DIZAJN DIGITALNIH FILTERA POMOĆU ARDUINA

Đolonga, Antonio

Graduate thesis / Diplomski rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split / Sveučilište u Splitu**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:228:493543

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-12**

Repository / Repozitorij:

Repository of University Department of Professional Studies

zir.nsk.hr

UNIVERSITY OF SPLIT

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U SPLITU

SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Stručni diplomski studij Elektrotehnika

ANTONIO ĐOLONGA

Z A V R Š N I R A D

Dizajn digitalnih filtera pomoću Arduina

Split, rujan 2023.

SVEUČILIŠTE U SPLITU

SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Stručni diplomski studij Elektrotehnika

Predmet: Digitalna obrada signala

Z A V R Š N I R A D

Kandidat: Antonio Đolonga

Naslov rada: Dizajn digitalnih filtera pomoću Arduina

Mentor: Tonko Kovačević

Split, rujan 2023.

SADRŽAJ

SAŽETAK 1
1. UVOD 2
2. VREMENSKI SIGNALI 3
2.1. Uzorkovanje-kreiranje diskretnog signala 5
3. VRSTE FILTERA 6
4. IMPULSNI ODZIV 8
4.1. Konačni impulsni odziv 9
4.2. Beskonačni impulsni odziv 10
5. KONVOLUCIJA 13
6. FILTER POMIČNE SREDNJE VRIJEDNOSTI 16
7. FIR filteri 20
7.1. Nisko propusni FIR filter 27
7.2. Visoko propusni FIR filter 29
7.3. Pojasno nepropusni filter FIR filter 31
7.4. Pojasno propusni FIR filter 33
8. IIR FILTERI 35
8.1. Nisko propusni IIR filter 39
8.2. Visoko propusni IIR filter 42
8.3. Pojasno propusni IIR filter 45
8.4. Pojasno nepropusni IIR filter 49
8.5. Filtarske aproksimacije IIR filtera 53
8. ZAKLJUČAK 58
LITERATURA 59
POPIS SLIKA 60
POPIS TABLICA 62
PRILOZI 63
PRILOG 1. Kod za FIR filter pomoću prozora 63
PRILOG 2. Kod za IIR filter 67
PRILOG 3. Kod za unos vanjskih FIR koeficijenata 71

PRILOG 4. Kod za unos vanjskih IIR koeficijenata.....72
PRILOG 5. Kod za Filter pomične srednje vrijednosti.....74

SAŽETAK

Dizajn digitalnih filtera pomoću Arduina

Ovaj završni rad objašnjava teoriju kreiranja digitalnih FIR i IIR filtera i pomičnog filtra srednje vrijednosti, te uz njih prikazuje praktične primjere s brojevima koji su uvršteni u platformu Arduino. U radu je objašnjen proces nastajanja diskretnog signala, njegov impulsni odziv koji može biti konačan i beskonačan, postupak konvolucije koji je bitan za FIR filtere, te transformacija analognih filtera u digitalne IIR filtere.

Ključne riječi: **FIR, IIR, konvolucija, impulsni odziv**

SUMMARY

Design of digital filters using Arduino

This thesis explains the theory behind the creation of digital FIR and IIR filters and a moving average filter, along with practical examples with numbers included in the Arduino platform. The paper explains the process of creating a discrete signal, its impulse response which can be finite or infinite, the convolution process which is essential for FIR filters, and the transformation of analog filters into digital IIR filters.

Key words: **FIR, IIR, convolution, impulse response**

1. UVOD

Sam pojam digitalne obrade signala obuhvaća postupak kojim se određeni signal obrađuje na računalu ili nekoj razvojnoj platformi. Razlog zašto je nastala digitalna obrada signala je složenost obrade analognih signala, zbog toga što su analogni signali obično neprekidni, te je za njihovu obradu potrebno vršiti složene matematičke operacije.

