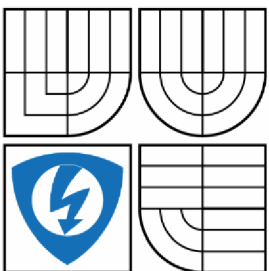


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

## ČÍSLICOVÁ FILTRACE – VÝUKOVÁ SIMULACE

DIGITAL FILTERS – LEARNING SIMULATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

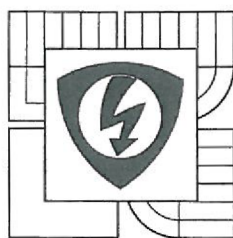
LUKÁŠ HAMPL

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. SOŇA ŠEDIVÁ, Ph.D.

BRNO 2016



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav automatizace a měřicí techniky

# Bakalářská práce

bakalářský studijní obor  
Automatizační a měřicí technika

**Student:** Lukáš Hampel

**Ročník:** 3

**ID:** 164275

**Akademický rok:** 2015/16

**NÁZEV TÉMATU:**

## Číslicová filtrace - výukové simulace

### POKYNY PRO VYPRACOVÁNÍ:

Cílem bakalářské práce je vytvoření programu umožňujícího návrh číslicových FIR a IIR filtrů v prostředí LabVIEW. Bude se jednat o výukovou simulaci sloužící pro předmět Sběr, analýza a zpracování dat.

1. Proveďte literární rešerši v oblasti číslicové filtrace a návrhů číslicových FIR a IIR filtrů.
2. Navrhněte metodiku řešení programu pro návrh číslicových filtrů v prostředí LabVIEW, který by demonstroval vlastnosti navrženého filtru.
3. Naprogramujte navržené řešení v prostředí LabVIEW. Program by měl umožňovat uživateli vybrat typ filtru, princip návrhu filtru, měnit základní parametry filtru, zobrazit charakteristiky navrženého filtru a dále zobrazit původní a filtrovaný signál.
4. Ověřte funkčnost vytvořeného programu.
5. Demonstrujte a zhodnoťte výhody a nevýhody jednotlivých navržených filtrů.

### DOPORUČENÁ LITERATURA:

[1] Merins, A. – Signal analysis – Wavelets, Filter Banks, Time-Frequency Transforms and Applications. John Wiley and Sons, Chichester, 1999, ISBN 0-471-98626-7

**Termín zadání:** 8. 2. 2016

**Termín odevzdání:** 23.5.2016

**Vedoucí práce:** Ing. Soňa Šedivá, Ph.D.

**Konzultanti bakalářské práce:**

doc. Ing. Václav Jirsík, CSc.

předseda oborové rady

### UPOZORNĚNÍ:

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



# Abstrakt

Tato práce je zaměřená na teoretický rozbor a aplikaci metod návrhu dvou základních typů číslicových filtrů. Konkrétními číslicovými filtry jsou v této bakalářské práci filtry FIR (konečná impulzní odezva) a IIR (nekonečná impulzní odezva). Hlavním cílem této bakalářské práce je kromě teoretického rozboru o číslicové filtraci vytvořit program v prostředí LabVIEW, pomocí kterého může uživatel navrhnout číslicové filtry obou typů. Program by měl sloužit studentům ve výuce pro snadnější pochopení problematiky číslicové filtrace. Důraz je zde kladen na možnost podrobného návrhu obou filtrů, kdy si uživatel může vyzkoušet, jak se mění jednotlivé vlastnosti filtru při sebemenších změnách nastavení. Za pomoci tohoto programu je, kromě filtrování vlastního nastaveného vstupního signálu, důležité hlavně vzájemné porovnávání navržených číslicových filtrů, podle několika kritérií.

## Klíčová slova

Číslicová filtrace, FIR, IIR, LabVIEW, vzájemné porovnávání filtrů, filtrace signálu, výuková simulace

## Abstract

This thesis focuses on the theoretical analysis and application of methods of the two basic types of the digital filters designs. The specific digital filters in this bachelor's thesis are filters FIR (final impulsive response) and IIR (infinite impulsive response). The main aim in this bachelor's thesis is, besides the theoretical analysis about the digital filtration, to create a program in the LabVIEW. A user can design digital filters of both types with the help of this program. This program should serve as a learning tool for students to easier understand the problematics of the digital filtration. The emphasis is on the possibility of the design of both filters when user can try how each property is changed with the slightest change of settings. Apart from the filtration of the own input signal, the reciprocal comparison of the both digital filters according to a few criteria is very important.

## Keywords

Digital filtration, FIR, IIR, LabVIEW, reciprocal comparison of the filters, filtration of a signal, learning simulation

## **Bibliografická citace**

HAMPL, L. Číslicová filtrace - výukové simulace. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2016. 65 s. Vedoucí bakalářské práce Ing. Soňa Šedivá, Ph.D.

## **Prohlášení**

„Prohlašuji, že svou bakalářskou práci na téma Číslicová filtrace - výukové simulace jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **22. května 2016**

.....

podpis autora

## **Poděkování**

Děkuji vedoucí bakalářské práce Ing. Soně Šedivé, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce. Dále děkuji všem svým přátelům, rodině a přítelkyni za velkou podporu, která mi pomohla dokončit práci

V Brně dne: **22. května 2016**

.....

podpis autora

# Obsah

Úvod.....	11
1 Signály a filtry .....	12
1.1 Filtrace .....	12
1.2 Druhy filtrů .....	13
2 Číslicové filtry .....	15
2.1 Matematický popis číslicového filtru.....	16
2.2 Stabilita číslicového filtru .....	17
2.3 Filtr s konečnou impulzní odezvou .....	17
2.4 Metody návrhu FIR filtru .....	21
2.4.1 Metoda váhové posloupnosti.....	21
2.4.2 Optimalizační metoda Equi-ripple .....	22
2.5 Filtr s nekonečnou impulzní odezvou .....	23
2.6 Metody návrhu IIR filtru.....	24
2.6.1 Optimalizační metody návrhu IIR filtrů.....	25
2.6.2 Návrh IIR podle podobnosti s analogovým systémem.....	25
3 Návrh programu .....	27
3.1 Kritéria pro posouzení navrženého filtru .....	27
3.1.1 Početní náročnost .....	28
3.1.2 Otočení fáze signálu .....	28
3.1.3 Řád filtru a jednotlivý počet pólů a nul.....	29
3.1.4 Odchylka filtrovaného signálu .....	29
3.1.5 Poměr signálu vůči šumu – SNR.....	29
4 Softwarové Prostředí .....	30
4.1 LabVIEW .....	30
4.2 Front panel .....	30
4.3 Block diagram .....	31
5 Realizace programu.....	32
5.1 Uživatelské rozhraní.....	32
5.2 Nastavení vstupního signálu .....	33
5.2.1 Nastavení vstupního signálu pomocí základních typů .....	33
5.2.2 Nastavení vstupního signálu pomocí více harmonických složek.....	35
5.2.3 Nastavení vstupního signálu zadáním textu .....	36

5.3	Nastavení a vzájemné porovnávání navržených filtrů .....	37
5.3.1	Zobrazení vstupních signálů .....	37
5.3.2	Nastavení FIR filtru.....	37
5.3.3	Nastavení IIR filtru .....	39
5.3.4	Vstupní a filtrované signály .....	41
5.4	Vykreslování pólů a nul obou typů filtru .....	42
5.5	Zatížení počítače při výpočtu filtrů a filtraci.....	43
5.6	Ovládání programu.....	43
6	Ověření funkčnosti programu.....	45
6.1	První příklad testování programu.....	45
6.1.1	Nastavení filtrů pro filtraci prvního signálu .....	46
6.1.2	Výsledky filtrace prvního příkladu .....	47
6.1.3	Zhodnocení výsledků prvního příkladu.....	47
6.2	Druhý příklad testování programu .....	48
6.2.1	Nastavení filtrů pro filtraci druhého signálu .....	48
6.2.2	Výsledky druhého příkladu .....	49
6.2.3	Zhodnocení výsledků druhého příkladu .....	50
	Závěr .....	51



# Obrázky

obr 1. Frekvenční charakteristika dolní a horní propusti .....	12
obr 2. Frekvenční charakteristiky pásmové propusti a zádrže .....	13
obr 3. Impulzní odezva FIR filtru .....	15
obr 4. Impulzní odezva IIR filtru .....	15
obr 5. Základní realizační bloky pro sestavení číslicových filtrů .....	16
obr 6. Určení stabilního číslicového filtru. Převzato z [10] .....	17
obr 7. Blokové schéma FIR filtru. Převzato z [9] .....	18
obr 8. FIR filtr s lineární fází, typ 1. Převzato z [1] .....	20
obr 9. FIR filtr s lineární fází, typ 2. Převzato z [1] .....	20
obr 10. FIR filtr s lineární fází, typ 3. Převzato z [1] .....	20
obr 11. FIR filtr s lineární fází, typ 4. Převzato z [1] .....	20
obr 12. Blokové schéma IIR filtru.....	24
obr 13. Frekvenční charakteristika základních aproximací. Převzato z [7] .....	26
obr 14. Rozdělený program na části.....	33
obr 15. Nastavení vstupního signálu pomocí základních typů.....	34
obr 16. Generátor základních signálů.....	34
obr 17. Nastavení vstupního signálu poskládáním z více složek .....	35
obr 18. Generátor signálu pomocí skládání více signálů .....	36
obr 19. Nastavení vstupního signálu zadáním textu.....	36
obr 20. Generátor signálu pomocí zadávání textem .....	37
obr 21. Zobrazení vstupního a skutečného signálu .....	37
obr 22. Návrh filtru FIR .....	38
obr 23. Filtr FIR v programu LabVIEW .....	39
obr 24. Kontrola nastavení filtru FIR.....	39
obr 25. Filtr IIR v programu LabVIEW .....	39
obr 26. Návrh filtru IIR .....	40
obr 27. Kontrola nastavení filtru IIR.....	40
obr 28. Vstupní a filtrované signály .....	41
obr 29. Funkce pro výpočet SNR.....	42
obr 30. Vykreslování pólu a nul obou filtrů .....	42
obr 31. Funkce pro získání pólů a nul .....	42
obr 32. Zobrazení náročnosti programu a doby filtrace .....	43
obr 33. Část pro ovládání programu.....	44

obr 34. Ukázka prvního příkladu vstupního signálu .....	45
obr 35. Porovnání vstupních a filtrovaných signálu prvního příkladu .....	47
obr 36. Ukázka druhého příkladu vstupního signálu.....	48
obr 37. Porovnání vstupního a filtrovaných signálu druhého příkladu .....	49

## Tabulky

Tabulka 1. Výhody a nevýhody analogových filtrů.....	14
Tabulka 2. Výhody a nevýhody číslicových filtrů .....	14
Tabulka 3. Shrnutí jednotlivých typů FIR filtrů.....	19
Tabulka 4. Porovnání výsledků u prvního příkladu .....	47
Tabulka 5. Porovnání výsledků u druhého příkladu .....	50

# ÚVOD

Tato bakalářská práce se zabývá problematikou číslicové filtrace a číslicových filtrů, jedná se především o filtr IIR (nekonečná impulzní odezva) a filtr FIR (konečná impulzní odezva). Podle zadání má být vytvořen program, který umožňuje uživateli navrhnout číslicový filtr, na kterém je možné demonstrovat výhody a nevýhody návrhu. K naprogramování má být použit program LabVIEW od National Instrument. Tento program má stále větší oblibu u významných firem po celém světě pro jeho jednoduchost a průhlednost kódu.

Číslicová filtrace je metoda úpravy signálu, která má velký potenciál v různých odvětvích. Dnešní doba si žádá v mnoha oborech měření nejrůznějších veličin a jejich následnou úpravu, proto má číslicová filtrace velice významnou roli. Jednat se může například o odstraňování šumu, nebo změny jednotlivých složek signálu. Toho využívají jednotlivé filtrační metody, které mají například uplatnění ve strojním, elektrotechnickém, biomedicínském nebo audio oboru. Jedná o čistě softwarovou metodu, která se dá neustále zdokonalovat.

Cílem práce tedy je navrhnout program, který využívá filtraci signálu pomocí obou typů filtrů. Pomocí těchto filtrů odstranit vybrané složky z nastaveného vstupního signálu. Pro snadnější určení vhodnosti číslicového filtru je vykreslováno několik frekvenčních charakteristik spolu s vykreslením pólů a nul obou navržených filtrů. Uživatel má také možnost zobrazení výstupního filtrovaného signálu po jeho filtraci nastavenými FIR a IIR filtry.

Pro jednoznačnou demonstraci rozdílu vlastností jednotlivých filtrů jsou zde použita vyhodnocovací kritéria, která posuzují každý z navržených filtrů stejnou měrou, čímž je patrný rozdíl mezi každým z navržených typů filtru. Pomocí těchto kritérií může uživatel snadno porovnat oba typy, ale také blíže pochopit problematiku číslicové filtrace a způsob vhodného návrhu filtru na daný vstupní signál.

# 1 SIGNÁLY A FILTRY

Signály jsou obecně děleny na spojité a nespojité. Spojitým signálem je označovaná spojitá funkce, nebo po částech spojitá funkce  $f(x)$ , která je spojitá hodnotou  $x$ , tou je nejčastěji čas [1]. Hodnota  $x$  však může být i jiná veličina. Spojitými signály se tato bakalářská práce nebude nadále zabývat.

Oproti tomu diskrétní signál se na rozdíl od analogového signálu nemění spojitě v čase, je tvořen posloupností vzorků. Tato posloupnost signálu bývá nejčastěji tvořena opakujícím se měření. Funkce diskrétního signálu má tvar  $f_i = f(i)$ , kde hodnota  $i$  jsou celočíselné izolované okamžiky, neboli dílčí mezivýsledky, kterých docílíme kvantováním. [1]

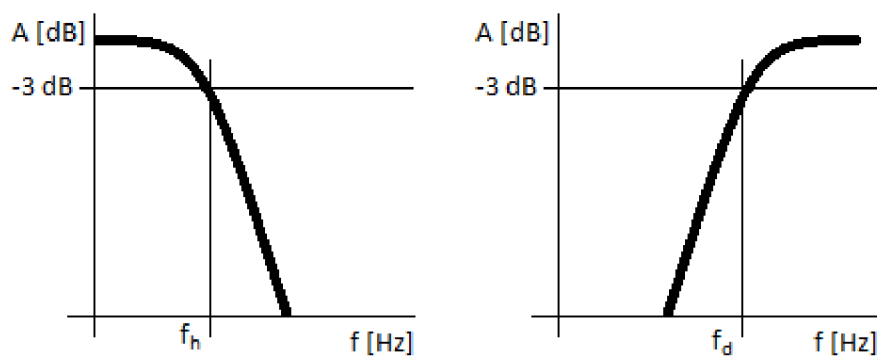
## 1.1 Filtrace

Filtrace je určitý druh zpracování signálu, při kterém jsou některé složky signálu potlačovány, aby vznikl chtěný signál. Jednoduše řečeno filtrace slouží k ovlivňování jednotlivých harmonických složek signálu.

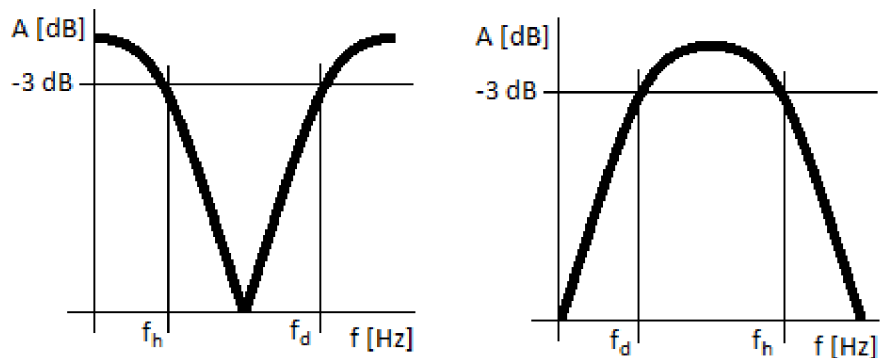
Úkolem filtrace bývá nejčastěji potlačení nechtěných kmitočtových spekter signálu, to znamená odfiltrování frekvencí jako například šumu, nebo výběr pouze určitých frekvencí signálu.

Obvykle jsou použity následující typy filtrů:

- Dolní propust (DP) – odstraňuje vysoké frekvence ze signálu - obr 1
- Horní propust (HP) – odstraňuje nízké frekvence ze signálu - obr 1
- Pásmová propust (PP) – propustí frekvence pouze z určitého frekvenčního pásma - obr 2
- Pásmová zádrž (PZ) – zadržuje frekvence určitého frekvenčního pásma - obr 2



obr 1. Frekvenční charakteristika dolní a horní propusti



obr 2. Frekvenční charakteristiky pásmové propusti a zádrže

Předchozí obr 1 a obr 2 zobrazují frekvence  $f_d$  a  $f_h$ , kde je označen tzv. bod zlomu. Tento zlom je zde zobrazen jako pokles signálu o 3 dB neboli dělením amplitudy přibližně číslem 1,41. Výsledný výstupní signál odpovídá poklesu přibližně na 70 % vstupní hodnoty amplitudy signálu.

Pro větší útlum signálu se dále používají poklesy o 6 dB tj. zmenšení na 50 % vstupní hodnoty amplitudy signálu. Další používaný útlum je poklesy o 20 dB, ten zaručuje snížení na přibližně desetinu původní amplitudy signálu.

## 1.2 Druhy filtrů

V minulosti se nejčastěji využívaly tzv. analogové filtry, ty byly zastoupeny pasivními prvky R, L, C a jejich vzájemnými kombinacemi [1]. S příchodem počítačů a výpočetní techniky se však filtry začaly vyvíjet programově. Tyto softwarové filtry se označují jako číslicové filtry.

Na rozdíl od analogových filtrů, které mají popis pomocí Laplaceovy transformace, se číslicové filtry popisují pomocí Z transformace, tato transformace má tvar podle rovnice (1).

Rovnice Z - transformace

$$X(z) = \sum_{n=0}^{\infty} x(n) \cdot z^{-n} \quad (1)$$

Jak analogové, tak číslicové filtry mají specifické vlastnosti. Výhody a nevýhody analogových a číslicových filtrů shrnují následující tabulky.

