passband ripple)
- rs.....odstup nepropustného pásma (stopband ripple)
- wn....normovaný mezní (hraniční, kritický, zlomový) kmitočet - pokles o 3dB, v případě propustí a zadrží vektor dvou normovaných mezních kmitočtů specifikující dané pásmo.

Výstupní parametry :

- b.....vektor koeficientů vstupních vzorků standardní diferenční rovnice
- a.....vektor koeficientů výstupních vzorků standardní diferenční rovnice

Návrh dolních propustí:

$[b, a] = \text{butter}(N, wn)$
 $[b, a] = \text{cheby1}(N, rp, wn)$
 $[b, a] = \text{cheby2}(N, rs, wn)$
 $[b, a] = \text{ellip}(N, rp, rs, wn)$

Návrh horních propustí :

$[b, a] = \text{butter}(N, wn, 'high')$
 $[b, a] = \text{cheby1}(N, rp, wn, 'high')$
 $[b, a] = \text{cheby2}(N, rs, wn, 'high')$
 $[b, a] = \text{ellip}(N, rp, rs, wn, 'high')$

Návrh pásmových propustí :

[b,a] = butter (N, [wn1 wn2])
[b,a] = cheby1 (N, rp, [wn1 wn2])
[b,a] = cheby2 (N, rs, [wn1 wn2])
[b,a] = ellip (N, rp, rs, [wn1 wn2])

[wn1 wn2]vektor dvou normovaných mezních kmitočtů specifikujících propustné pásmo

Návrh pásmových zádrží :

[b,a] = butter (N, [wn1 wn2], 'stop')
[b,a] = cheby1 (N, rp, [wn1 wn2], 'stop')
[b,a] = cheby2 (N, rs, [wn1 wn2], 'stop')
[b,a] = ellip (N, rp, rs, [wn1 wn2], 'stop')

[wn1 wn2].....vektor dvou normovaných mezních kmitočtů specifikujících nepropustné pásmo

Návrh filtrů libovolného tvaru (včetně vícepásmových propustí)

Aproximace funkce lineární po úsecích racionální lomenou funkcí metodou nejmenších čtverců

[b,a] = yulewalk (N, f, m)

Vstupní parametry :

N.....řád filtru

f.....vektor normovaného kmitočtu (musí začínat 0 a končit 1, tj. Nyquistovým kmitočtem)

m.....vektor amplitud (musí odpovídat jednotlivým kmitočtům, délka vektorů f a m musí být stejná!)

Prvky obou vektorů (f,m) představují souřadnice specifikující amplitudovou charakteristiku.

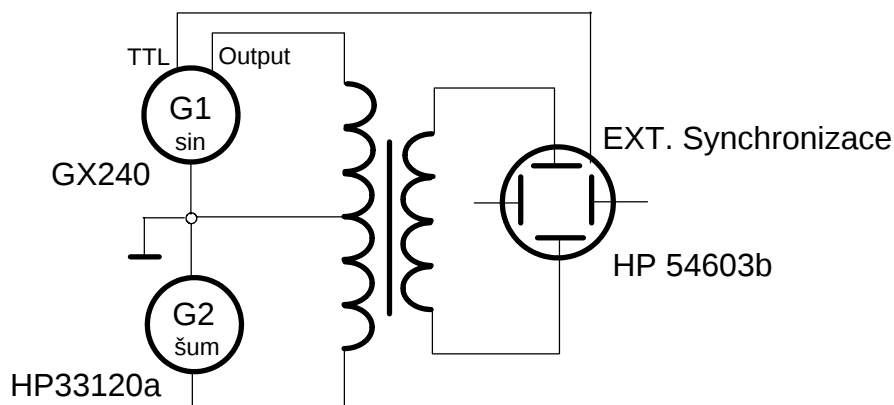
Např.:

m = [0, 0, 1, 1, 0, 0, 1, 1, 0, 0]
f = [0, 0.1, 0.2, 0.4, 0.4, 0.5, 0.6, 0.8, 0.8, 1]
[b,a] = yulewalk (20, f, m).

ZADÁNÍ :

V grafickém programovacím jazyce HP-VEE vytvořte program realizující číslicový filtr IIR, kterým odstraníte ze sinusového zdroje signálu frekvence 800Hz (generátor GX240) šum, který vytváří generátor HP 33120a. Šum je přičten k užitečnému signálu v transformátoru a simuluje např. přenosovou trasu na které může k podobnému “znehodnocování” signálu docházet. Vzniklý “zašuměný” signál zobrazte na osciloskopu a “přetáhněte” do programu HP VEE ve kterém realizujete příslušný filtr. Koeficienty a,b filtru navrhnete využitím funkce YULEWALK. Výsledkem práce budou časové průběhy na vstupu a výstupu filtru a jejich frekvenční spektra.

Schéma zapojení :



POSTUP:

1/ Na GX 240 nastavte frekvenci 800Hz, sinus a napětí tak, aby se na osciloskopu zobrazila sinusovka s napětím $U_{ss} = 9V$. Zapněte generátor HP 33120a, nastavte funkci NOISE (šum) a napětí tak, aby došlo k výraznému “zašumění” původního signálu (např. $8V_{ss}$)

2/ V programu HP-VEE vyvolejte ovladač osciloskopu. Nepoužívejte tlačítko autoscale. Nastavte zdroj spuštění (TRIG SOURCE) na externí. Vypněte kanál 2. Nastavte vhodnou citlivost kanálu 1. Rychlost časové základny (TIMEBASE) zvolte tak, aby se zobrazily minimálně 4 periody. Stiskněte tlačítko digitize a ověřte zda došlo k požadovanému zobrazení signálu. Na ovladači osciloskopu ještě vytvořte výstup (WF CH1) pro zobrazení časového průběhu signálu na vstupu filtru. Zároveň tím získáme matici s navzorkovanými hodnotami vstupního signálu filtru.

3/ Spočítejte vzorkovací frekvenci osciloskopu a určete vektory normovaného kmitočtu a vektor amplitud pro filtr šestého řádu. Pomocí funkce YULEWALK navrhnete koeficienty a , b číslicového filtru. Pro určení vzorkovací frekvence si zjistěte délku časového průběhu (TIME SPAIN) a počet vzorků vstupního signálu filtru (na výstupu WF CH1).

4/ Podle obr.3 naprogramujte IIR filtr 6 řádu.

Některé objekty které je vhodné využít :

UNBUILD WAVEFORM - vstupem jsou data typu waveform (časový průběh) obsahuje pole vzorků typu REAL časového průběhu a délku trvání tohoto časového průběhu (TIMESPAIN). Opakem je objekt BUILD WAVEFORM. Využijte ho pro opětovné sestavení časového průběhu poté co navzorkované hodnoty upravíte filtrem a shromáždíte ve SLIDING COLLECTORU do pole.

FOR COUNT - Cyklus jehož délku nastavíte v závislosti na počtu navzorkovaných hodnot. Tu zjistíte funkcí TOT SIZE aplikovanou na pole vzorků. Ústředním objektem filtru je objekt SHIFT REGISTER (posuvný registr) ve kterém se data v každém taktu posunou o krok vpřed. To znamená, že se hodnota x na vstupu registru objevuje postupně na pozici CURRENT, 1PREV, 2PREV atd. Pro zápis matematických operací využijeme objekty FORMULA. Pro získání frekvenčního spektra signálu před a za filtrem použijeme objekt MAGNITUDE SPECTRUM.