Sve vrste obrade koje se mogu vršiti na analognom signalu mogu se vršiti i na digitalnom signalu, ali mnogo lakše jer se radi o diskretnom signalu, te većinu operacija obrade provodi samo računalo.

Ovaj rad se konkretno bavi filtriranjem signala na računalu odnosno dizajnom digitalnih filtera pomične srednje vrijednosti, FIR i IIR filtera. Razlog zašto se dizajniraju digitalni filteri, umjesto analognih je njihova jednostavnost izvedbe. Kod analognog filtriranja potrebno je imati komponente poput otpornika, kondenzatora, zavojnice, operacijskih pojačala, osciloskopa za provjeru ispravnosti strujnog kruga, dok je za digitalno filtriranje potrebno samo razviti algoritam(napisati program) u određenom programskom jeziku, te se učinak može vidjeti odmah.

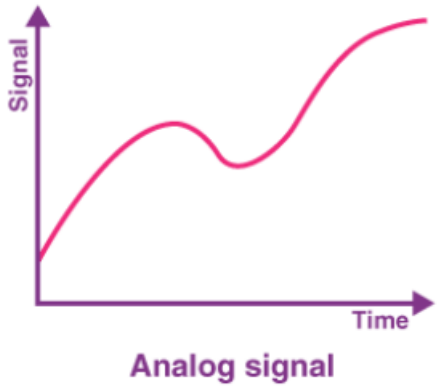
Analogna obrada signala također ima svoju programsku podršku i alate, ali simulacije izvedene na računalu je potrebno testirati u stvarnom okruženju s nizom gore navedenih komponenata.

Za potrebe ovog rada digitalno filtriranje se vršilo na platformi Arduino. Ona je odabrana zbog svoje iznime jednostavnosti i cijene, te zbog ugrađene funkcije prikazivanja(iscrtavanja) signala. Postupak obrade signala se mogao vršiti na bilo kojoj drugoj platformi te bi rezultat bio gotovo isti, samo potrebno je poznavati algoritme za obradu koji se kasnije implementiraju(isprogramiraju). Jedine razlike između platformi mogu biti njihove tehničke specifikacije ili ograničenja koja određena platforma ima, te dizajner digitalnih filtera sam bira platformu koja mu najviše odgovara. Arduino je ovdje simbolično odabran da se pokaže da najosnovniji algoritmi za filtriranje digitalnih filtera funkcioniraju.

2. VREMENSKI SIGNALI

Postoje dvije vrste vremenskih signala, a to su analogni i diskretni signal.

Analogni signal je signal u kojem svaki vremenski trenutak odgovara nekoj fizikalnoj veličini električnog napona ili struji. Na slici 2.1. vidimo prikaz najobičnijeg analognog signala.

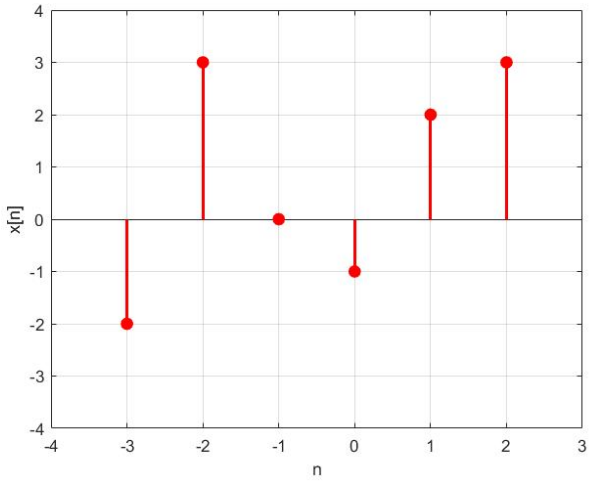


Slika 2.1 Analogni signal

Kontinuirani (analogni) signal se obično označava sa u(t) ili x(t) gdje je t nezavisna varijabla u vremenskoj domeni. Oznaka nezvisne varijable t u oblim zagradama ( ) obično definira da se radi o kontinuiranom signalu. Kod kontinuiranog signala varijabla t može poprimit bilo koju vrijednost realnog broja.