Tabulka 1. Výhody a nevýhody analogových filtrů

<b>Analogové filtry</b>	
<b>Výhody</b>	<b>Nevýhody</b>
<p>Velký frekvenční rozsah</p> <p>Levnější</p> <p>Realizovatelnost z jednoduchých obvodů</p>	<p>Menší rozsah</p> <p>Citlivé na šum a nepřesné</p> <p>Nelinearita jednotlivých součástek</p> <p>Omezený dynamický rozsah</p> <p>Špatná výrobní reprodukovatelnost</p>

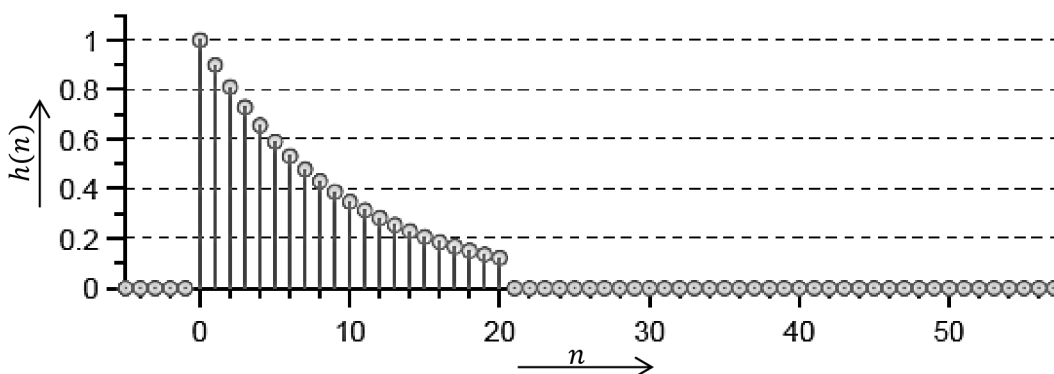
Tabulka 2. Výhody a nevýhody číslicových filtrů

<b>Číslicové filtry</b>	
<b>Výhody</b>	<b>Nevýhody</b>
<p>Vysoká přesnost</p> <p>Rychlost</p> <p>Nemá drift</p> <p>Flexibilita parametrů (softwarová změna)</p> <p>Neomezený dynamický rozsah</p> <p>Jednoduchá aplikace na diskrétní signály</p>	<p>Vyžadují A/D převodníky</p> <p>Vzorkovací frekvence určuje frek. rozsah</p>

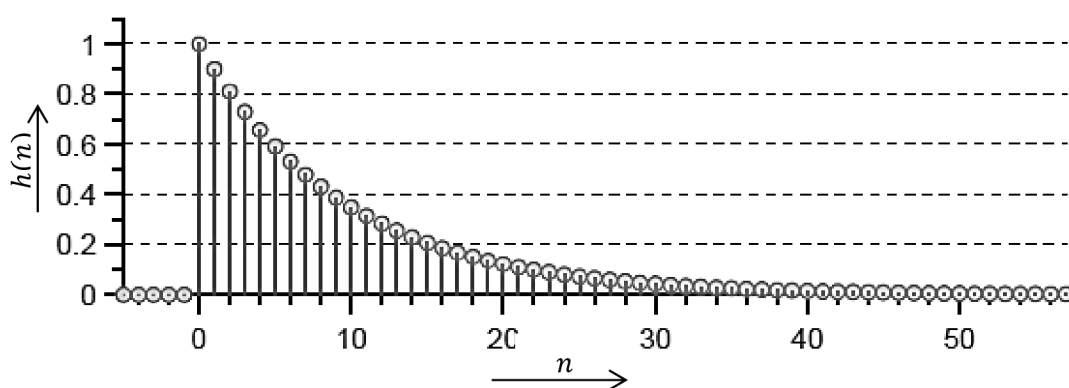
## 2 ČÍSLICOVÉ FILTRY

Tato práce se bude nadále zabývat pouze číslicovými filtry. Číslicové filtry dělíme podle několika kritérií. Předchozí obr 1 a obr 2 zobrazovaly pouze základní typ dělení filtrů, to bylo rozdělení podle jednotlivých propustí signálu DP, HP, PP, PZ. Další kritéria, podle kterých jsou děleny číslicové filtry [2]:

- Podle délky impulzní odezvy
  - Filtry s konečnou impulzní odezvou FIR viz obr 3.
  - Filtry s nekonečnou impulzní odezvou IIR viz obr 4.
- Podle struktury blokového schématu
  - Nerekurzivní filtry NRDF (nemají zpětnou vazbu)
  - Rekurzivní filtry (mají zpětnou vazbu), většinou IIR filtry



obr 3. Impulzní odezva FIR filtru

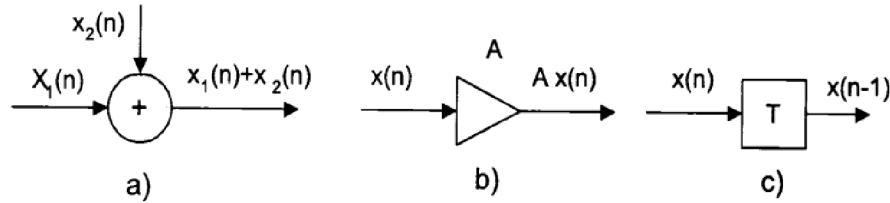


obr 4. Impulzní odezva IIR filtru

Filtry typu FIR a IIR jsou základními číslicovými filtry, které se používají převážně v diskretních systémech. Jak FIR, tak IIR filtr se dá zrealizovat pomocí tří základních bloků, tyto bloky jsou zobrazeny na obr 5. [5]

Jedná se o následující bloky:

- Sčítací člen - sečte dva signálů, v obr 5 část a)
- Násobička - vynásobí daný signál konstantou, v obr 5 část b)
- Zpoždovací člen - Zpoždění o jeden krok, v obr 5 část c)



obr 5. Základní realizační bloky pro sestavení číslicových filtrů

## 2.1 Matematický popis číslicového filtru

Číslicové filtry jsou popsány pomocí diferenciálních rovnic. U číslicových filtrů je výhodnější zapisovat jednotlivé diference pomocí posloupností [3]. Tvar diferenční rovnice:

$$y_n = \sum_{i=0}^r L_i \cdot x_{n-i} - \sum_{i=1}^m K_i \cdot y_{n-i} \quad (2)$$

Suma s hodnotami  $K$  označuje rekurzivní část. Jsou-li všechny členy  $K_i = 0$ , pak systém nemá rekurzivní část. Hodnoty  $L_i$  jsou tedy násobičky nerekurzivní části. Písmena  $r$  a  $n$  označují celkový počet hodnot nerekurzivní a rekurzivní části. Po transformaci se dá diferenciální rovnice převést na tzv. obrazový přenos, který je charakterizován následující rovnicí.

Obrazový přenos číslicového filtru

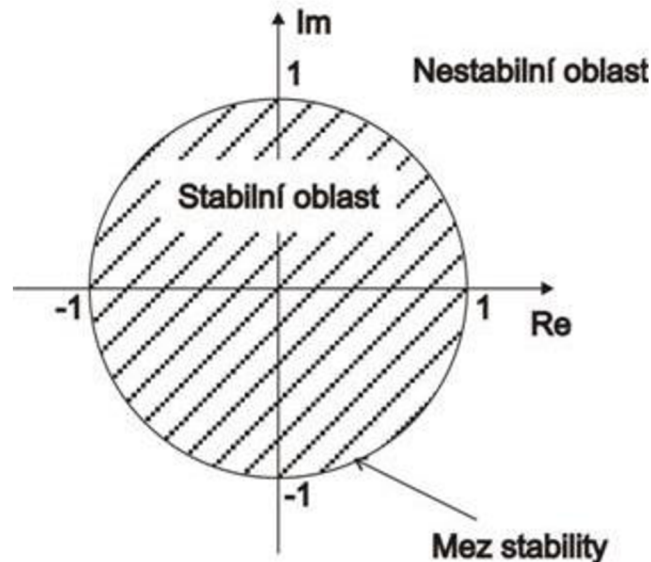
$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^r L_i \cdot z^{r-i}}{\sum_{i=1}^m K_i \cdot z^{m-i}} \quad (3)$$

Přenosová funkce je označena  $H(z)$ . Posloupnost  $X(z)$  je vstupním signálem. Posloupnost  $Y(z)$  je výstupním signálem číslicového filtru. Přenosová funkce se dá také rozdělit na část s póly a nulami. Póly jsou ve jmenovateli a jsou označeny  $K_i$  se zpožděním  $z^{m-i}$ . Nuly jsou v čitateli zlomku a jsou označeny  $L_i$  se zpožděním  $z^{r-i}$ .



## 2.2 Stabilita číslicového filtru

Číslicový filtr je stabilní, jestliže jeho výstupní odezva na každý konečný vstupní signál je rovněž konečná [3]. Tuto podmínku stability zajistíme nastavením jednotlivých pólů v přenosové rovnici daného filtru. Všechny póly číslicového filtru musí ležet v jednotkové kružnici, v opačném případě se jedná o nestabilní systém, v našem případě nestabilní číslicový filtr. Jednotková kružnice je vyobrazena na obr 6.



obr 6. Určení stabilního číslicového filtru. Převzato z [10]

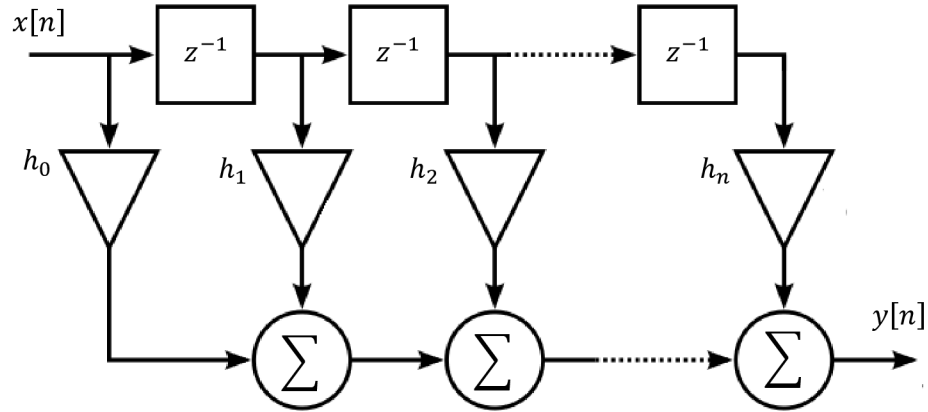
Výše uvedený obr 6 zobrazuje stabilní a nestabilní oblast diskrétního systému, do kterých spadá číslicový filtr. Obrázek se skládá z kružnice a dvou částí, tou je stabilní a nestabilní oblast. Pro nestabilní systém postačí, aby alespoň jeden pól ležel mimo tuto kružnici. Jednotková kružnice, která tvoří přechod mezi stabilní a nestabilní oblastí se nazývá mez stability. Pokud má systém pól, nebo póly na této kružnici je potřeba použít početních metod, které buď potvrdí, nebo vyvrátí celkovou stabilitu systému.

## 2.3 Filtr s konečnou impulzní odezvou

FIR filtr (finite impulse response) neboli filtr s konečnou impulzní odezvou, někdy nazýván také jako nerekurzivní filtr. Stabilita tohoto filtru je vždy zaručena díky přenosové rovnici filtru (5), ze které je patrné, že tento číslicový filtr má pouze několika násobný pól umístěn v počátku. Podle obr 6 je jasné, že jediný pól FIR filtru leží ve stabilní oblasti a splňuje tedy podmínku o stabilitě.

Na rozdíl od IIR filtrů jsou tyto filtry realizovatelné pouze číslicově a nemají obdobnou analogovou formu.

FIR filtr je sestaven ze tří základních bloků, viz obr 5. Příklad blokového schématu FIR filtru je na následujícím obr 7.



obr 7. Blokové schéma FIR filtru. Převzato z [9]

Z blokového schématu na obr 7 je patrné, že FIR filtr nemá zpětnou vazbu, jedná se tedy o nerekurzivní filtr. Z obrázku se dá odvodit tvar diferenciální rovnice:

$$y_n = \sum_{n=0}^{N-1} x_n h_n \quad (4)$$

Tato rovnice vyjadřuje hodnoty impulzní charakteristiky, které jsou přímo systémové realizační konstanty. Rovnice představuje vztah mezi posloupností vstupních proměnných  $x_n$  a posloupností impulzní odezvy  $h_k$ . Odezva impulzní charakteristiky je na jednotkový skok [1].

Z diferenciální rovnice se dá vypočítat obrazový přenos FIR filtru

$$H(z) = \sum_{n=0}^{N-1} h_n z^{-n} \quad (5)$$

Tento přenos soustavy je dán pouze nulami, dále má n-násobný pól v počátku, který způsobuje zpoždění filtru.

Dalším popisem FIR filtru je pomocí frekvenční charakteristiky.

$$G(\omega) = H \cdot (e^{j\omega T}) = \sum_{n=0}^{N-1} h_n \cdot e^{-jn\omega T} \quad (6)$$

Frekvenční charakteristika je periodická s periodou  $\frac{2\pi}{T}$  a dá se tedy vyjádřit Fourierovou řadou s koeficienty  $h_n$ . Jeden ze způsobů návrhu filtru toho typu je za pomoci frekvenční charakteristiky, tato návrhová metoda bude více popsána v následujících kapitolách. Z konečného počtu složek Fourierovy řady vyplývá, že charakteristika je spojitá funkce až na hranice přenášeného pásma. [1]

Při navrhování FIR filtru je v praxi žadané, aby měly FIR filtry fázovou frekvenční charakteristiku přesně lineární. Pokud má filtr lineární fázovou charakteristiku, dochází při zpracování časové řady ke skupinovému zpoždění. Toto zpoždění se projevuje na každé harmonické složce a tvar časové řady není zkreslen. Této linearitě se dá dosáhnout splněním jedné z uvedených podmínek impulzní charakteristiky diskrétního systému. [11]

$$h_n = h_{(N-1-n)} \quad \text{nebo} \quad h_n = -h_{(N-1-n)} \quad (7)$$

První podmínka platí pro symetrickou impulzní charakteristiku, druhá podmínka platí pro antisymetrickou impulzní charakteristiku.

Jednotlivé typy jsou shrnuty v následující tabulce. Tabulka 3 udává příklady použití jednotlivých typů, vzhledem k počtu vzorků a podmínce impulzní charakteristiky. Pomocí Eulerova vztahu se jednotlivé typy mohou popsat rovnicí frekvenční charakteristiky, tyto rovnice jsou uvedeny v literatuře [11].

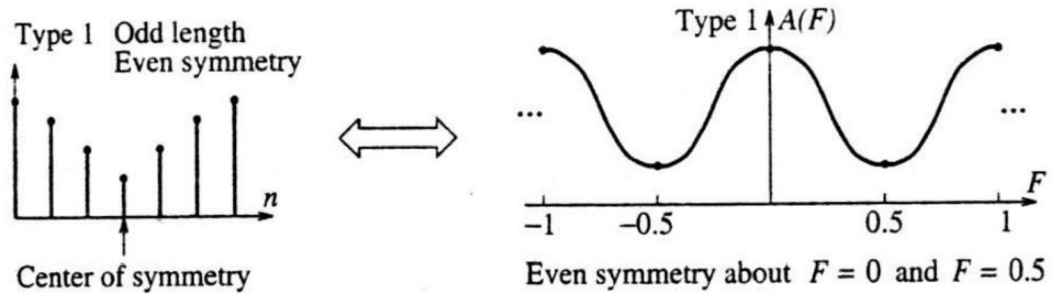
Tabulka 3. Shrnutí jednotlivých typů FIR filtrů

Typ	Číslo obrázku	Impulzní charakteristika	Počet vzorků	Použití filtrů (DP, HP, PP, PZ)
1	obr 8	$h_n = h_{(N-1-n)}$	Lichý	Všechny typy
2	obr 9	$h_n = h_{(N-1-n)}$	Sudý	Nehodí se pro HP
3	obr 10	$h_n = -h_{(N-1-n)}$	Lichý	Pouze pro PP
4	obr 11	$h_n = -h_{(N-1-n)}$	Sudý	Pouze pro PP a HP

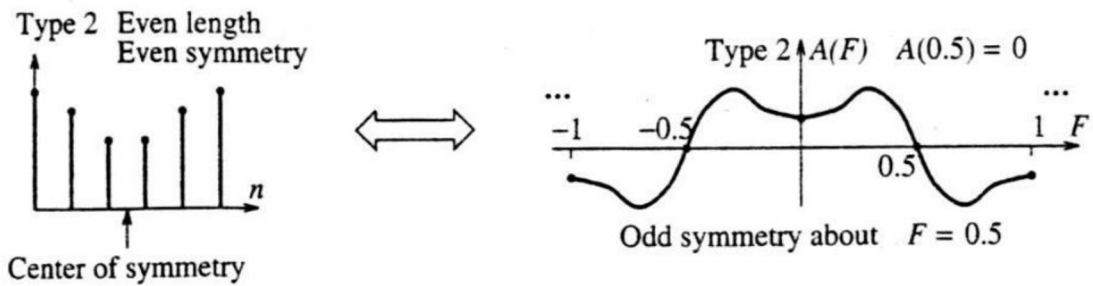
Podle předchozí tabulky jsou na následující straně 4 obrázky, každý popisuje jeden z uvedených 4 druhů FIR filtrů.

U jednotlivých obrázků jsou 2 charakteristiky. Na levé straně je zobrazena impulzní charakteristika a počet vzorků, zda se jedná o lichý nebo sudý počet vzorků.

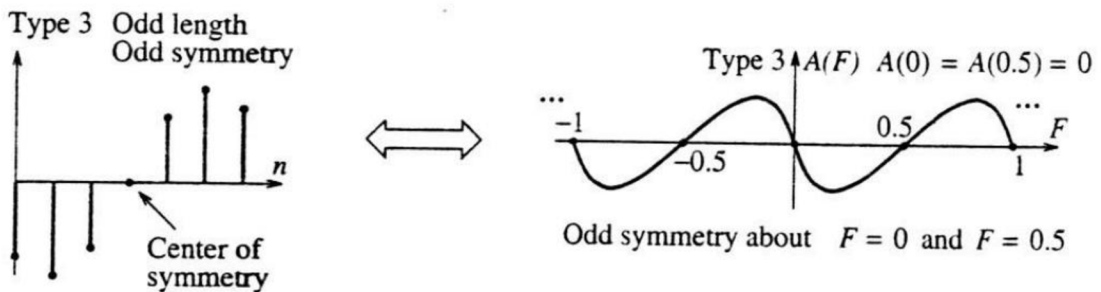
Pravá strana zobrazuje amplitudovou charakteristiku jednotlivých typů filtrů spolu se symetrií signálu.



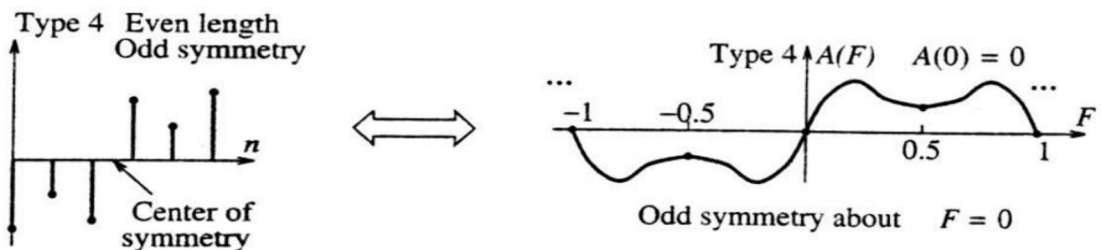
obr 8. FIR filtr s lineární fází, typ 1. Převzato z [1]



obr 9. FIR filtr s lineární fází, typ 2. Převzato z [1]



obr 10. FIR filtr s lineární fází, typ 3. Převzato z [1]



obr 11. FIR filtr s lineární fází, typ 4. Převzato z [1]

## 2.4 Metody návrhu FIR filtru

K návrhu se používají nejčastěji tyto metody [3] [16]:

- Metoda vzorkování frekvenční charakteristiky
- Metoda váhové posloupnosti (metoda oken)
- Optimalizační metoda – Equi ripple

U prvních dvou metod je obrovská nevýhoda, že mají obtížné nastavování přesných hranic propustného a nepropustného pásma. Přesných hodnot frekvenčních hranic se dosahuje vyšším řádem filtru. Tyto metody jsou pro svoji jednoduchost návrhu hojně využívány. [18]

Pro přesné nastavování hranic je potřeba využít optimalizační metodu návrhu filtru FIR.

Kvůli použití v programu LabVIEW bude v následující práci podrobněji popsána pouze metoda váhové posloupnosti a optimalizační metoda Equi-ripple.

### 2.4.1 Metoda váhové posloupnosti

Tato metoda váhové posloupnosti je v praxi nejvíce využívanou metodou, pomocí které se navrhují filtry typu FIR [2]. Metoda je také někdy nazývána jako metoda oken, protože pro omezení nekonečně dlouhé impulzní charakteristiky se používají tzv. okna.

Předpokladem metody jsou známé požadavky ve frekvenční oblasti, ve které má filtr pracovat. Pro jednoduchost je zadána žádoucí frekvenční charakteristika vyjádřená Fourierovou řadou s periodou  $\frac{2\pi}{T}$ :

$$G(\omega) = H(e^{j\omega T}) = \sum_{n=-\infty}^{\infty} h_n \cdot e^{-j\omega n T}, \quad (8)$$

Kde  $\omega$  značí, že se jedná o periodickou funkci. Koeficienty  $h_n$  jsou přímo hledanými koeficienty impulzní charakteristiky a tedy i přenosové funkce. Pro koeficienty  $h_n$ , které jsou zanedbatelně malé, se zvolí vhodná aproximace a nemusí být dále počítány. [18] [1]

$$h_n = \frac{T}{2\pi} \cdot \int_{-\frac{\pi}{T}}^{\frac{\pi}{T}} G(\omega) \cdot e^{j\omega n T} \cdot d\omega \quad (9)$$

Protože posloupnost  $h_n$  je nekonečně dlouhá je potřeba ji omezit na rozsah  $N$  členů vynásobením celého spektra tzv. oknem, které má určitou konečnou délku. Obecná rovnice okna je ve tvaru v rovnici (10).

$$w(t) = \sum_{n=n_0}^{\frac{N-1}{2}} w_n \delta(t - nT), \quad (10)$$

Po vynásobení oknem dostaneme zkrácenou impulzní charakteristiku. Z této zkrácené impulzní charakteristiky lze výslednou frekvenční charakteristiku získat pomocí diskrétní Fourierovy transformace.

V rovnici (10) je hodnota  $n_0$  použita, když se jedná o zadanou reálnou frekvenční charakteristiku, v tomto případě se  $n_0 = \frac{-N}{2} + 1$ . [1]

Pro návrh touto metodou je velmi důležitý výběr okna, každé okno má své výhody i nevýhody, proto se pro různé případy vždy může hodit jiný typ okna. Okna, která je možná použít v programu LabVIEW jsou více popsáné v [15]. Zde je také matematický popis jednotlivých oken, spolu s důkladnějším popisem vlastností daného okna.

## 2.4.2 Optimalizační metoda Equi-ripple

Metoda nese název Equi-ripple, neboli metoda rovnoměrně zvlněných aproximací, kvůli stejným amplitudám všech vln v propustné části navržené frekvenční charakteristiky [12]. Tato metoda je založená na aproximaci nastavených frekvencí pomocí aproximace Chebyshev a následného minimalizačního algoritmu Parks McClellan.

Metoda funguje na principu minimalizování maximální chyby navržené funkce od ideální funkce filtru pomocí Parks-McClellan algoritmu. Tento algoritmus je založen na minimalizování rozdílu chyby mezi aktuální frekvenční charakteristikou navrženého filtru a frekvenční charakteristikou ideálně navrhnutého filtru, podle zvolených frekvenčních rozsahů. [11] [12]

Proto jsou hledány takové koeficienty pomocí Chebyshev aproximace, které mají nejmenší maximální chybu. Po nalezení nejlepších koeficientů je sestavena aproximační funkce a navržen výsledný filtr [11].

Pomocí této metody lze nastavit přesné hranice propustných a nepropustných pásem. Kvůli iteračnímu postupu a neexistenci analytického vzorce je tato metoda náročná na výpočetní techniku. Velkou výhodou této metody je kromě nastavení přesných hranic propustnosti, také možnost použití nejnižšího řádu filtru pro dané zadání oproti předcházejícím metodám. V literatuře [11] a [12] je obsažen podrobnější popis spolu s příklady výpočtu pro tuto metodu.

## 2.5 Filtr s nekonečnou impulzní odezvou

IIR filtr (infinite impulse response) je filtr s nekonečnou impulzní odezvou. Tento typ filtru je vždy rekurzivní, neboli v zapojení tohoto filtru se používají zpětné vazby. Využití těchto filtrů je daleko větší oproti filtrům FIR díky tvaru přenosové rovnice. [4]

Kvůli bohatší rovnici na póly a nuly je tu ale na rozdíl od FIR filtrů problém se stabilitou. Stabilitu filtru je proto tedy nutné hlídat. Filtry typu IIR nemají lineární průběh fázové frekvenční charakteristiky, mohou tedy různě fázově ovlivňovat všechny harmonické složky signálu. [11]

Výhodou IIR oproti FIR filtru je menší řádová náročnost filtru, kdy je filtrovaný signál s podobnou odchylkou obou typů filtrů. Tento předpoklad celkově zjednodušuje nárok na hardware a výpočetní techniku, převážně na paměť při výpočtu stavových proměnných. [2]

Další velkou výhodou IIR filtrů je možnost dosahování strmějších přechodů mezi propustným a nepropustným pásmem, při stejném řádu filtru oproti filtrům FIR. Díky této strmosti mají filtry IIR menší zpoždění, při stejných požadavcích na tvar výstupní frekvenční charakteristiky. [6] [11]

IIR filtr se dá popsat diferenciální rovnicí ve tvaru [4]:

$$y_n = \sum_{i=0}^r L_i x_{n-i} - \sum_{i=1}^m K_i y_{n-i}, \quad (11)$$

kde  $L_i$  a  $K_i$  jsou systémové koeficienty, konkrétně  $K_i$  koeficient zpětných vazeb, protože má v diferenční rovnici zpožděný krok o hodnotu 1. Konstanta  $L_i$  značí koeficient dopředných vazeb. Koeficient  $r$  odpovídá počtu zpoždění v nerekurzivní části a obdobně hodnota  $m$  počet zpoždění v rekurzivní části, toto zpoždění udává celkový řád systému. Celkový výstup IIR filtru je značen  $y_n$ .

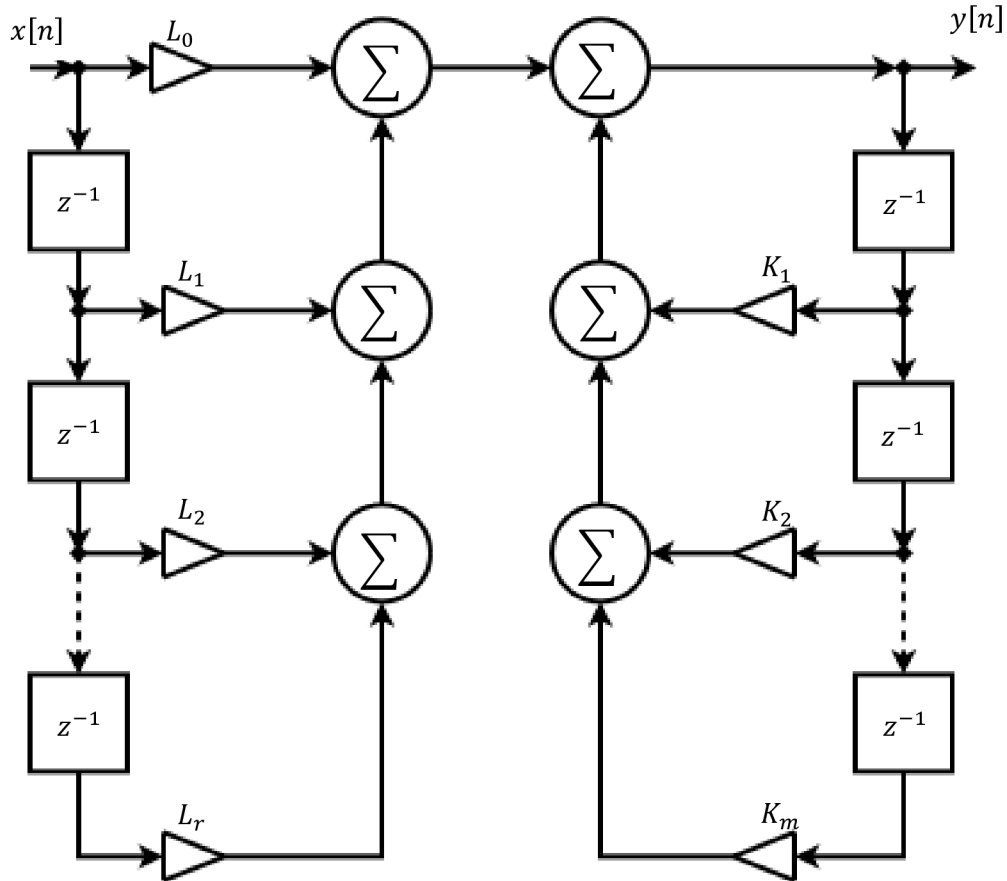
Z diferenční rovnice se dá vypočítat obrazový přenos systému [4] [1]:

$$H(z) = \frac{\sum_{i=0}^r L_i \cdot z^{r-i}}{\sum_{i=0}^m K_i \cdot z^{m-i}} = A \cdot \frac{\prod_{i=1}^r (z - n_i)}{\prod_{i=1}^m (z - p_i)} \quad (12)$$

Rovnice (12) popisuje operátorový přenos soustavy, zde je rovnice dána podílem jednotlivých polynomů, ve kterých jsou definované jednotlivé systémové koeficienty  $L_i$  a  $K_i$ . Tato rovnice se dá roznásobit do kořenového tvaru, ze kterého jsou snadno patrné hodnoty jednotlivých pólů  $p_i$  a nul  $n_i$  spolu s koeficientem zesílení  $A$ .

Tvar frekvenční charakteristiky je dán hodnotami  $H(z)$ , ten je v literatuře [1].

IIR je rekurzivní filtr, má tedy zpětnou vazbu. Pomocí tří základních bloků nadeřinovaných na obr 5 vypadá obecné blokové schéma IIR filtru podle následujícího obr 12.



obr 12. Blokové schéma IIR filtru

V levé části je dopředná vazba IIR filtru, tato část je stejná jako u předchozího typu číslicového filtru. Pravá část se označuje rekurzivní. Počet zpoždění v rekurzivní části udává celkový řád systému.

## 2.6 Metody návrhu IIR filtru

Pro návrh IIR filtru se dají metody návrhu rozdělit na 2 hlavní kategorie: [1] [3]

- Optimalizační metody
- Metoda založená na podobnosti s analogovým filtrem



### 2.6.1 Optimalizační metody návrhu IIR filtrů

Tyto metody jsou z časového hlediska velmi náročné, protože rovnice nejsou řešeny explicitně, neboli přímo. Optimalizační metody vychází z vytvoření tzv. účelové funkce popisující IIR filtr. Funkce je dále minimalizována až na určité optimum, podle požadovaných vlastností filtru. [3]

Díky minimalizovanému tvaru jednotlivých funkcí pro dané metody lze dosáhnout poměrně nízkého řádu přenosových funkcí, spolu s ostatními požadavky, jako jsou například požadavky na kvantování, ty jsou detailněji popsány v literatuře [1] a [3].

Jednotlivé optimalizační metody jsou popsány v literatuře [18], kde jsou i konkrétní příklady výpočtu.

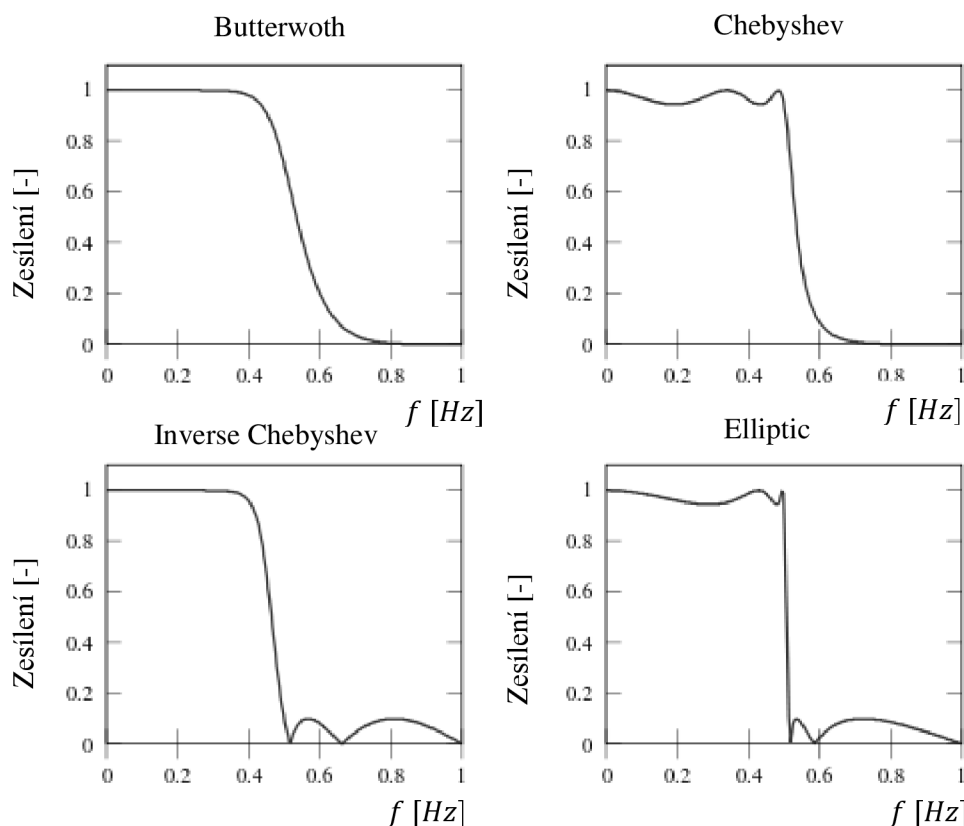
### 2.6.2 Návrh IIR podle podobnosti s analogovým systémem

V počátcích číslicových filtrů se tento druh návrhu číslicových filtrů hojně využíval k získání potřebných výsledků, protože materiály k problematice analogových filtrů jsou rozsáhle zpracovány se všemi potřebnými daty. Tato metoda je tedy založená na převzetí aproximací používaných při návrhu analogových filtrů.

Mezi základní filtry, odvozené z metod založených na podobnosti s analogovým systémem patří: Butterworthovy, Besselovy, Čebyševovy nebo Caurovy eliptické filtry.

Důkladnější popis metod, které převádí analogové filtry na digitální, jsou popsány v literatuře [1], [3] a [14].

IIR filtry, které se vyskytují v prostředí LabVIEW mají návrh filtrů založený pouze na podobnosti s analogovými filtry. Tyto filtry využívají aproximaci, díky níž má nastavený druh IIR filtr specifické vlastnosti. Na následujícím obr 13 jsou vykresleny základní aproximace, které se vyskytují v programu LabVIEW.



obr 13. Frekvenční charakteristika základních aproximací. Převzato z [7]

Na obr 13 je vykreslena frekvenční charakteristika základních aproximací. V levém horním rohu obr 13 je aproximace Butterwoth, tato aproximace má hladké zesílení v propustné části, používá se především ve filtrech, kde je potřebné nulové zvlnění v propustném pásmu. Velkou nevýhodou této aproximace je široké přechodové pásmo mezi propustným a nepropustným pásmem. [8] [17]

Pravý horní roh obr 13 zobrazuje Chebyshev aproximaci. Tato charakteristika je poněkud více zvlněná v propustné části, nežli u předchozí aproximace. V nepropustné části je hladká. Filtr navržen touto aproximací nepotřebuje tak vysoký řád, jako filtr navržen aproximací Butterwoth. [8] [17]

Aproximace typu Inverse Chebyshev je v levém dolním rohu obr 13. Tato aproximace má zvlněnou část v pásmu potlačení signálu. Kromě tohoto zvlnění na druhé straně pásma má podobné vlastnosti jako Chebyshev. [8] [17]

Pravý dolní roh obr 13 vykresluje Elliptic aproximaci. Tato charakteristika má nejstrmější charakteristiku. Nevýhodou je zvlnění v propustném pásmu a několik nul v pásmu potlačení signálu. Jsou počítány pomocí eliptických integrálů, podle kterých nese tato aproximace název. [8] [17]

## 3 NÁVRH PROGRAMU

Podle zadání bakalářské práce má být zrealizován program, který je určen hlavně pro výukové účely studentů Vysokého učení technického. Program by měl umožňovat návrh vstupního signálu, spolu s celkovým návrhem obou typů filtrů. U jednotlivých filtrů by mělo být na výběr typ návrhu filtru a nastavení frekvenční oblasti, ve kterých bude filtr pracovat.

Pro snadnější návrh budou v programu při nastavování filtru vykreslovány pomocné charakteristiky, díky kterým se uživatel dokáže lépe zorientovat v návrhu filtru. Mělo by se jednat hlavně o frekvenční charakteristiku filtru, ze které bude zřetelné, v jaké oblasti dochází k ovlivňování frekvenčních spekter. Dalším grafem bude fázová frekvenční charakteristika a vykreslení všech pólů a nul nastaveného filtru. Při nastavování filtru bude současně vykreslen vstupní a filtrovaný signál. Tento graf poslouží k tomu, zda je filtr navržen podle očekávání.

Dalším důležitým aspektem programu by mělo být zobrazení celkového porovnání číslicových filtrů obou typů. Mělo by být zřetelné, jaký typ filtru se pro nastavený vstupní signál více hodí. Kritéria musí zhodnotit celé spektrum výhod a nevýhod, proto pro celkové porovnávání bylo navrženo několik kritérií, které hodnotí filtry z více úhlů pohledu. Kritéria budou více popsány v následující kapitole.

Kvůli přesnosti jednotlivých kritérií je potřeba určit, jaký signál si přeje uživatel na výstupu. Kvůli nepatrným chybám a zpomalení celého programu, je programové odfiltrování nechtěných složek signálu nesmysl, proto si uživatel při návrhu vstupního signálu také nastaví tzv. Skutečný signál. Tento skutečný signál se v bakalářské práci vyskytuje, jako uživatelem chtěný signál po výstupu z filtrace.

### 3.1 Kritéria pro posouzení navrženého filtru

Pro vyhodnocení vhodnosti použití filtru byly zvoleny následující kritéria:

- Početní náročnost
- Otáčení fáze signálu
- Řád filtru a jednotlivý počet pólů a nul
- Skutečná odchylka filtrovaného signálu
- Poměr signálu vůči šumu – SNR

### 3.1.1 Početní náročnost

Početní náročnost se dá určit několika způsoby, v této bakalářské práci byla zvolena metoda časové prodlevy. Odečet početní náročnosti spočívá v zjištění časové doby, po kterou bude počítač počítat a nastavovat řád filtru. Jednoduše řečeno doba výpočtu nastaveného typu filtru.

Kvůli dnešním technologiím, tedy dobrému výpočetnímu výkonu se odečtená doba bude pohybovat v řádech milisekund. LabVIEW nabízí možnost odečtu časového zpoždění pouze v milisekundách. Těmito hodnotami v milisekundách by vznikaly velké odchylky, které by mohli dosahovat i řádů stovek procent. Tento problém bude ošetřen použitím  $x$  zacyklení výpočtu filtrů, tedy časové prodloužení  $x$ -krát. Tento způsob mnohonásobně zlepšuje přesnost výpočtu.

Hodnota zacyklení, by měla být nastavitelná, kvůli použití programu na mnoho zařízeních s jiným výpočetním výkonem.

Toto kritérium bude pouze se souhlasem uživatele a to na dobu, která bude dostatečná k vyhodnocení výsledků. Uživatel bude moci kdykoli zapnout, nebo vypnout zacyklení programu, aby nebyl limitován zpomalenou rychlostí při další práci v programu.

### 3.1.2 Otočení fáze signálu

Jakmile je na navrhnutý vstupní signál použit číslicový filtr, dochází k otáčení fáze a časovému zpoždění. Pro zjištění otočení fáze bude nutné vytvořit podprogram, který umožní vyčíslit hodnotu pootočení signálu 1. harmonické vůči nastavenému skutečnému signálu. Tento podprogram bude mít velké využití i u posledního kritéria a to zjištění odchylky signálu.

Kromě číselného výpisu bude použito i grafické zobrazení, kde se porovnají 4 signály, kterými jsou filtrovaný signál FIR, filtrovaný signál IIR, nastavený vstupní signál a nastavený skutečný signál.