Za razliku od analognog signala kod kojeg je vrijednost poznata u svakom vremenskom trenutku, postoji diskretni signal kod kojeg je vrijednost poznata samo u **točno određenim diskretnim trenutcima.**

Diskretni signal označava se sa $u[n]$ ili $x[n]$ gdje je n nezavisna varijabla u vremenskoj domeni. Oznaka nezvisne varijable n u uglatim zagradama [ ] obično definira da se radi o diskretnom signalu. Kod diskretnog signala varijabla n može biti samo i jedino cijeli broj.
Na slici 2.2. (ispod) se vidi prikaz diskretnog signala:



Slika 2.2. Diskretni signal

Matematički zapis signala sa slike 2.2 je:

$$x[n] = -2\delta[n+3] + 3\delta[n+2] + 0\delta[n+1] - 1\delta[n+0] + 2\delta[n-1] + 3\delta[n-2] \quad (2.1)$$

Da bi se razumijeli dijelovi jednadžbe (2.1) i njeno značenje potrebno ju je analizirati po segmentima:

- Kod izraza $-2\delta[n+3]$ vrijednost signala iznosi -2 u trenutku $n = -3$.
- Kod izraza $+3\delta[n+2]$ vrijednost signala iznosi +3 u trenutku $n = -2$.
- Kod izraza $+0\delta[n+1]$ vrijednost signala iznosi 0 u trenutku $n = -1$.
- Kod izraza $-1\delta[n+0]$ vrijednost signala iznosi -1 u trenutku $n = 0$.
- Kod izraza $+2\delta[n-1]$ vrijednost signala iznosi 2 u trenutku $n = 1$.
- Kod izraza $+3\delta[n-2]$ vrijednost signala iznosi 3 u trenutku $n = 2$.

Sređivanjem jednadžbe(2.1) dobija se:

$$x[n] = -2\delta[n+3] + 3\delta[n+2] - 1\delta[n] + 2\delta[n-1] + 3\delta[n-2] \quad (2.2)$$

Diskretni signal se također može zapisati i pomoću diskretnog niza:

$$x[n] = \{-2, 3, 0, \underline{-1}, 2, 3\} \quad (2.3)$$

U nizu se vidi je broj -1 podcrtan što definira vrijednost u trenutku $n = 0$.

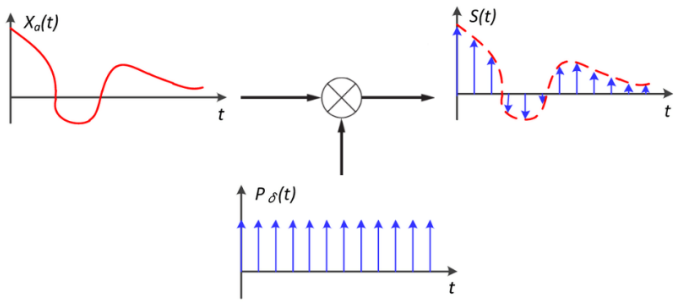
2.1. Uzorkovanje-kreiranje diskretnog signala

Uzorkovanje je postupak kojim se nezavisna varijabla, obično vrijeme, konvertira iz kontinuirane domene u diskretnu domenu.
Kontinuirani signal ima vrijednosti za sve točke u vremenu dok diskretni signal ima vrijednosti samo za određene trenutke.