Podle tohoto kritéria by mělo být patrné, jakým způsobem filtry ovlivňují fáze jednotlivých harmonických složek vstupního signálu.

### 3.1.3 Řád filtru a jednotlivý počet pólů a nul

Řád filtru se nastavuje přímo v zadávání filtru, proto se toto kritérium zaměří hlavně na počet pólů a nul a jejich rozložení v rovině  $Z$ .

Graf pro rozložení pólů a nul má určitou velikost a předpokládá se, že studenti nebudou nastavovat obrovské řady filtrů. Kdyby k nastavení velkého řádu došlo, zcela by se znehlednil graf, který zobrazuje polohu pólů a nul. Druhou stránkou nastavení velkého řádu je například přesnější odečet kritéria časové náročnosti. Musí se tedy najít optimální hodnota řádu, která bude dostačující pro všechna kritéria.

### 3.1.4 Odchylka filtrovaného signálu

Nepochybně důležitým kritériem je skutečná odchylka filtrovaného signálu oproti skutečnému signálu. Toto kritérium bude složitější na výpočet, ale bude hodně charakterizovat daný typ filtru.

Důležitým krokem pro výpočet odchylky filtrovaného signálu je odstranění fázového posunu 1. harmonické filtrovaného signálu, které by jinak znehodnocovalo výsledky. Jakmile bude skutečný a filtrovaný signál se stejným fázovým posunem 1. harmonické složky, bude možné určit rozdíl signálu. Rozdíl skutečného a filtrovaného signálu se při správném nastavení filtru bude blížit k hodnotě nula.

Předpokladem je, že pro menší odchylku filtrovaného signálu bude potřeba vyšší řád filtru, nebo nastavit frekvenční oblast filtru s dostatečným odstupem od frekvencí skutečného signálu. Důležitost tohoto kritéria má velkou váhu oproti ostatním, ale samotné vyhodnocení filtru a určení významnosti jednotlivých kritérií, už bude na samotném uživateli.

### 3.1.5 Poměr signálu vůči šumu – SNR

Kritérium SNR (signal to noise ratio) má dva stavy. Jakmile bude na chybovém signálu přidán šum, bude funkce SNR sloužit ke změření schopnosti filtru odstranění šumu ze signálu vůči skutečnému signálu. Zde je výsledek počítán v decibelech a v procentech vůči počáteční amplitudě šumu.

Pokud tento šum nebude do vstupního signálu přidán, kritérium SNR vypočte zesílení, nebo útlum filtrovaného signálu vůči skutečnému signálu. Zde jsou opět hodnoty vyčísleny v decibelech, nebo v procentech vůči skutečnému signálu.

## 4 SOFTWAREVÉ PROSTŘEDÍ

### 4.1 LabVIEW

LabVIEW neboli (Laboratory Virtual Instrument Engineering Workbench) přeloženo do češtiny „laboratorní pracoviště virtuálních přístrojů“. Tento program je od americké firmy NI. Prostředí LabVIEW slouží k programování systémů, ale také pro měření a analýzu signálů, nebo například k programování složitých systémů jako je robot [13].

Prostředí je formou grafického jazyka, to spočívá v názorném ukázání kódů pomocí různých prostředků, se kterými uživatel pracuje a podle kterých tvoří kód. Tento grafický jazyk má viditelnou posloupnost programu, díky čemu se snáze odlaďují chyby a zbytečné kusy kódu oproti textovým programovacím prostředím.

Samotné programování v LabVIEW není složité. Program má jen základní ovládací tlačítka jako například run, pause, abort a podobně. Ty slouží k základnímu ovládnutí, více o programu a programovacím jazyce je v [13].

Program LabVIEW má dva základní panely, v kterých se vytváří program:

- Front panel.
- Block diagram.

### 4.2 Front panel

Front panel je přeložen do češtiny jako přední panel a slouží ke komunikaci se samotným uživatelem. V tomto panelu se pomocí vstupních ovládacích prvků a výstupních indikátorů pracuje v programu. Ovládací prvky se jednoduše vybírají z palety ovládacích prvků, kde jsou řazeny podle svých kategorií.

Mezi příklady vstupních ovládacích prvků patří různá tlačítka a přepínače, díky nimž uživatel spouští a ovládá program. Dalšími vstupními prvky jsou datové, mezi ně patří: textové řetězce, číselné veličiny, pole nebo třeba cluster, jednoduše řečeno vstupní data, která jsou potřebná pro daný program.

Výstupními indikátory jsou led diody, které mohou signalizovat stavy, dále to jsou grafy a jiné zobrazovače. Pomocí těchto výstupních indikátorů se dají odečítat potřebné hodnoty. Výstupem z programu jsou i textové a číselné zobrazovače, díky nimž uživatel ihned vidí potřebné parametry a hodnoty. Všechny ovládací prvky a datové prvky jsou více popsány v [13], kde je i detailnější popis programu LabVIEW.

## 4.3 Block diagram

Block diagram se překládá do češtiny jako blokový diagram neboli panel, kde je samotný kód programu. Tento kód se tvoří pomocí grafického propojování bloků z předního panelu, které tvoří vstupní proměnné nebo vložené funkce z palety funkcí. Jednotlivé spojení se vytváří pomocí drátů, které mají barvu podle datového typu. Díky pospojování vstupů, funkcí a výstupů vzniká grafický program, podobný blokovému schématu.

Jak už bylo uvedeno, do blokového diagramu se vkládají funkce, které jsou jako u předního panelu rozděleny do několika kategorií. Pro výpočty nabízí paleta funkcí spoustu matematických operací, také goniometrické funkce a podobně. Jako u každého vyššího programovacího jazyku i zde nechybí pokročilé funkce jako cykly, case, event, nebo například sequence a tak dále. Díky kombinaci cyklů a struktury case jde například sestavit stavový automat, který má spoustu možností uplatnění. Program má také možnost pracovat s poli, textovými řetězci nebo soubory [13].

Velice přínosnou vlastností LabVIEW je používání rozšiřujících karet k programu, díky kterým může uživatel pracovat s reálnými signály nebo ovládat výstupní reálné systémy. Jakmile je program dokončen, dá se ihned zkompileovat a spustit.

## 5 REALIZACE PROGRAMU

Vytvořený program byl pomocí programu LabVIEW ve verzi z roku 2013. Kvůli přehlednosti celého kódu, ale také zabránění opakovatelnosti jednotlivých částí kódu, bylo k hlavnímu programu *Číslicová filtrace-výuková simulace.vi* vytvořeno několik podprogramů. Celý program se tedy skládá z celkově 12 podprogramů a jednoho hlavního programu.

Zde je celý seznam vytvořených podprogramů:

- *Celkové využití CPU.vi*
- *Fázové posunutí.vi*
- *Nastavení osy X.vi*
- *Ošetření stejné velikosti všech polí.vi*
- *Póly Nuly FIR.vi*
- *Póly Nuly IIR.vi*
- *Průměr z hodnot.vi*
- *Přepočet na dB.vi*
- *SNR.vi*
- *Volba vstupní skutečného signálu.vi*
- *Vstupní chyby FIR.vi*
- *Vstupní chyby IIR.vi*

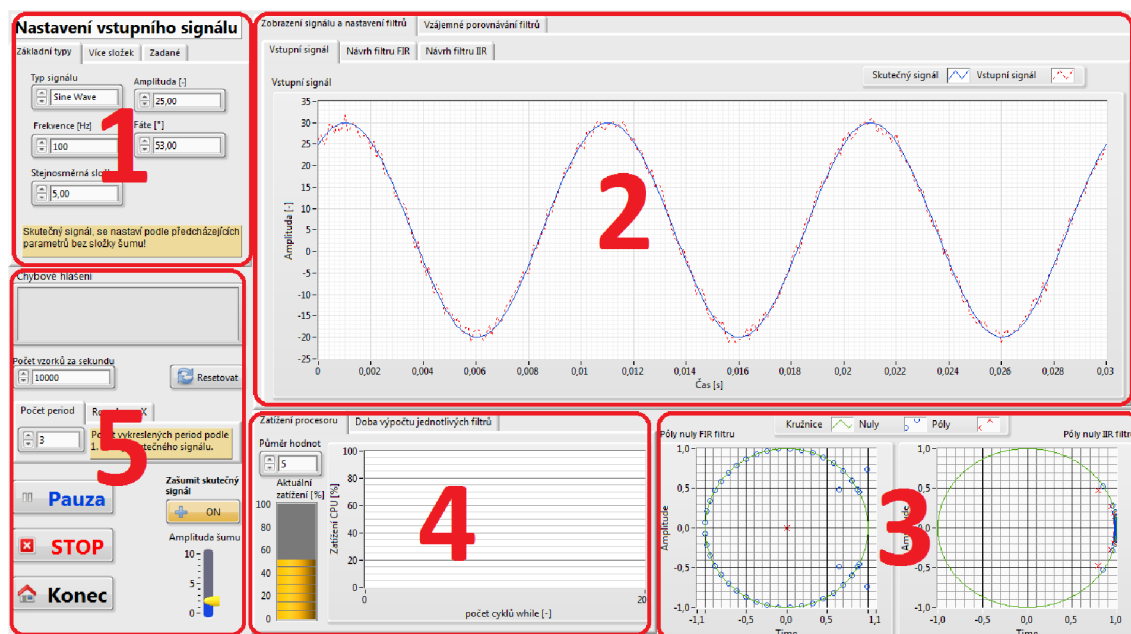
### 5.1 Uživatelské rozhraní

Vstupní okno programu je tvořeno z pěti hlavních částí, což je zobrazeno na obr. 14. Příloha A zobrazuje pro důkladnější náhled uživatelské rozhraní ve větší velikosti.

Části programu jsou:

- Nastavení vstupního signálu – 1. část
- Nastavení a vzájemné porovnávání navržených filtrů – 2. část
- Vykreslování pólů a nul obou typů filtru – 3. část
- Zatížení počítače při výpočtu filtrů a filtraci – 4. část
- Ovládací prvky programu – 5. část





obr 14. Rozdělený program na části

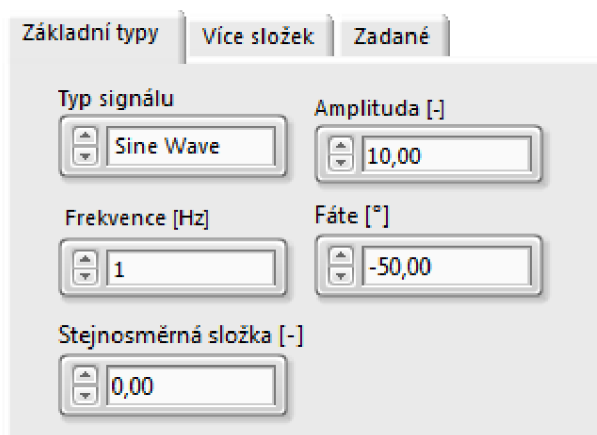
Kvůli více možnostem nastavení, nebo například přepínání mezi vykreslováním jednotlivých oken mají části 1, 2, 4, 5 v sobě zakomponované záložky. Tyto záložky jsou vždy umístěny v horní části jednotlivých částí programu. Jednotlivé části jsou více popsány v následujících kapitolách.

## 5.2 Nastavení vstupního signálu

V programu jsou tři možnosti nastavení vstupního signálu. Těmito možnostmi je základní nastavení, nastavení pomocí více harmonických složek, ale také nastavení vstupního signálu zadáním textových příkazů. Jednotlivé možnosti nastavení budou více popsány v následujících kapitolách.

### 5.2.1 Nastavení vstupního signálu pomocí základních typů

Po zapnutí programu je v levé části možnost vybrání jedné z volby záložky pro nastavení vstupního signálu. První možnost záložky se jmenuje *základní typy*. Po za kliknutí této volby v záložce se objeví následující pod okno podle obr 15.

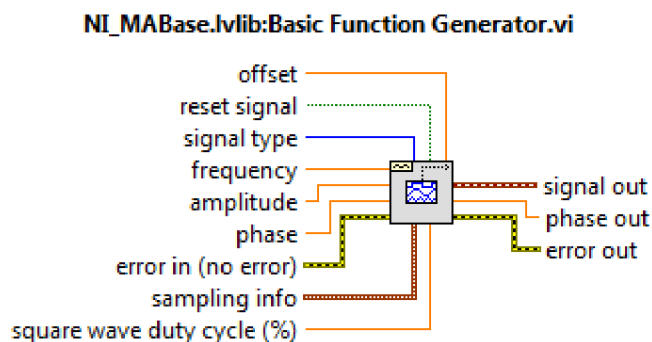


obr 15. Nastavení vstupního signálu pomocí základních typů

Zde je možné nastavovat základní druhy signálu - sinusový, pilovitý, obdélníkový a trojúhelníkový signál. Po výběru typu signálu může uživatel zvolit počáteční fázi, frekvenci a amplitudu signálu. Další možností je přidání stejnosměrné složky.

Skutečný signál se u této záložky bere stejný jako nastavený vstupní signál do filtru, jediná změna oproti nastavenému vstupnímu signálu zde může nastat přidáním šumu. V případě přidání šumu je skutečný signál podle nastavených parametrů bez složky šumu.

Na generování jednoduchých signálů byla použita funkce z knihovny *Signal Processing / Waveform Generation*. Z funkce jsou využity všechny možné vstupy. Výsledný signál je na výstupu *signal out*.



obr 16. Generátor základních signálů

## 5.2.2 Nastavení vstupního signálu pomocí více harmonických složek

Druhý výběr v přepínací záložce z první části podle obr 14 se jmenuje *Více složek*. Tato volba je vykreslena na následujícím obr 17.

Amplituda harmonických složek [-]	Frekvence harmonických složek [Hz]	Fázový posun harmonických složek [°]	Volba skutečného signálu
0	10	25	0
20	55	63	
1	63	91	
1	1	0	
0			

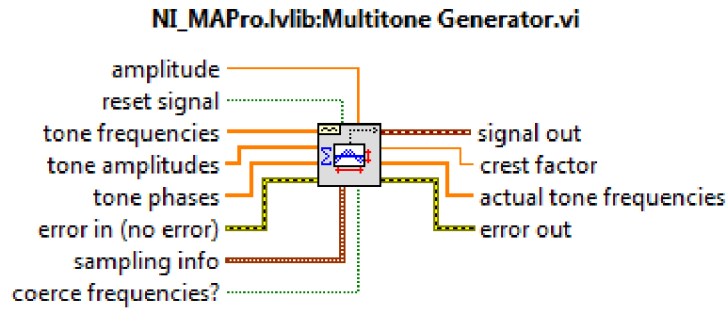
obr 17. Nastavení vstupního signálu poskládáním z více složek

V této podzáložce je možné sestavit vstupní signál pomocí součtu více sinusových signálů. Zde se nastavené signály zadávají do pole, při zadávání je důležité, aby první nastavená frekvence měla nejmenší hodnotu z celého pole frekvencí. Tento požadavek je z důvodu snadnějšího získání fáze skutečného signálu.

Při zadávání vstupního signálu výběrem touto metodou je důležité, aby všechna pole měla stejné rozměry. Pro ošetření, zda jsou všechna vstupní pole alokována na stejný počet prvků, byl vytvořen podprogram *Ošetření stejné velikosti všech polí.vi* (Příloha F). Tento podprogram při jiné velikosti libovolného pole vypíše uživateli varovné hlášení a zastaví zacyklením program, dokud nebudou mít všechna pole stejnou velikost.

Ke třem polím nastavující fázi, frekvenci a amplitudu bylo přidáno pole přepínačů. Pomocí za kliknutí některého přepínače se vybírá skutečný signál. Další podprogram *Volba vstupního skutečného signálu.vi* (Příloha H) odstraní nechtěné signály ze vstupních polí, aby vznikl skutečný nastavený vstupní signál.

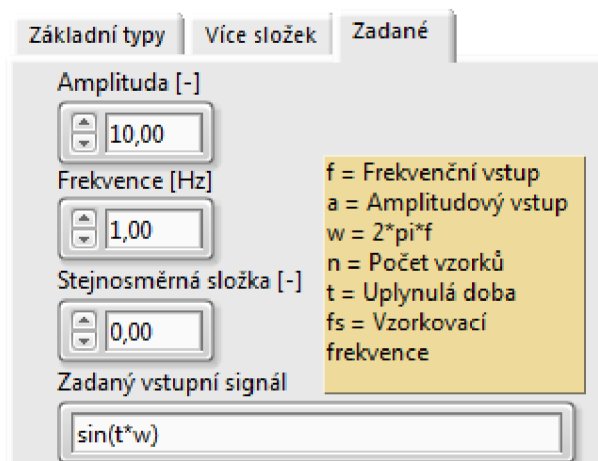
Na generování vstupního signálu touto metodou byla použita funkce z knihovny *Signal Processing/ Waveform Generation*



obr 18. Generátor signálu pomocí skládání více signálů

### 5.2.3 Nastavení vstupního signálu zadáním textu

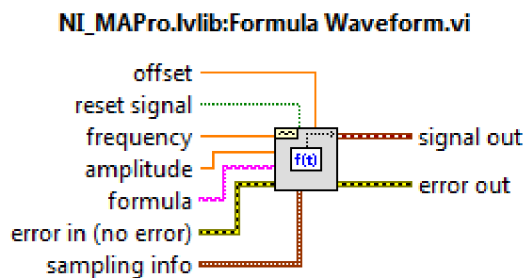
Poslední výběr ze záložky v první části celého programu je možnost zadat signál pomocí textových příkazů.



obr 19. Nastavení vstupního signálu zadáním textu

Z tohoto obr 19 je patrné, že vstupní signál se zde nastavuje pomocí textového řetězce. Tento textový řetězec podle čísel a klíčových znaků nastaví vstupní signál. V nápovědě v pravé části obrázku jsou zobrazené klíčové znaky, pomocí kterých je možné nastavovat signál. Písmeno  $f$  a písmeno  $a$  jsou pro odečet z nastavitelných polí s názvem *amplituda* a *frekvence*. Skutečný signál se zde nastavuje pouze zašuměním signálu jako v případě kapitoly 5.2.1.

Na generování vstupního signálu touto metodou zadávání textu byla použita funkce z knihovny *Signal Processing / Waveform Generation*



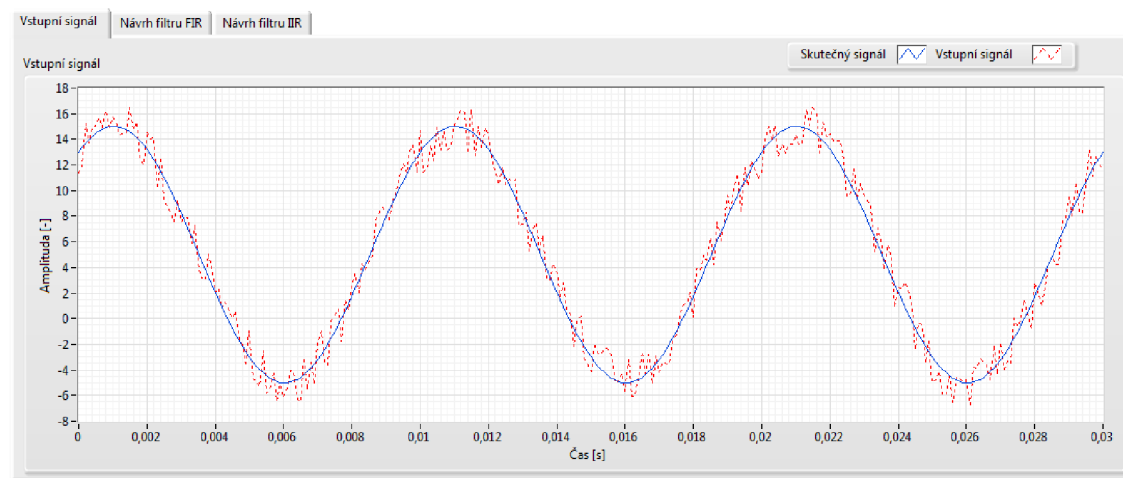
obr 20. Generátor signálu pomocí zadávání textem

## 5.3 Nastavení a vzájemné porovnávání navržených filtrů

Podle obr 14, je ve druhé části možnost zhlédnout horní a spodní záložky. Horní záložka přepíná mezi nastavovacím a vyhodnocovacím režimem. V této kapitole se první popíše nastavovací režim, kde je první volbou *vstupní signál*.