Diskretni signal nastaje množenjem analognog signala $X_a(t)$ s nizom diracovih impulsa $\delta(t)$ koji su međusobno vremenski razmaknuti. Vremenski razmak definira svako koliko se uzima(uzorkuje) analogni signal. Poznavanjem vremenskog razmaka izmedu dva diracova impulsa $\delta(t)$ može se izračunati frekvencija kojom se uzimaju uzorci analognog signala, te se ta frekvencija zove frekvencija uzorkovanja(engl. *sampling frequency*).
Frekvencija uzorkovanja se računa prema izrazu (2.3):

$$f_s = \frac{1}{T_s} \tag{2.3}$$

Gdje je T_s vremenski razmak između dva diracova impulsa $\delta(t)$ impulsa, a f_s frekvencija uzorkovanja. Na slici 2.3. se nalazi analogni signal $X_a(t)$ pomnožen s nizom diracovih delta impulsa $\delta(t)$ (impulsni vlak) daje na svom izlazu signal $S(t)$. Može se vidjeti da signal $S(t)$ svojim oblikom ima sličnosti s analognim signalom $X_a(t)$.



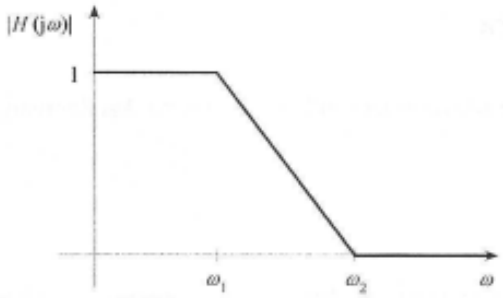
Slika 2.3. Nastajanje diskretnog signala

Da bi analogni signal ispravno uzorkovao mora biti zadovoljen sljedeći uvjet:

$$f_s \geq 2 \cdot f_{max}$$

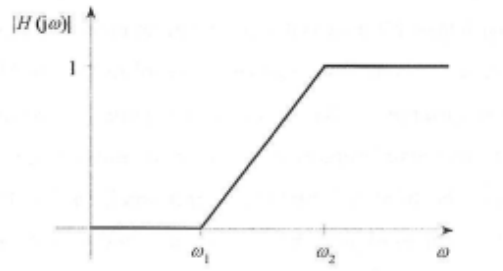
3. VRSTE FILTERA

Nisko propusni filtar je filtar koji propušta sve frekvencije koje su niže od granične frekvencije te prigušuje sve signale veće od granične frekvencije. Na slici 3.1. vidimo da filtar od $\omega = 0$ do neke ω_1 propušta sve frekvencije(propusni pojas), od ω_1 do ω_2 filtar prelazi iz propusnog pojasa u nepropusni pojas, te se to područje definira kao prijelazni pojas, a nakon frekvencije ω_2 signali se guše(nepropusni pojas).



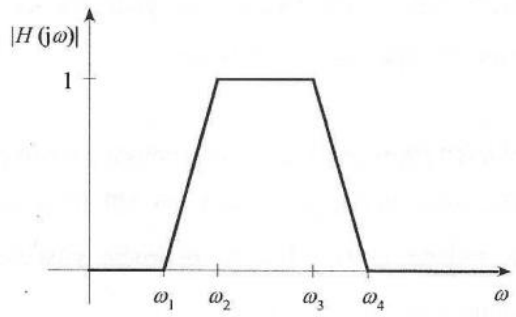
Slika 3.1. Frekvencijska karakteristika nisko propusnog filtra

Visoko propusni filtar je filtar koji blokira sve frekvencije koje su niže od granične frekvencije te propušta sve frekvencije veće od granične frekvencije. Na slici 3.2. vidimo da filtar od $\omega = 0$ do neke ω_1 blokira sve frekvencije(nepropusni pojas), od ω_1 do ω_2 filtar prelazi iz nepropusnog pojasa u propusni pojas, te se to područje definira kao prijelazni pojas, a nakon frekvencije ω_2 sve frekvencije se propuštaju (propusni pojas).