### 5.3.1 Zobrazení vstupních signálů

První touto možností v následující podzáložce je *Vstupní signál*.

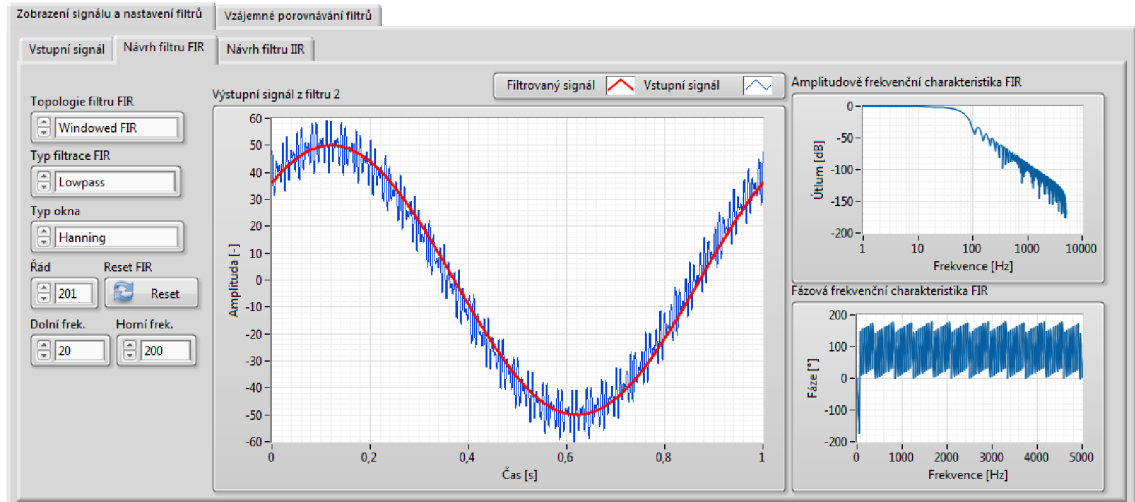


obr 21. Zobrazení vstupního a skutečného signálu

Tato volba na obr 21 slouží pouze k informativnímu zobrazení 2 signálů. Těmito signály jsou skutečný signál a nastavený vstupní signál pro filtraci. Tento signál je kompletní včetně šumu, pokud je tedy zakomponován do vstupního signálu.

### 5.3.2 Nastavení FIR filtru

Druhá volba v první záložce slouží k nastavení filtru FIR. Vlastnosti filtru jsou více popsány v teoretické části bakalářské práce, konkrétně v kapitole 2.3.



obr 22. Návrh filtru FIR

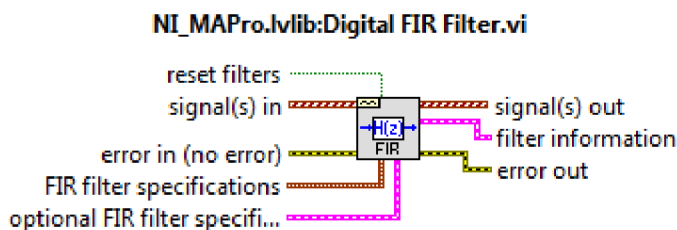
Jak je zřejmé na obr 22, tak v levé části se nastavují charakteristické konstanty a vlastnosti filtru FIR. První ovladač je pro nastavení typologie filtru. Zde jsou na výběr 2 možnosti návrhu filtru a možnost vypnout filtr. Možnost vypnout filtr, vypne filtraci signálu, takže se neprojeví žádné změny na výstupním signálu z filtru.

Návrhy filtrů jsou metodou oken a metodou Equi-ripple. Vlastnosti této optimalizační metody jsou více rozepsány v kapitole 2.4.1. Při nastavování filtru touto metodou je potřeba dodržování několika pravidel. Jedná se především o nastavování startovací a stopovací hranice DP a HP, kde je důležité dodržovat určité rozestupy mezi jednotlivými hranicemi. Pro nízké rozestupy dochází k rozkmitání propustných částí, pro velké rozestupy je nemožné nastavit filtr, a proto je uživatel upozorněn chybovým hlášením. Tento rozdíl startovací a stopovací hodnoty se zvyšujícím řádem filtru snižuje.

Program LabVIEW umožňuje uživateli při druhé volbě návrhu filtru na výběr několik druhů oken: Flat Top, Kaiser - Bessel, Blackman – Harris, Exact Blackman, Blackman, Triangular, Hamming a Hanning. Jednotlivé vlastnosti oken jsou více popsány v [14]. U nastavování řádu touto metodou návrhu je důležité dodržovat lichý řád více v [1] a [11]. Co se týče velikosti řádu filtru, tak pro striktnější dodržení hranic a strmější přechod spolu s větším útlumem nepropustných částí je lepší nastavovat vyšší řády, například kolem hodnoty 201.

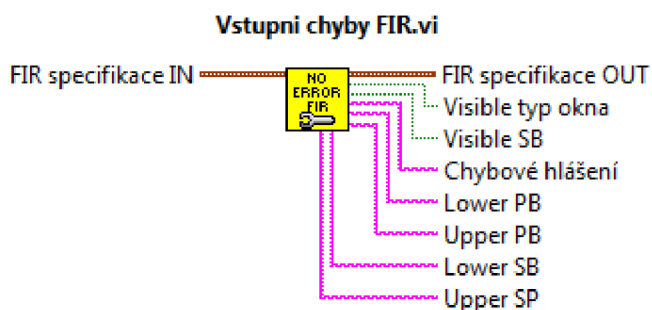
Pro nastavování je vhodné používat pomocné vykreslovací grafy v pravé části programu, kde je zřetelné, jak jsou jednotlivé hranice propustí nastaveny. Prostřední část čelního panelu (obr 22) je tvořena grafem se dvěma průběhy. První je nastavený vstupní signál a druhým signálem je filtrovaný signál, aby měl uživatel možnost okamžitě pozorovat změny signálu podle nastavení filtru.

Pro návrh FIR filtru byla použita funkce z knihovny *Signal Processing/ Waveform conditioning*.



obr 23. Filtr FIR v programu LabVIEW

Pro ošetření chyb nastavování filtru FIR byl vytvořen podprogram, který v případě chyby vypíše varovné hlášení a vypne filtr.

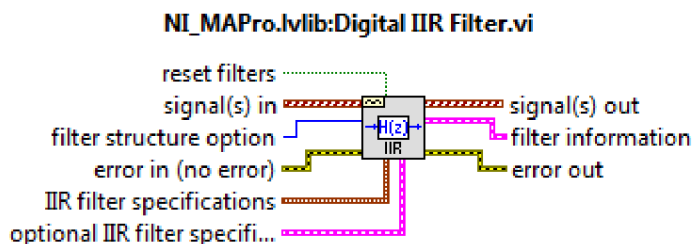


obr 24. Kontrola nastavení filtru FIR

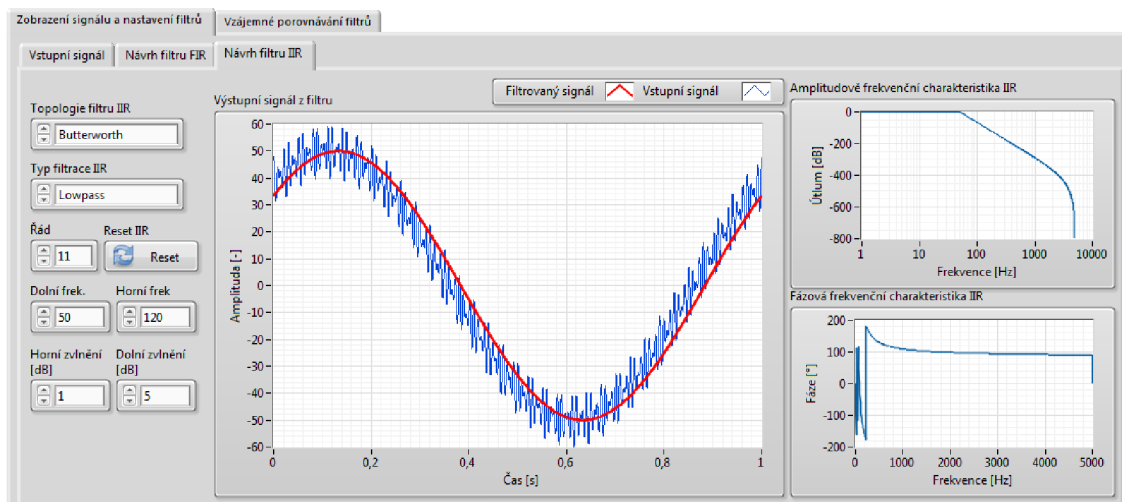
Podprogram (Příloha M) hlídá hodnoty nastavovaných frekvencí dolní a horní propusti pro obě metody návrhu. Kvůli zřetelnějšímu nastavování jsou přepsány nadpisy vstupních ovladačů frekvencí u metody Equi-ripple. Upravena je také viditelnost ovladačů, které nejsou potřebné pro danou typologii návrhu filtru. To je hlavně kvůli lepší orientaci v programu.

### 5.3.3 Nastavení IIR filtru

Poslední volbou v hlavním okně je možnost *Návrh filtru IIR*. Ukázka čelního panelu při této volbě návrhu filtru IIR je na obr 26. Pro filtraci signálu filtrem IIR byla vybrána funkce z knihovny LabVIEW. Konkrétně *Signal Processing/ Waveform conditioning*.



obr 25. Filtr IIR v programu LabVIEW

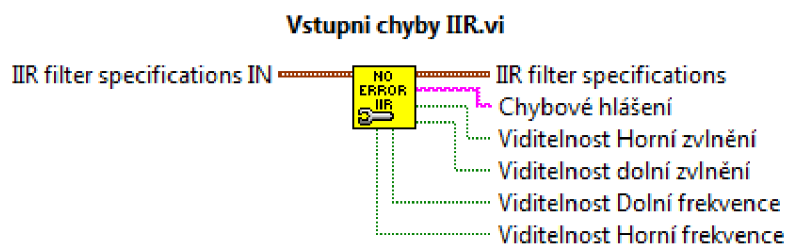


obr 26. Návrh filtru IIR

Jako u nastavení filtru FIR je zde podobné rozložení uživatelského prostředí. Grafy vykreslují stejné charakteristiky jako u filtru FIR. Rozdíl je pouze v nastavení filtru. V tomto případě se v *topologii filtru IIR* volí typ aproximace v návrhu filtru převzatého z analogových filtrů. Vlastnosti jednotlivých aproximací jsou více popsány v kapitole 2.6.2.

Pro nastavení filtru IIR se zde volí typ filtrace (DP, HP, PP, nebo PZ), řád filtru, dolní a horní frekvence a nastavení maximální horní a dolní zvlnění při zvolené aproximaci. Při nastavování řádu filtru IIR nejsou potřebné tak vysoké hodnoty řádu jako u filtru FIR, naopak postačují hodnoty řádu filtru okolo 10.

Pro ošetření chyb nastavování filtru byl vytvořen podprogram, který je na následujícím obr 27.



obr 27. Kontrola nastavení filtru IIR

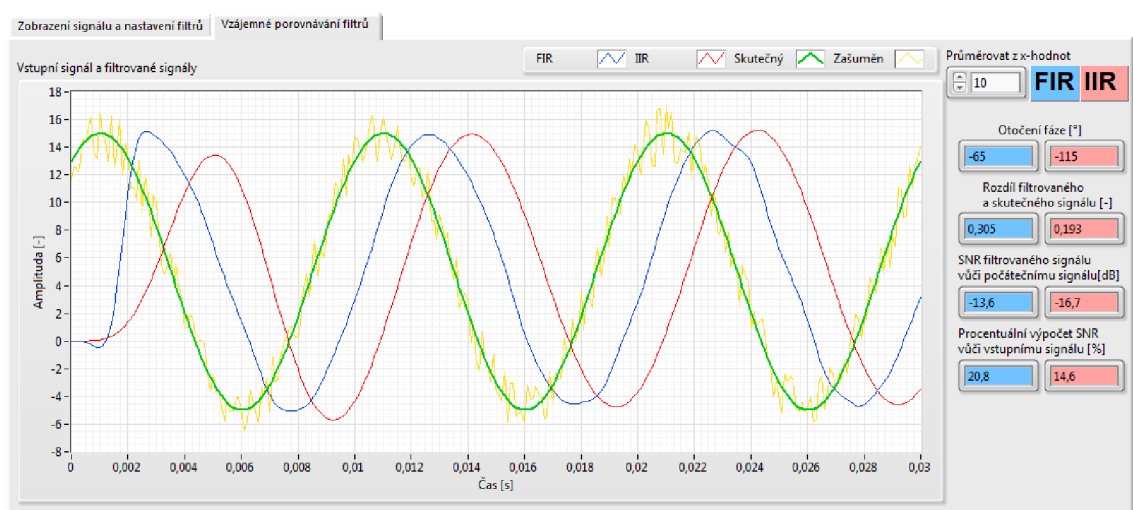
Tento program hlídá, zdali není horní frekvence nižší jako dolní frekvence. Jako u podprogramu pro kontrolu nastavení filtru FIR, se i zde upravuje viditelnost ovladačů, které nejsou potřebné pro nastavení filtru při zvolené topologii. Příloha J zobrazuje blokový diagram podprogramu z obr 27.



### 5.3.4 Vstupní a filtrované signály

V předchozích kapitolách 5.3.1, 5.3.2 a 5.3.3 byla popsána pouze první volba hlavní záložky druhé části programu (obr 14). Druhá možnost *Vzájemné porovnávání filtrů* slouží k porovnávání obou navržených filtrů podle několika kritérií. Tyto kritéria jsou uvedeny v předchozích kapitolách bakalářské práce.

Vyobrazení této možnosti záložky je na následujícím obr 28.



obr 28. Vstupní a filtrované signály

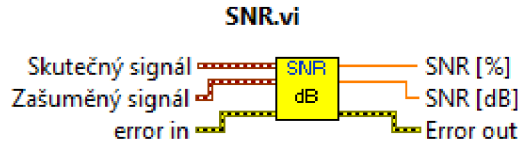
Levý graf slouží k zobrazení 4 signálů. Tyto 4 signály jsou výstupní signál z navrženého FIR filtru, výstupní signál z navrženého IIR filtru, vygenerovaný skutečný vstupní signál a vygenerovaný vstupní signál se šumem nebo bez něj. Tento graf slouží ke srovnání, jak vypadají výstupní signály z filtrů oproti skutečnému signálu. Také je patrné, jak jsou výstupní signály z filtru fázově posunuty oproti skutečnému navrženému vstupnímu signálu.

V pravé části obrázku jsou vypočítána kritéria pro jednotlivé filtry. Jsou zde vypočtené hodnoty otočení fáze, celkového rozdílu filtrovaného a skutečného signálu, ale také výpočet SNR pro jednotlivé filtry.

V pravé části je také možnost průměrování hodnot podle zadaného počtu, pro menší schodkovost programu. K průměrování hodnot byl naprogramován podprogram *Průměrování z hodnot.vi* (Příloha I).

Aby byl možný výpočet rozdílu a SNR jednotlivých filtrů, je potřeba sjednotit fáze nastaveného vstupního a filtrovaného signálu. Pro nastavení stejného fázového posunu 1. harmonické složky slouží podprogram *podprogram fázové posunutí.vi*. Příloha G zobrazuje blokový diagram tohoto podprogramu.

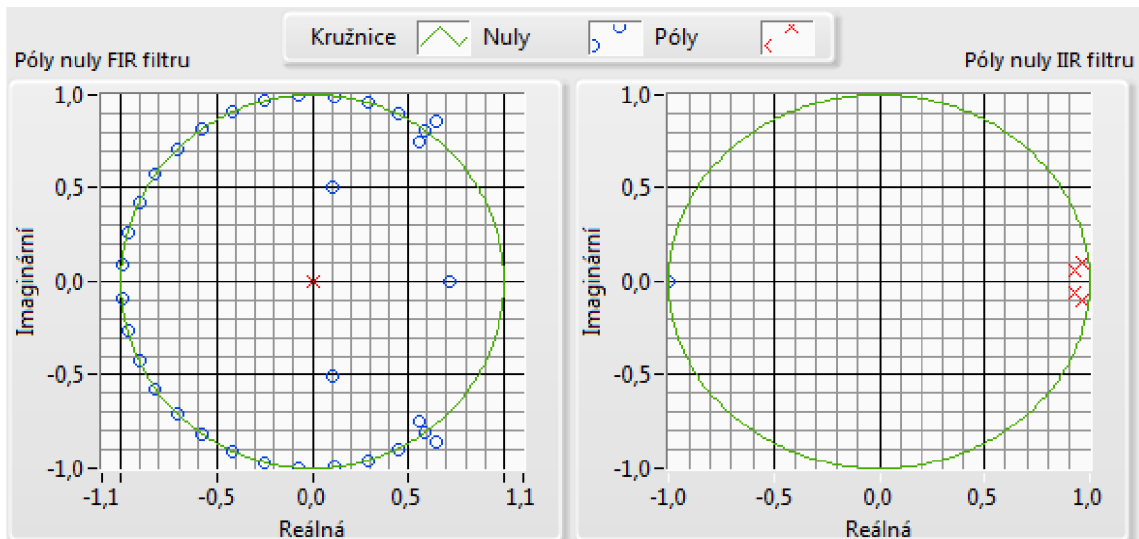
Výpočet SNR probíhá, i když je šum vypnutý. SNR je primárně pro počítání šumu, ale v případě jeho vypnutí se vypočtená hodnota SNR rovná porovnání efektivních hodnot skutečného a filtrovaného signálu. Pro výpočet se využívá podprogramu *SNR.vi* (Příloha O).



obr 29. Funkce pro výpočet SNR

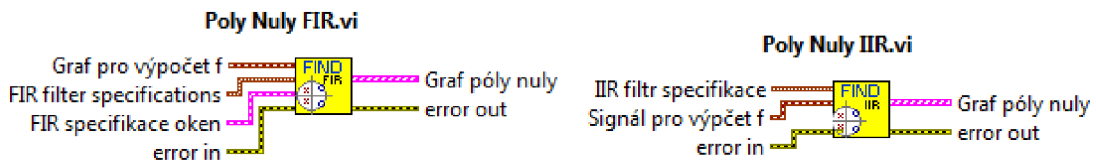
## 5.4 Vykreslování pólů a nul obou typů filtru

Program dále vykresluje póly a nuly navrženého filtru, viz obr 30.



obr 30. Vykreslování pólů a nul obou filtrů

Pro vykreslování pólů a nul má každý filtr svůj vlastní podprogram. Oba podprogramy pracují na podobném principu, ale kvůli různému nastavení filtrů bylo jednodušší pro každý filtr vytvořit samostatný podprogram. Příloha K – podprogram pro vykreslení pólů a nul filtru FIR. Příloha L – podprogram pro vykreslení pólů a nul filtru IIR.