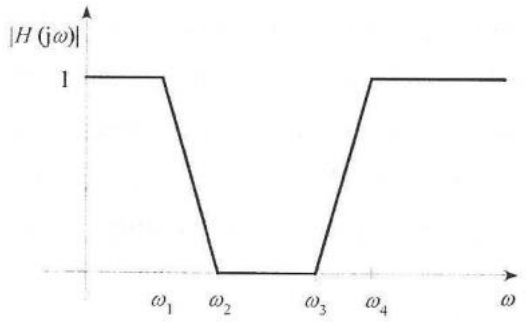
Slika 3.2. Frekvencijska karakteristika visoko propusnog filtra

Pojasno propusni filtar je filtar koji blokira sve frekvencije koje su niže od donje granične frekvencije i sve frekvencije koje su veće od gornje granične frekvencije, a propušta sve frekvencije koje su veće od donje granične frekvencije, a manje od gornje granične frekvencije. Na slici 3.3. vidimo da filtar blokira sve frekvencije(nepropusni pojas) od $\omega = 0$ do ω_1 te od ω_4 do $+\infty$, a od ω_1 do ω_2 te do ω_3 do ω_4 filtar se nalazi u prijelaznom pojasu. Sve frekvencije od ω_2 do ω_3 filtar propušta(propusni pojas).



Slika 3.3. Frekvencijska karakteristika pojasno propusnog filtra

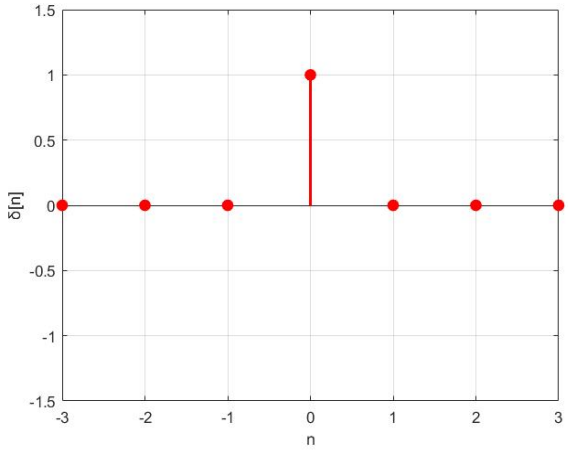
Pojasno nepropusni filtar je filtar koji propušta sve frekvencije koje su niže od donje granične frekvencije i sve frekvencije koje su veće od gornje granične frekvencije, a propušta sve frekvencije koje su veće od donje granične frekvencije, a manje od gornje granične frekvencije. Na slici 3.4. vidimo da filtar propušta sve frekvencije(propusni pojasa) od $\omega = 0$ do ω_1 te od ω_4 do $+\infty$, a od ω_1 do ω_2 te od ω_3 do ω_4 filtar se nalazi u prijelaznom pojasu. Sve frekvencije od ω_2 do ω_3 filtar prigušuje(nepropusni pojas).



Slika 3.4. Frekvencijska karakteristika pojasno nepropusnog filtra

4. IMPULSNI ODZIV

Delta funkcija $\delta[n]$ je normalizirani impuls, gdje uzorak u diskretnom trenutku $n = 0$ ima vrijednost 1 dok ostali uzorci imaju vrijednost 0. Delta funkcija se još zove i jedinični impuls. Na slici 4.1. se može vidjeti prikaz delta funkcije.



Slika 4.1. Delta funkcija $\delta[n]$

Za delta funkciju vrijedi $\delta[n]$:

$$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & \end{cases} \tag{4.1}$$

Dovede li se delta funkcija $\delta[n]$ na ulaz određenog sustava može se saznati impulsni odziv $h[n]$ nekog sustava. Dovođenjem delta funkcije(jediničnog impulsa**) δ[n] na ulaz nekog sustava umjesto klasičnog signala x[n] dobija se impulsi odziv h[n] na izlazu tog istog sustava umjesto y[n].** Postupak se može vidjeti na slici 4.2.

Impulsni odziv po svojoj definiciji može biti konačan i beskonačan.

Slika 4.2. Impulsni odziv sustava

4.1. Konačni impulsni odziv

Najbolji prikaz rada konačnog impulsnog odziva može se vidjeti riješavanjem jednostavnog primjera te primjenom prethodno uspostavljenih pravila.