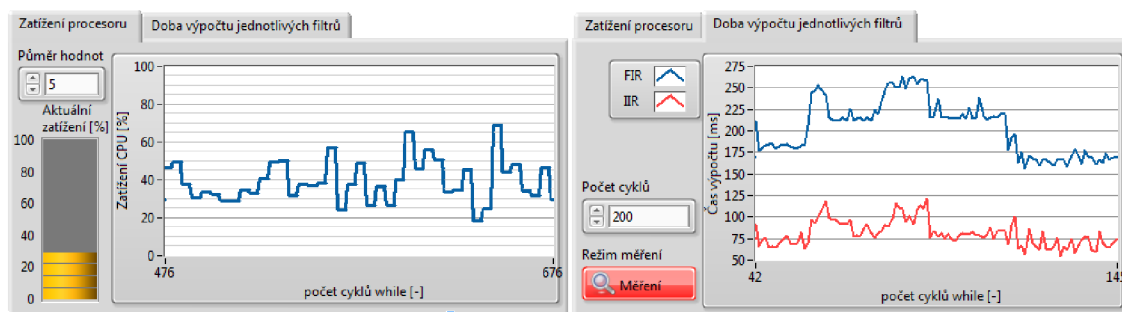


obr 31. Funkce pro získání pólů a nul

Podprogramy pracují na principu sestavení dopředných a zpětných koeficientů pro jednotlivé filtry. K tomu je potřeba vzorkovací frekvence, kterou lze získat z libovolného vstupního signálu. Koeficienty jsou ve tvaru pole, tyto pole jsou dále upravovány. Po úpravě dat se na výstup pošle klastr 3 polí. První dvě pole jsou pro dopřednou a zpětnou vazbu, poslední pole slouží k vykreslení jednotkové kružnice.

## 5.5 Zatížení počítače při výpočtu filtrů a filtraci

Aby měl uživatel přehled, jak celý program zatěžuje procesor, a jak je zvlášť doba filtrace pomocí filtru FIR a IIR náročná, byly do programu implementovány funkce pro zjištění těchto výpočtů. Následující obrázek ukazuje, jak zobrazení náročnosti vypadá.



obr 32. Zobrazení náročnosti programu a doby filtrace

Kvůli šetření místa v programu, je zde přepínání mezi celkovým zatížením procesoru a měření doby filtrace.

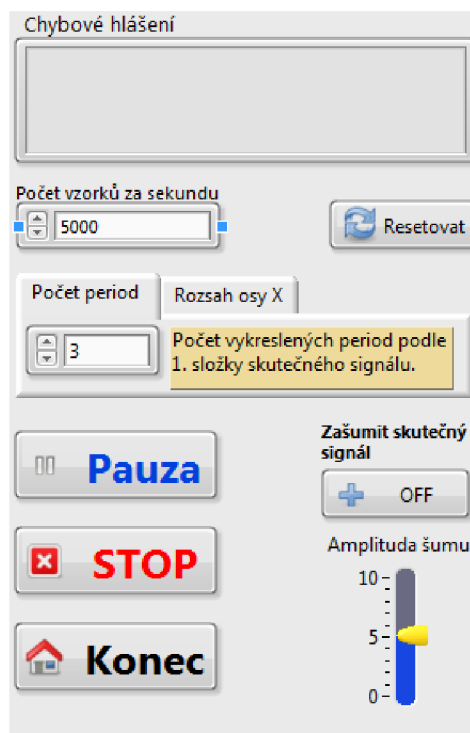
Pro měření zatížení celkového procesoru (levá část obr 32) byl vytvořen speciální podprogram *Celkové využití Cpu.vi* (Příloha D), který na výstupu zobrazuje aktuální hodnotu zatížení procesoru v procentech. Jelikož je tato hodnota hodně proměnlivá vzhledem ke každému kroku programu, byl vytvořen další podprogram *Průměrování z hodnot.vi* (Příloha I). Toto průměrování je nezávislé na běhu programu, čímž se program nezpomaluje.

Pravá část obr 32 zobrazuje možnost měření doby filtrace každého filtru zvlášť. Jak toto měření probíhá je více popsáno v návrhu kritérií, konkrétně v kapitole 3.1.1. Pro měření je důležité potvrdit stiskem tlačítka, které zacyklí filtraci na nastavenou hodnotu počtu cyklů. Po stisku tlačítka je chod celého programu výrazně zpomalen, proto se doporučuje používat toto měření pouze pro získání hodnot času výpočtu.

## 5.6 Ovládání programu

Poslední částí v uživatelském prostředí slouží k ovládání chodu programu, ale také k nastavení vykreslovacích grafů. Náhled této části je na obr 33.

Tato část obsahuje mnoho funkcí pro nastavení programu, ale taky položku pro vypisování chybových hlášení. Je zde možnost ovládní rozsahů os-x pro všechny vykreslovací periodické grafy. Uživatel si může vybrat, zda chce vykreslit určitý počet period skutečného signálu, nebo přímo zadat libovolný rozsah pro všechny grafy. Příloha E zobrazuje blokové schéma tohoto podprogramu.



obr 33. Část pro ovládní programu

Tato část také obsahuje přidávání šumu k vstupnímu signálu s nastavitelnou amplitudou. Tlačítko *Resetovat*, které resetuje signál na programově nastavené počáteční hodnoty jak vzorkovacích hodnot, tak i vlastností pro aktuální nastavení vstupního signálu z první části. Spodní 3 tlačítka slouží pro pauzu, pozastavení, nebo vypnutí celého programu LabVIEW.

## 6 OVĚŘENÍ FUNKČNOSTI PROGRAMU

Pro dostatečné ověření signálu byly použity dva druhy úloh. Prvním úkolem programu je odstranění šumu ze signálu a druhým úkolem odfiltrování nechtěných složek signálu. Vzorkovací frekvence byla nastavena na hodnotu 10 000 vzorků za sekundu, kvůli hladkým průběhům při frekvencích do 200 Hz.

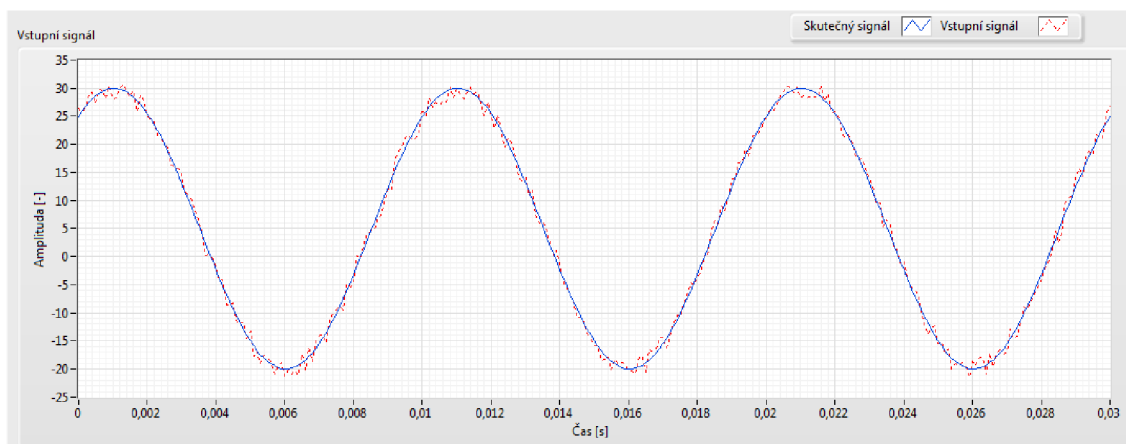
V této kapitole budou jednotlivé filtrace pomocí nastavených filtrů porovnávány podle kritérií, které byly uvedeny v teoretické části. Seznam těchto kritérií je následující: Početní náročnost, otáčení fáze signálu, řád filtru s počtem pólů a nul, odchylka celého signálu, SNR.

Nastavení filtrů bylo čistě experimentálně a mělo sloužit hlavně pro tyto příklady. Kvůli odlišení nastavených filtrů byl použit index u jednotlivých filtrů, který označuje číslo příkladu. Pro první příklad byl nastaven FIR1 a IIR1 pro druhý příklad FIR2 a IIR2.

### 6.1 První příklad testování programu

V tomto příkladu je hlavním úkolem odstranění šumu ze signálu.

- Metoda nastavení vstupního signálu – základní typy (sinusovka)
- Amplituda – 25
- Frekvence – 100 Hz
- Fáze –  $53^\circ$
- Stejnosečná složka – 5
- Šum – rovnoměrný šum, amplituda cca 2



obr 34. Ukázka prvního příkladu vstupního signálu

## 6.1.1 Nastavení filtrů pro filtraci prvního signálu

Filtry jsou voleny tak, aby byli schopny odstranit šum a měli obstojné výsledky, ale zároveň neměli příliš vysoké nároky na výpočet.

### FIR1

- Typ návrhu: metoda oken
- Typ filtrace: Dolní propust
- Typ okna: Blackman-Harris
- Řád filtru: 37
- Dolní frekvenční hranice: 120 Hz

Pro odfiltrování šumu prvního příkladu byl zvolen filtr FIR1 metodou oken. Tato metoda při nízkých řádech dosahuje nepřesných hranic filtrace, která je metodou stanovena na hodnotu přibližně okolo 200 Hz. Této špatné vlastnosti je využito v prospěch filtrace. Takto nastavený filtr dosahuje útlum -6dB okolo frekvence 640 Hz. V těchto frekvencích se již vyskytuje nastavený šum, proto je takto „špatně“ nastavený filtr schopen odstranit šum i při nízkém řádu filtru.

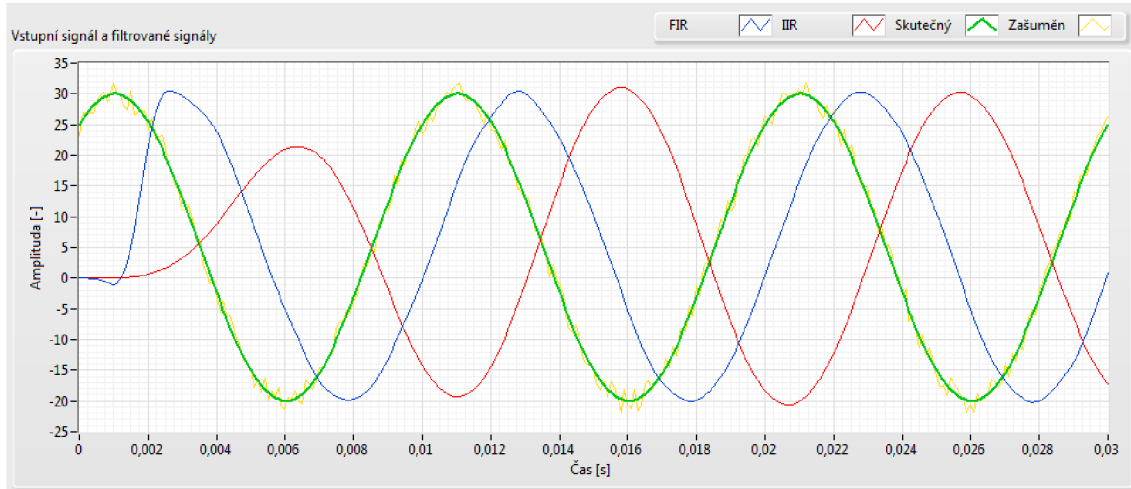
### IIR1

- Typ návrhu: Chebyshev
- Typ filtrace: Dolní propust
- Řád filtru: 5
- Dolní frekvenční hranice: 120 Hz
- Horní hranice zvlnění: 0,001 dB

Při nastavení filtru IIR byla zvolena aproximace typu Chebyshev, které se vyznačuje zvlněním v propustné části. Pro utlumení tohoto propustného zvlnění byla stanovena hodnota zvlnění na maximální hodnotu 0,001 dB, čímž pouze nepatrně ovlivňuje propustnou část. Tento filtr dosahuje útlum -6 dB již při frekvencích okolo 190 Hz

## 6.1.2 Výsledky filtrace prvního příkladu

Porovnání filtrovaných signálu spolu se skutečným signálem a zašuměným signálem je na obr 35 a vypočtené a odečtené hodnoty z grafů jsou vypsány v tabulce 4.



obr 35. Porovnání vstupních a filtrovaných signálu prvního příkladu

Tabulka 4. Porovnání výsledků u prvního příkladu

Kritérium	Početní náročnost (200x zacyklo)	Otočení fáze	Řád filtru	Odchylka	SNR	Výsledný šum
Jednotka	[ms]	[°]	[-]	[-]	[dB]	[%]
FIR1	240	-65	37	0,313	-16,3	15,4
IIR1	155	-116	5	0,245	-19	11,3

## 6.1.3 Zhodnocení výsledků prvního příkladu

Podle obr 35, je patrné fázové pootočení obou filtrů. Podle vyčíslených hodnot z minulé tabulky je jasné, že filtr FIR1 má skoro poloviční otočení fáze oproti filtru IIR1. Co se týče počátečního zpoždění, tak podle obr 35 filtr FIR1 opět dosahuje lepších výsledků, ale v celkové početní náročnosti je kvůli vyššímu řádu početně náročnější na výpočet než filtr IIR1.

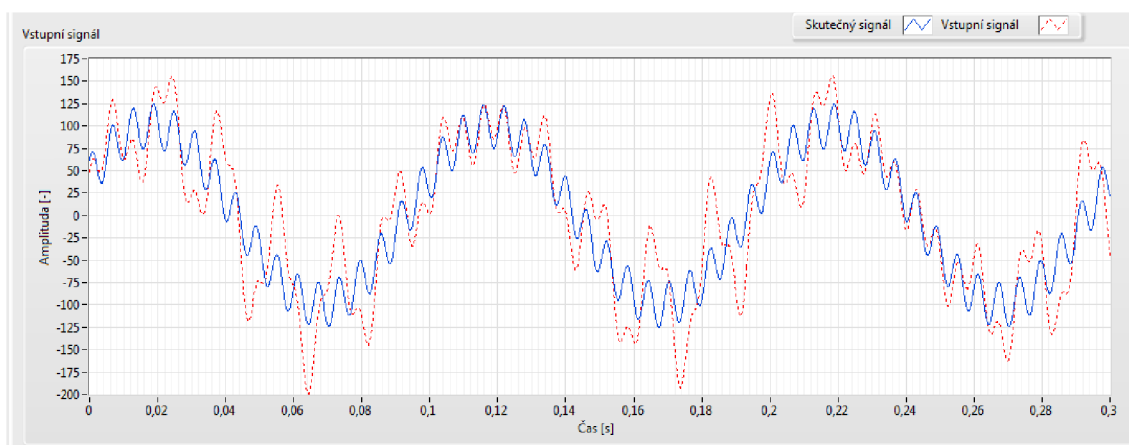
Co se týče nastavení filtrů, tak oba filtry dokázaly částečně odstranit nechtěný šum ze signálu a snížit výrazně jeho hodnotu. Střední hodnota šumu po filtraci byla snížena na 15% u FIR1 a 11% u IIR1. Kvůli malým hodnotám početní náročnosti, které se můžou zanedbat (velký počet zacyklení) dosahuje filtr IIR1 lepší výsledky hlavně co

se týče odstraňování šumu. Díky výraznému odstranění šumu pomocí obou filtrů na výstupním signálu je jak filtr IIR1 tak i filtr FIR1 správně navržen.

## 6.2 Druhý příklad testování programu

V tomto příkladu se otestuje schopnost odstranění dvou harmonických složek ze signálu. Pro jednoduchost je tento signál bez přidaného šumu.

- Metoda nastavení vstupního signálu – více složek
- Amplituda – 100, 35, 45, 25
- Frekvence – 10, 55, 63, 110 Hz
- Fáze – 25, 63, 265, 53°
- Skutečný signál – 1 a 4 složka
- Šum - vypnutý



obr 36. Ukázka druhého příkladu vstupního signálu

### 6.2.1 Nastavení filtrů pro filtraci druhého signálu

#### FIR2

- Typ návrhu: Equi-ripple
- Typ filtrace: Pásmová zadrž
- Řád filtru: 200
- Dolní frekvenční hranice: Start = 20 Hz, Stop = 50 Hz
- Horní frekvenční hranice: Stop = 70 Hz, Start 100 Hz

Protože v tomto případě je potřebné přesné nastavování frekvenčních hranic, kvůli odfiltrování určitých složek signálu, bez toho aniž by byla ovlivněna 1 a 4 složka signálu byla zvolena metoda návrhu filtru FIR2 typu Equi-ripple. Tato optimalizační



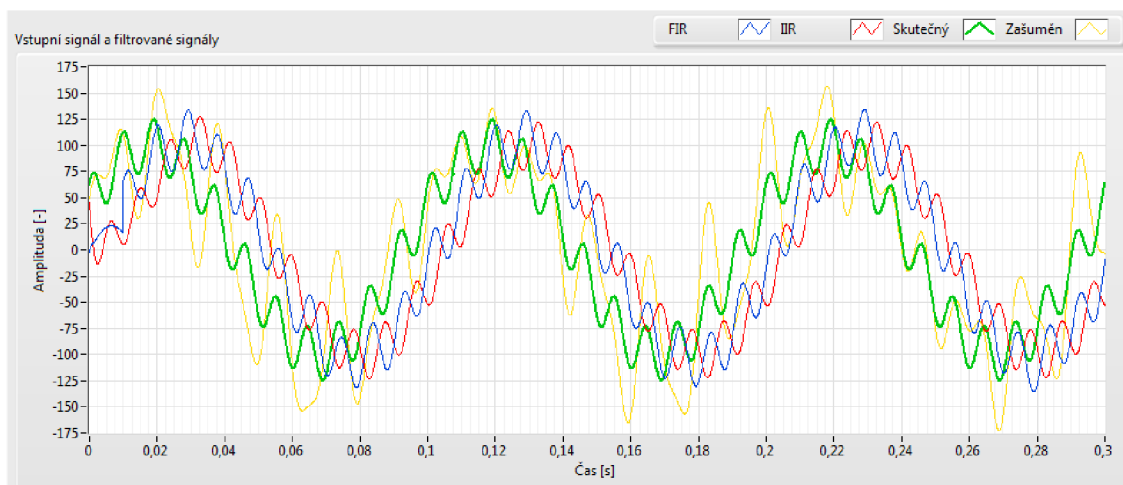
metoda návrhu je schopná přesného nastavení hranic filtrace, za předpokladu dostatečně vysokého řádu filtru. Nastavený filtr dosahuje útlumu -6 dB při frekvencích 30 Hz a 85 Hz. Kvůli vlastnostem metody návrhu, je filtr v propustné části zvlněn. Tyto výkyvy vln dosahují hodnot  $\pm 1$  dB, proto může být filtrovaný signál mírně zkreslen.

## IIR2

- Typ návrhu: Butterworth
- Typ filtrace: Pásmová zadrž
- Řád filtru: 5
- Dolní frekvenční hranice: 30 Hz
- Horní frekvenční hranice: 100 Hz

Co se týče filtru IIR2, tak byla zvolena aproximace typu Butterworth, kvůli nezvlněné propustné a nepropustné části. Pro tento návrh je útlum -6 dB dosažen v hodnotách frekvence 30 Hz a 95 Hz. Filtr nemá zvlněnou propustnou část jako filtr FIR2, ale zanedbatelného útlumu dosahuje až při frekvenci 140 Hz. Skutečný signál má harmonickou složku s hodnotou 110 Hz, kde frekvenční charakteristika filtru dosahuje útlumu -1,5 dB.

## 6.2.2 Výsledky druhého příkladu



obr 37. Porovnání vstupního a filtrovaných signálu druhého příkladu

Pro důkladnější porovnání jsou v následující tabulce vypočtené a odečtené hodnoty, charakterizující dané filtrace.

Tabulka 5. Porovnání výsledků u druhého příkladu

Kritérium	Početní náročnost (200x zacyklo)	Otočení fáze	Řád filtru	Odchylka	SNR (skut/filtr)	Výsledný šum
Jednotka	[ms]	[°]	[-]	[-]	[dB]	[%]
FIR2	620	-36	200	4,62	0,545	106,5
IIR2	105	-45	5	11,8	-0,0451	99,5

### 6.2.3 Zhodnocení výsledků druhého příkladu

Podle obr 37 je patrné, že u obou druhů filtrů dochází k fázovému pootočení, ale tyto hodnoty jsou si u obou filtrů podobné. Filtr FIR2 má obrovskou výhodu, že kvůli lineární fázi neovlivňuje fázově jiné harmonické složky. Proto filtrace signálu tímto filtrem dosahovala podstatně menších hodnot rozdílu mezi skutečným a filtrovaným signálem oproti filtru IIR2.

Kvůli zvlněné propustné části navrženého filtru FIR2 byla hodnota poměru skutečného a filtrovaného signálu 106 %, tedy zesiloval filtrovaný signál. Toto zvlnění by se dalo odstranit zvýšením řádu filtru. Filtr IIR2 díky nízkému řádu byl téměř 6x méně náročný na výpočet oproti filtru FIR2.

Kdyby byla zanedbána větší početní náročnost filtru FIR2, tak je tento nastavený filtr vhodnější pro tuto druhou úlohu.

# ZÁVĚR

Cílem Bakalářské práce bylo seznámení a popis číslicové filtrace, ale také popis dvou základních číslicových filtrů. Konkrétně se jedná o filtr s konečnou impulzní odezvou a filtr s nekonečnou impulzní odezvou. Dalším bodem zadání bylo vytvořit program, pomocí kterého si uživatel může navrhnout, ale také vyzkoušet číslicovou filtraci vstupního signálu. Program má podle zadání umožňovat vzájemné porovnání jednotlivých filtrů typu FIR a IIR podle několika kritérií, tak aby program mohl být použit jako výuková pomůcka.

Tato práce se dělí na teoretickou a programovou, neboli praktickou část. V teoretické části je obecně popis signálu, stability systémů a dělení filtrací z hlediska propustí. Další část teoretické části se zabývá řešerší IIR a FIR filtrů. V této části jsou popsány vlastnosti číslicových filtrů a rozdíl oproti analogovým filtrům. Závěrem teoretické části číslicových filtrů je popis metod, které slouží k návrhu jednotlivých číslicových filtrů v programu LabVIEW.

Praktická část začíná návrhem programu a popisem porovnávacích kritérií, kde se čtenář dozví, jak jsou jednotlivá kritéria počítána. Práce pokračuje stručným popisem programového prostředí LabVIEW. Následně se práce zaměřuje na samotný vytvořený program. V této části bakalářské práce je stručný popis uživatelského prostředí, spolu s nejdůležitějšími částmi programu. Celý program obsahuje několik podprogramů, ty jsou včetně celkového náhledu programu obsaženy v přílohách bakalářské práce.

Metod návrhu číslicových filtrů je v programu obsaženo několik, pro typ číslicového filtru FIR jsou v programu obsaženy dvě základní návrhové metody. Filtr IIR je možné navrhnout opět několika způsoby pomocí různě zvolených aproximací. U každého filtru může uživatel vybírat mezi jednotlivými propustmi a zádrží. Ke každému filtru se nastavují frekvenční hranice spolu s dalšími vlastnostmi. Celý program byl řešen pomocí cyklu while, takže jakákoli změna se ihned projeví v dalším cyklu programu.

Poslední kapitola popisuje odzkoušení navrženého programu a posouzení konkrétních příkladů navržených filtrů. Pro každý filtr byly odzkoušeny všechny možnosti návrhu, ale z důvodu stručnosti byly do práce zapracovány pouze dva příklady pro odzkoušení filtrů. V této kapitole jsou také shrnuty dosažené výsledky navrženými filtry, spolu se vzájemným porovnáním filtrů.

Využití vytvořeného programu vidím převážně v tom, že si uživatel může vyzkoušet konkrétní návrhy filtrů spolu s viditelnými výsledky při sebemenších změnách nastavení. Díky tomu může každý hlouběji proniknout do problematiky číslicové filtrace.

Práce by se dala dodatečně rozšířit o vzájemné porovnávání více filtrů najednou. Další rozšířením by mohla být filtrace externích signálů, spolu s možností ukládáním výstupních naměřených a vypočtených dat.

# Literatura

- [1] JAN, J. *Číslicová filtrace, analýza a restaurace signálů*. 2. upravené a rozšířené vydání Brno: VUTIUM, 2002. ISBN 8021415584.
- [2] JAKS, I. *Číslicové a měřicí systémy*. TU Liberec 2012
- [3] SMÉKAL Z. *Číslicové filtry*. Vydání 1. Nakladatelství VUT Brno 1993. ČP 2261. ISBN 80-214-0500-7
- [4] SOFKA P., ČMEJLA R., ŠMEJKAL L. *Číslicové filtry I*. ČVUT Praha. [online]. [cit. 17.11.2015]. Automatizace. Ročník 48. Číslo: červenec-srpen 2005. Dostupné z: [www.Technickýdeník.cz](http://www.Technickýdeník.cz)
- [5] SOFKA P., ČMEJLA R., ŠMEJKAL L. *Číslicové filtry II*. ČVUT Praha. [online]. [cit. 17.11.2015]. Automatizace. Ročník 48. Číslo: září 2005. Dostupné z: [www.Technickýdeník.cz](http://www.Technickýdeník.cz)
- [6] SOFKA P., ČMEJLA R., ŠMEJKAL L. *Číslicové filtry III*. ČVUT Praha. [online]. [cit. 17.11.2015]. Automatizace. Ročník 48. Číslo: říjen 2005. Dostupné z: [www.Technickýdeník.cz](http://www.Technickýdeník.cz)
- [7] Obrázek Frekvenční charakteristika základních aproximací [online]. [cit. 25.11.2015]. Dostupné z: [https://en.wikipedia.org/wiki/Butterworth\\_filter](https://en.wikipedia.org/wiki/Butterworth_filter)
- [8] *IIR filtry*. [online]. [cit. 17. 11. 2015]. Dostupné z: [http://proteus.fav.zcu.cz/~mautner/Azs/Azs9\\_Cislicova\\_filtrace\\_II\\_IIR.pdf](http://proteus.fav.zcu.cz/~mautner/Azs/Azs9_Cislicova_filtrace_II_IIR.pdf)
- [9] Obrázek FIR číslicového filtru. [online]. [cit. 17. 11. 2015]. Dostupné z: [https://en.wikipedia.org/wiki/Finite\\_impulse\\_response#/media/File:FIR\\_Filter.svg](https://en.wikipedia.org/wiki/Finite_impulse_response#/media/File:FIR_Filter.svg)
- [10] Obrázek stability číslicového filtru. [online]. [cit. 17. 11. 2015]. Dostupné z: <http://books.fs.vsb.cz/cislicovaregulace/data/kapitola6.html>
- [11] SCHWARZ D. *Lineární a adaptivní zpracování dat*. Vydání první. Vydalo Akademické nakladatelství CERM s.r.o. v Brně 15. Ledna 2012. ISBN 978-80-7204-779-6
- [12] SHENOI B. A. *Introduction to digital signal processing and filter design*, publikováno: Wiley-interscience, Rok vydání: 2006, ISBN-13 978-0-471-46482-2
- [13] VLACH, J, HAVLÍČEK, J. VLACH, M.. *Začínáme s LabVIEW*. Ilustrace Viktorie Vlachová. Praha: BEN - technická literatura, 2008. ISBN 9788073002459.
- [14] Oficiální stránky firmy National Instrument. [online]. <http://czech.ni.com/>
- [15] Digital Filter Desing Toolkit User Manual. Manuál od National Instrument. [online]. [cit. 24.1.2016]. Dostupné z: <http://www.ni.com/pdf/manuals/371353a.pdf>
- [16] STROUHAL, A. *Úzkopásmové filtry pro signály EKG*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 42s. Vedoucí bakalářské práce doc. Ing. Jiří Kozumplík, CSc.

- [17] JIRKŮ, T. *Počítačový návrh planárních filtrů*. VUT Brno, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky. Rok vydání 2006. Počet stran 118. Vedoucí bakalářské práce: prof. Dr. Ing. Zbyněk Raida
- [18] SYSEL, P. SMÉKAL Z. *Číslicové filtry*, Vydání první. Vydavatel VUT v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Rok vydání 2012, ISBN 978-80-214-4454-6

# Seznam příloh

- Příloha A. Čelní panel celého programu
- Příloha B. První část blokového diagramu celého programu
- Příloha C. Druhá část blokového diagramu celého programu
- Příloha D. Podprogram celkové využití CPU
- Příloha E. Podprogram pro nastavení osy X
- Příloha F. Podprogram pro ošetření stejné velikosti všech polí
- Příloha G. Podprogram fázového posunutí
- Příloha H. Podprogram pro nastavení volby skutečného signálu
- Příloha I. Podprogram pro průměr z hodnot
- Příloha J. Podprogram pro přepočet na dB
- Příloha K. Podprogram pro vykreslení pólů a nul FIR
- Příloha L. Podprogram pro vykreslení pólů a nul IIR
- Příloha M. Podprogram pro ošetření chybného nastavení filtru FIR
- Příloha N. Podprogram pro ošetření chybného nastavení filtru IIR
- Příloha O. Podprogram pro výpočet SNR

### Nastavení vstupního signálu

Základní typy: **Více složek** | Zadané

Typ signálu: Sine Wave | Amplituda [-]: 25,00

Frekvence [Hz]: 100 | Fáze [°]: 53,00

Stejnosečná složka [-]: 5,00

Skutečný signál, se nastaví podle předcházejících parametrů bez složky šumu!

Chybové hlášení

Počet vzorků za sekundu: 10000 | Resetovat

Počet period: 3 | Rozsah osy X: Počet vykreslených period podle 1. složky skutečného signálu.

**Pauza** | **STOP** | **Konec**

Zašumit skutečný signál: **ON** | Amplituda šumu: 10

### Zobrazení signálů a nastavení filtrů

Vzájemné porovnávání filtrů

Vstupní signál | Návrh filtru FIR | Návrh filtru IIR

Skutečný signál | Vstupní signál

Vstupní signál

Amplitude [-]

Čas [s]

### Zatížení procesoru

Doba výpočtu jednotlivých filtrů

Půměr hodnot: 5

Aktuální zatížení [%]

Zatížení CPU [%]

počet cyklů while [-]

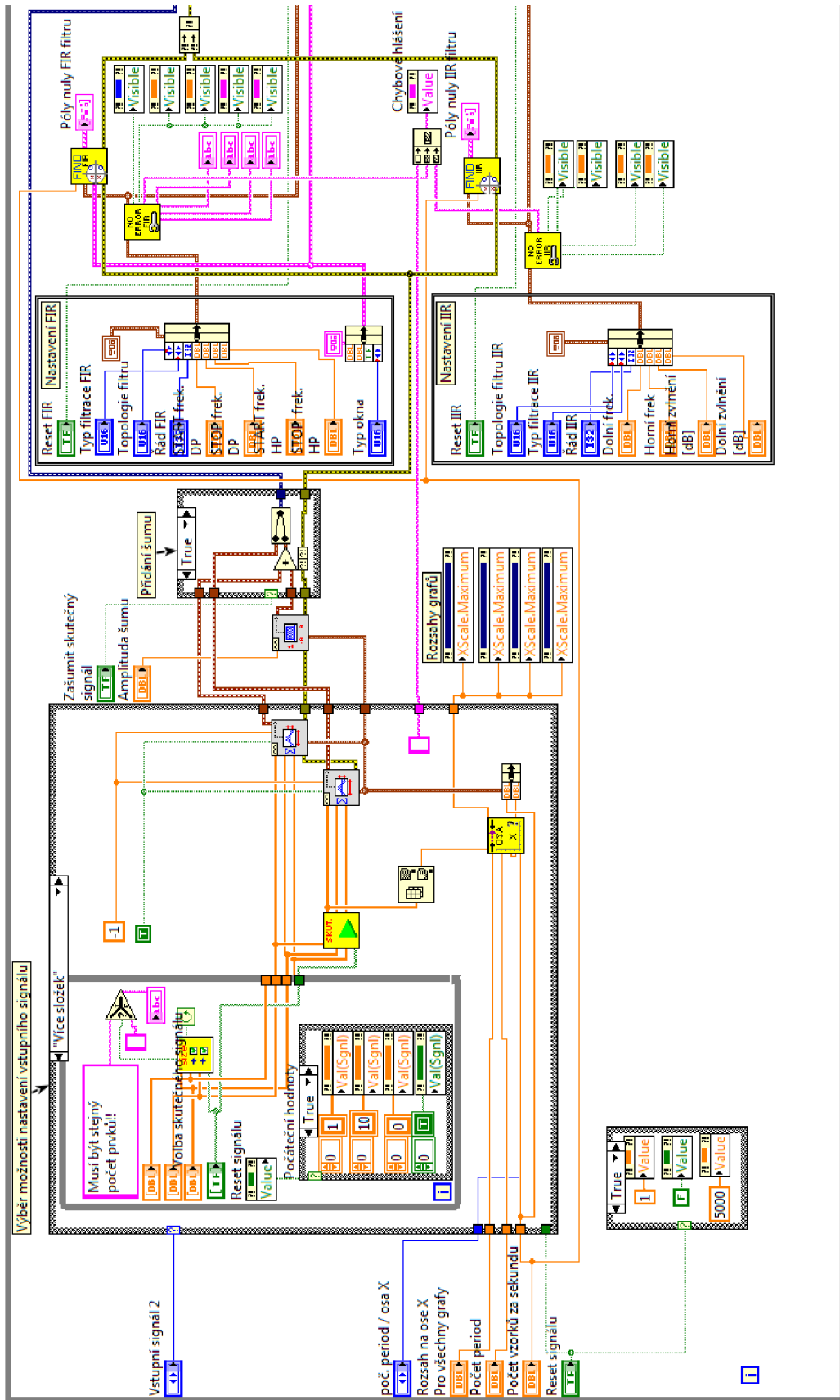
Póly nuly FIR filtru

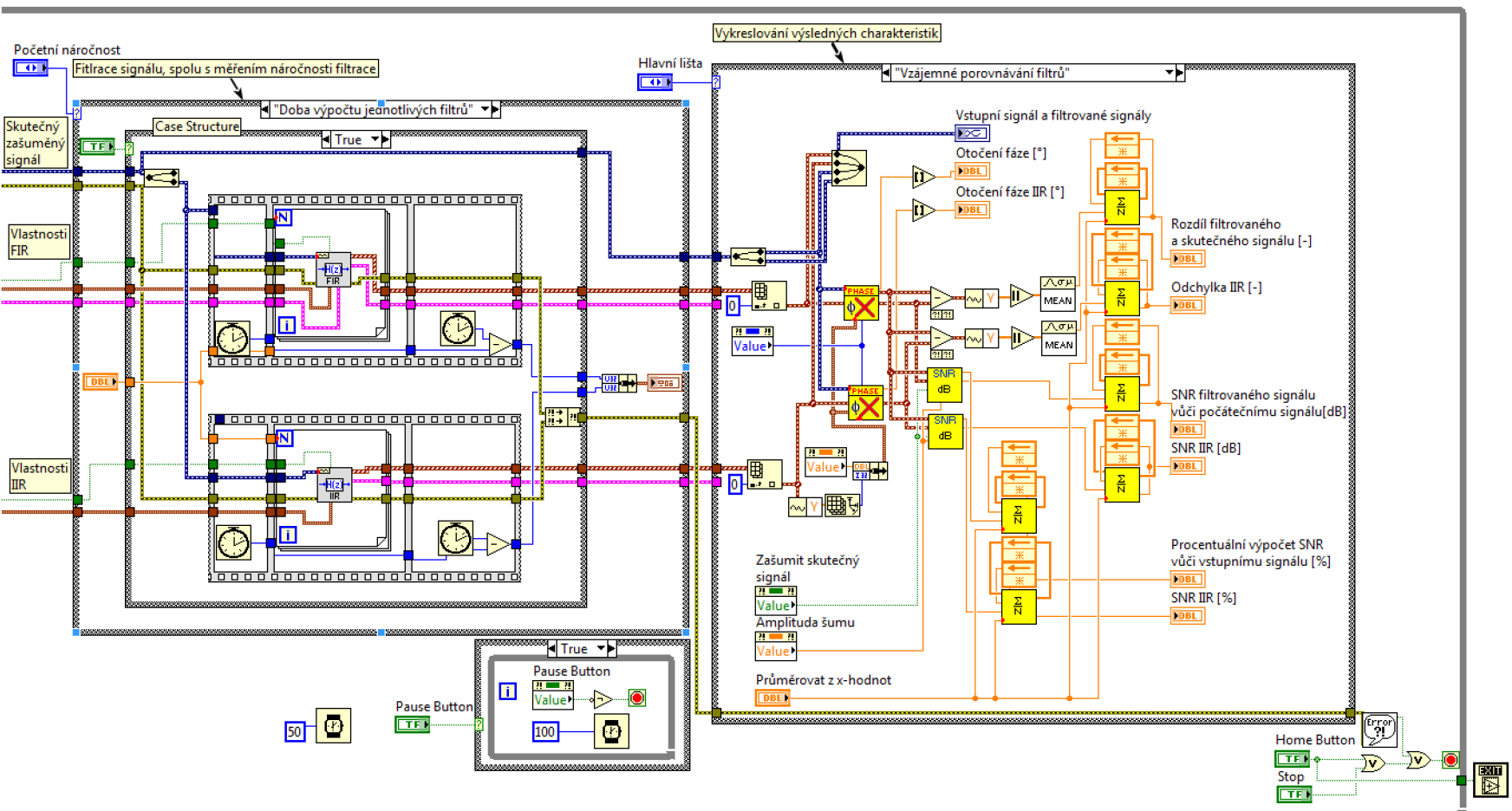
Kružnice | Nuly | Póly

Póly nuly IIR filtru

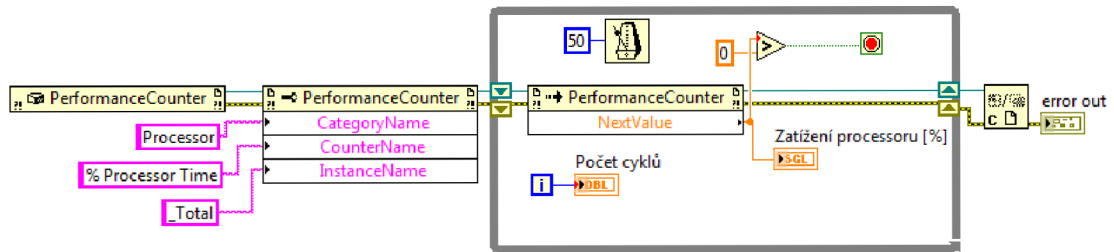


## Příloha B. První část blokového diagramu celého programu

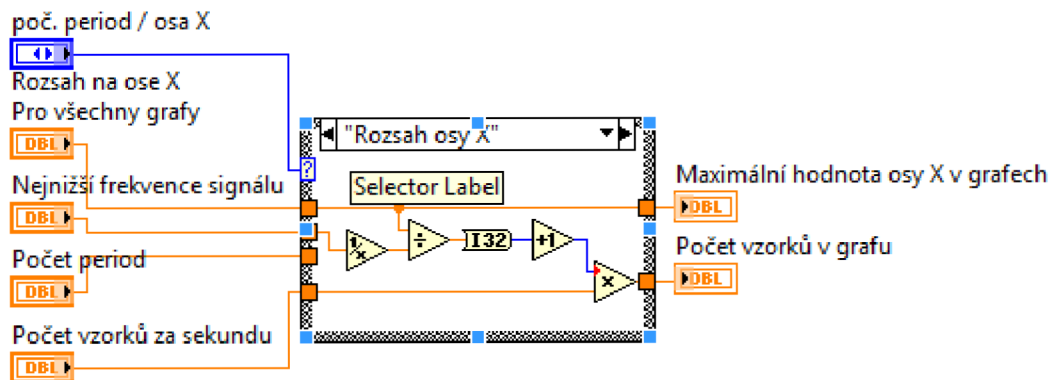




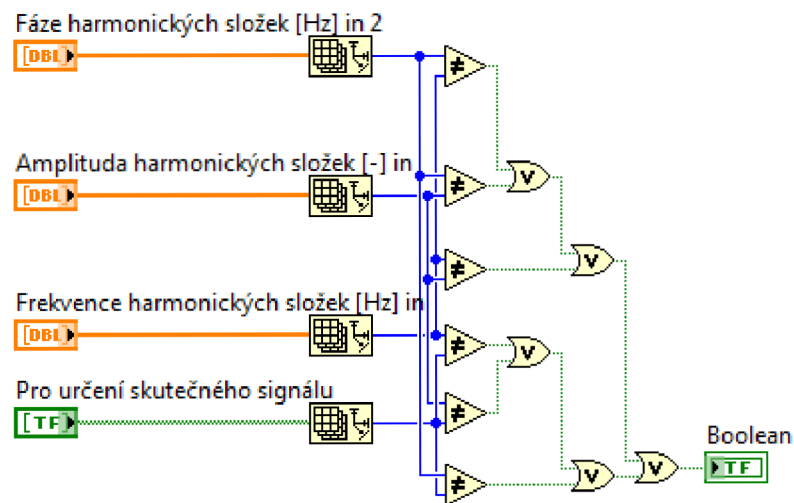
## Příloha D. Podprogram celkové využití CPU