Primjer 1:
Na temelju jednadžbe diferencija (4.2):

$$y[n] = x[n] + \frac{1}{2}x[n-1] - \frac{1}{4}x[n-3] - \frac{1}{4}x[n-5] \tag{4.2}$$

nađite impulsni odziv $h[n]$ diskretnog sustava.

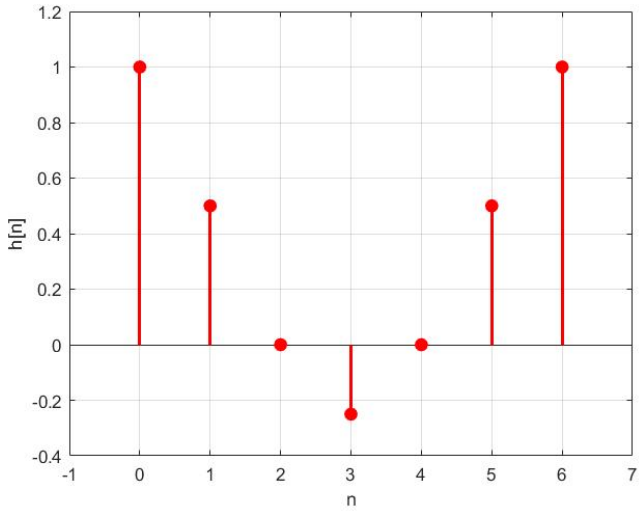
Rješenje:

Zamjenom x[n] sa $\delta[n]$ te zamjenom y[n] sa h[n] u jednadžbi diferencija(4.2) dobija se jednadžba za impulsni odziv(4.3):

$$h[n] = \delta[n] + \frac{1}{2}\delta[n-1] - \frac{1}{4}\delta[n-3] - \frac{1}{4}\delta[n-5] \tag{4.3}$$

Impulsni odziv može imati n impulsa čije će vrijednosti s vremenom poprimiti vrijednost 0.

Grafički prikaz sustava s konačnim impulsnim odzivom može se vidjeti na slici 4.3.



Slika 4.3. Prikaz konačnog impulsnog odziva

4.2. Beskonačni impulsni odziv

Kod beskonačnog impulsnog odziva sam odziv ne poprima vrijednost 0 nakon određenog broja n uzoraka nego istitrava u beskonaćnost. Razlog zašto impulsni odziv nikada ne pada na 0 su upravo prethodno izračunate vrijednosti izlaznog signala $y[n]$ koje se konstantno vraćaju kao ulazni signal kroz povratnu vezu.
Utjecaj prethodno izračunatih vrijednosti izlaznog signala se najbolje može vidjeti iz primjera koji slijedi.

Zadana je jednadžba diferencija (4.4):

$$y[n] - 0.6y[n-1] = x[n] \tag{4.4}$$

Riješavanjem jednadžbe (4.4). za $n = 0$ vrijedi:

$$y[n] = x[n] - 0.6y[n-1] \tag{4.5}$$

$$y[0] = x[0] + 0.6y[-1] = x[0] + 0 \tag{4.6}$$

$$y[0] = x[0] \tag{4.7}$$

Može se primjetiti da je izraz $0.6y[-1] = 0$, zbog toga što se pri računanju impulsnog odziva kod digitalnih filtera većinom radi o kauzalnim sustavima, zbog toga što računalni programi često ne mogu raditi s negativnim indeksima.

Daljnim raspisivanjem jednadžbe (4.5) i primjenom prethodno uspostavljenog odnosa $y[0] = x[0]$ slijedi za $n = 1$ da je:

$$y[1] = x[1] + 0.6y[0] = x[1] + 0.6x[0] \tag{4.8}$$

Na $y[1]$ utječe prethodni izlazni signal $y[0]$ za kojeg je ustanovljeno da je jednak $x[0]$.
Jednadžba diferencija (4.5) se nastavi ispisivati dalje za $n = 2$, te iz prethodne relacije vrijedi da je:

$$y[2] = x[2] + 0.6y[1] = x[2] + 0.6(x[1] + 0.6x[0]) \tag{4.9}$$

U jednadžbi (4.9) vidimo kako na izlazni signal y[2] u diskretnom trenutku $n = 2$ direktno utječe prethodni izlazni signal y[1].