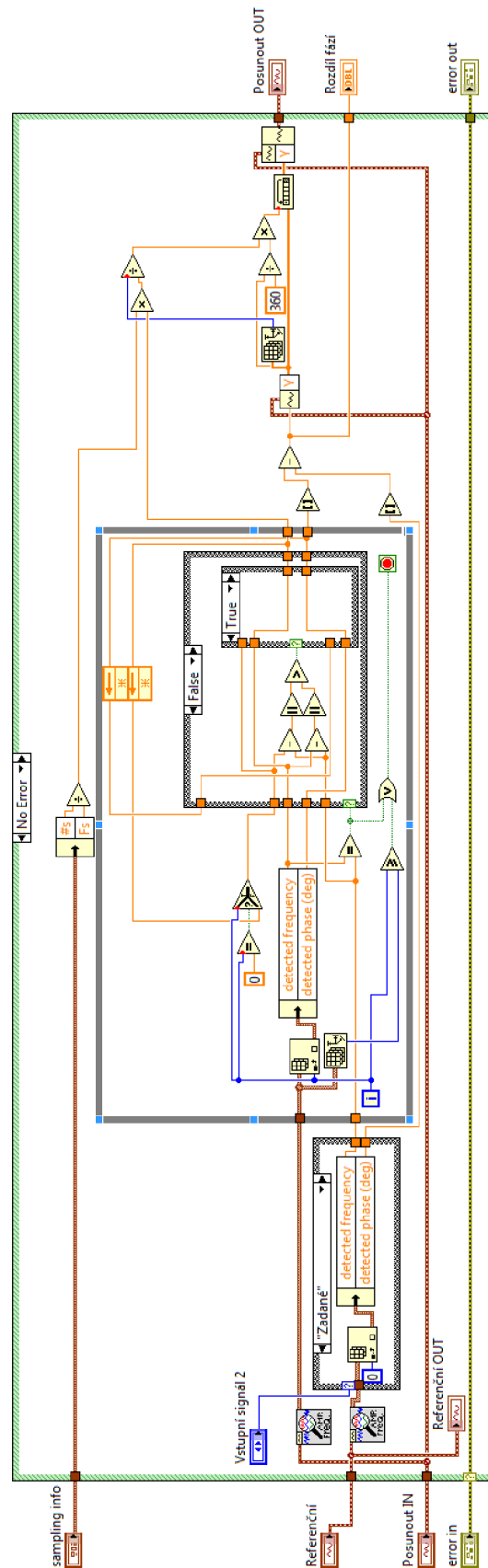
## Příloha E. Podprogram pro nastavení osy X



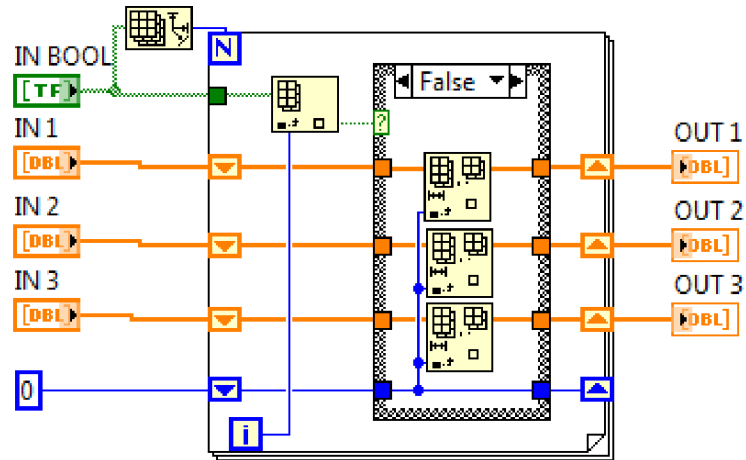
## Příloha F. Podprogram pro ošetření stejné velikosti všech polí



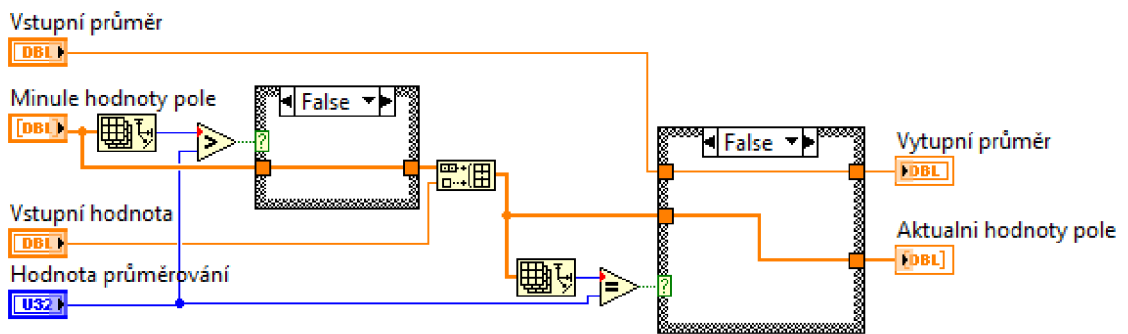
## Příloha G. Podprogram fázového posunutí



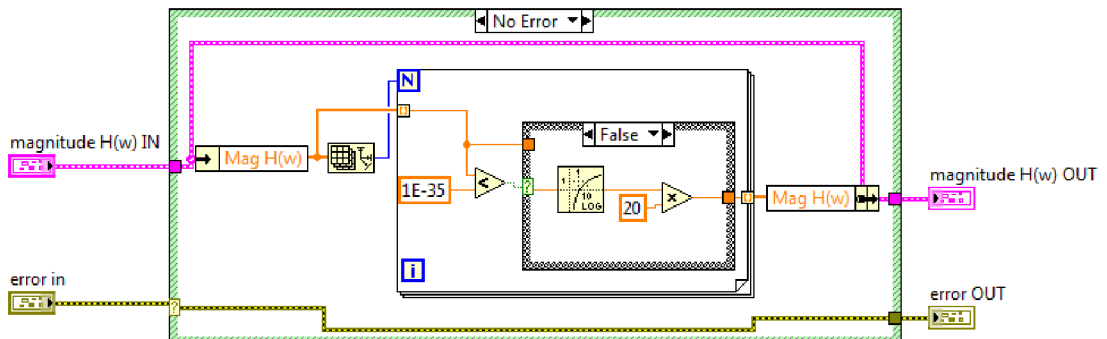
### Příloha H. Podprogram pro nastavení volby skutečného signálu



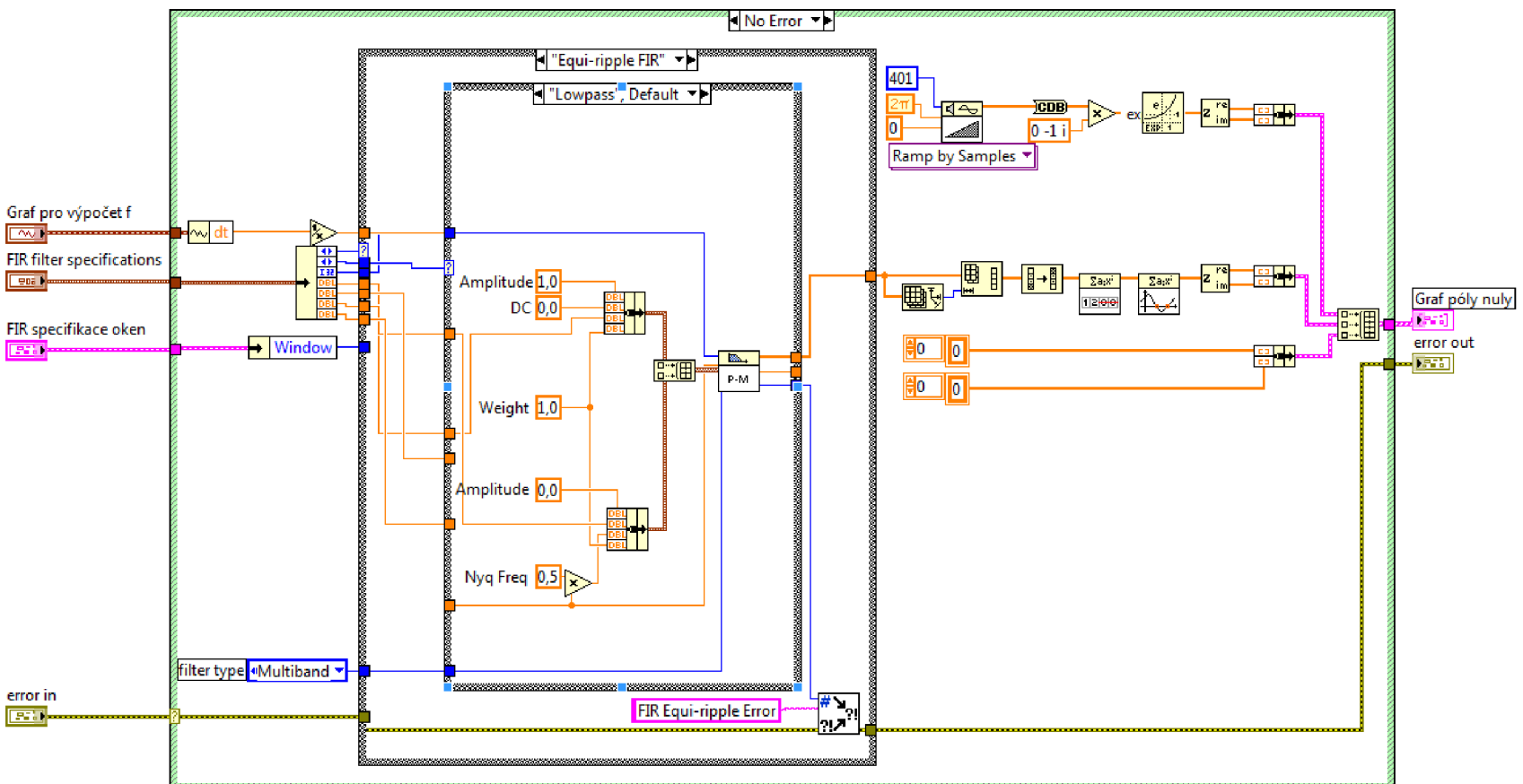
### Příloha I. Podprogram pro průměr z hodnot



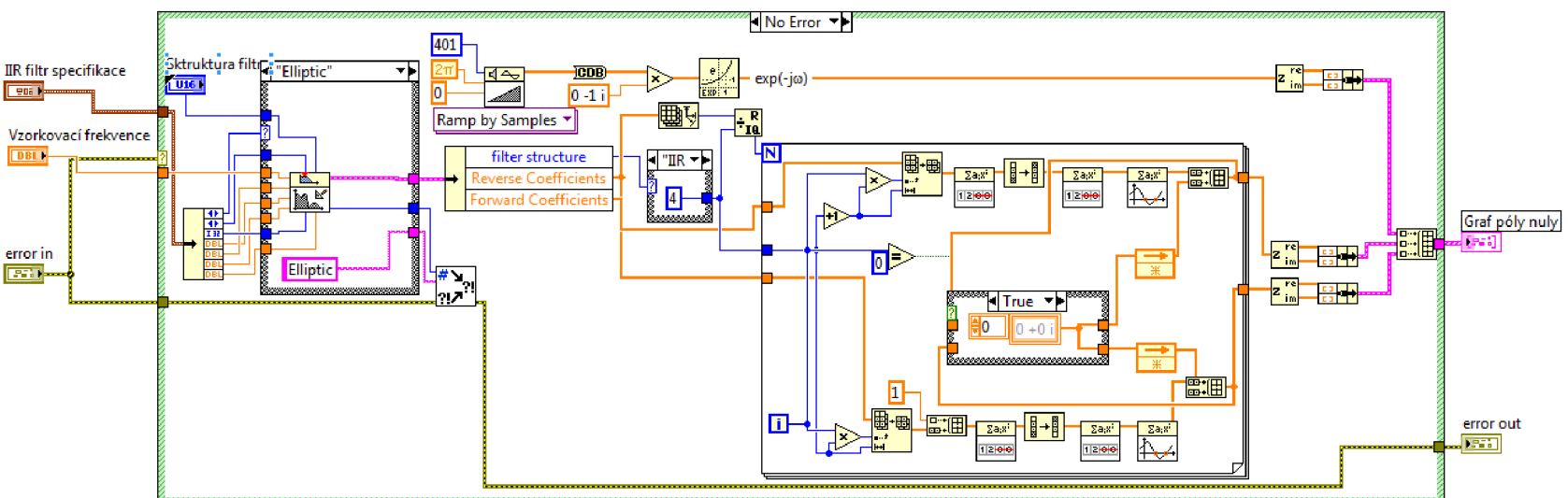
### Příloha J. Podprogram pro přepočet na dB



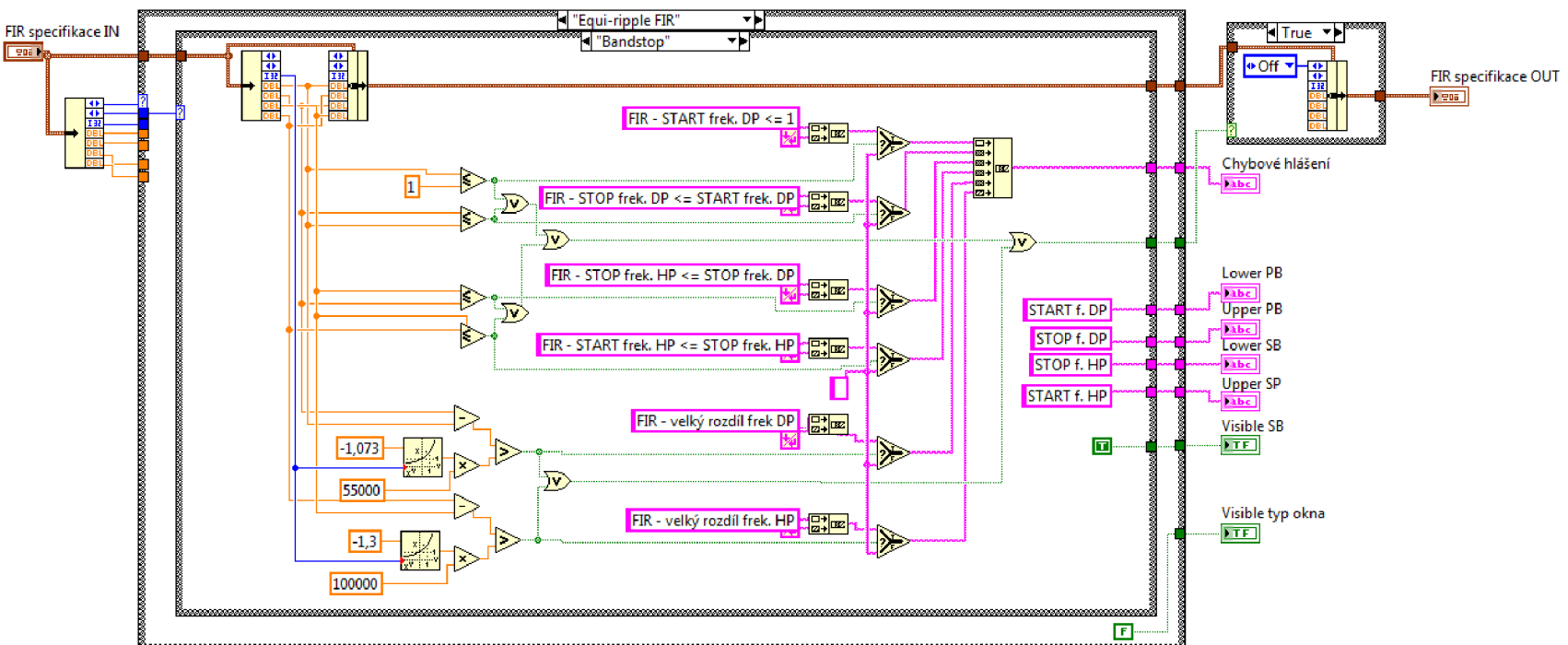
Příloha K. Podprogram pro vykreslení pólu a nul FIR



Příloha I. Podprogram pro vykreslení pólu a nul IIR

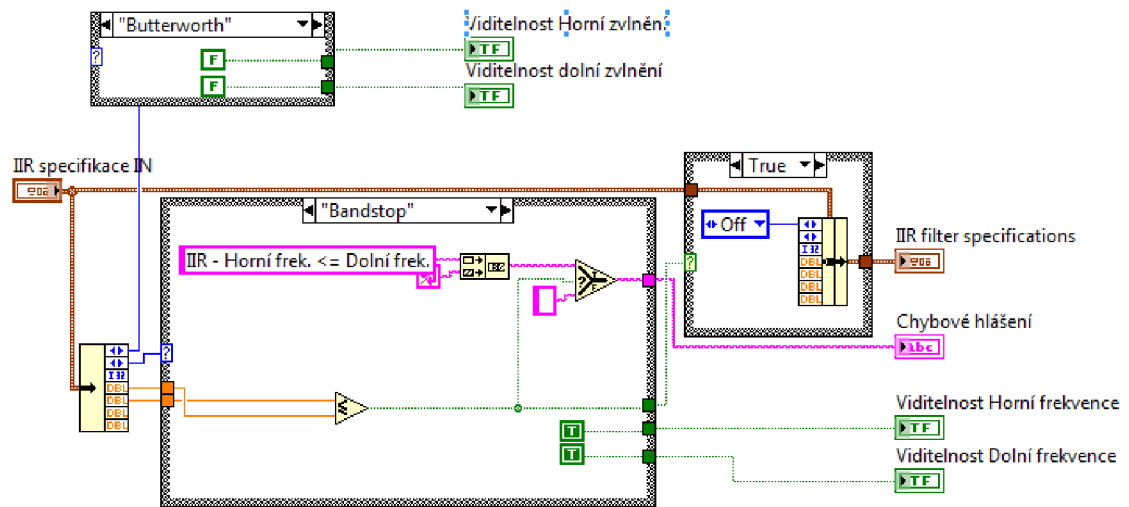


Příloha M. Podprogram pro ošetření chybného nastavení filtru FIR





## Příloha N. Podprogram pro ošetření chybného nastavení filtru IIR



## Příloha O. Podprogram pro výpočet SNR