Raspisivanjem jednadžbe (4.5) za $n = 3$ vrijedi:

$$y[3] = x[3] + 0.6y[2] = x[3] + 0.6\big(x[2] + 0.6(x[1] + 0.6x[0])\big) \tag{4.10}$$

$$y[3] = x[3] + 0.6(x[2] + 0.6x[1] + 0.6^2x[0]) \tag{4.11}$$

$$y[3] = x[3] + 0.6x[2] + 0.6^2x[1] + 0.6^3x[0] \tag{4.12}$$

Kod jednadžbe (4.10) za $n = 3$ vidimo kako na izlazni signal y[3] utječe y[2]. Jednadžba se tako može raspisivati u nedogled. Razlog zašto impulsni odziv nikada neće pasti na 0 je upravo utjecaj prethodnih izlaznih signala.

Standardiziranjem izraza (4.12) dolazi se do:

$$y[3] = \sum_{k=0}^{N-1} 0.6^k x[3-k] \tag{4.13}$$

Iz jednadžbe (4.13) se može zaključiti opći izraz:

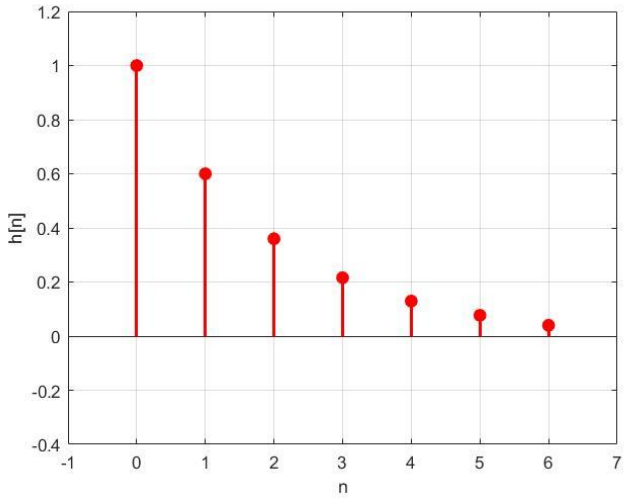
$$y[n] = \sum_{k=0}^{N-1} a^k x[n-k] \tag{4.14}$$

Gdje je konstanta $a = 0.6$, a u ovom slučaju promjenjivi $n = 3$.

Zamjenom $x[n]$ sa $\delta[n]$ i $y[n]$ sa $h[n]$ dobija se impulsni odziv $h[n]$.

$$h[n] = \sum_{k=0}^{N-1} a^k\, \delta[n-k] \tag{4.15}$$

Grafički prikaz impulnog odziva na slici 4.4. pokazuje da impulsni odziv teče u beskonačnost.



Slika 4.4. Beskonačni impulsni odziv

5. KONVOLUCIJA

Konvolucija je matematčki postupak kombiniranja dva signala u treći signal. Kod konvolucije se kombiniraju ulazni signal $x[n]$ i impulsni odziv sustava $h[n]$ s ciljem dobijanja trećeg signala $y[n]$. Ulazni signal $x[n]$ obično ulazi u određeni linearni sustav impulsnog odziva h[n] što rezultira izlaznim signalom y[n].

Zapis konvolucije je:

$$y[n] = x[n] * h[n] \tag{5.1}$$

Gdje znak * ne predstavlja matematičku operaciju množenja nego konvoluciju.

Konvolucija se može zapisati također u obliku:

$$y[n] = \sum_{k=0}^{N-1} x[k]h[n-k] \tag{5.2}$$

Razumijevanje konvolucije i njenih matematičkih izraza najbolje se može vidjeti na slijedećem primjeru.

1. Parametar n zamjeni se parametrom k kod $x[n]$ i $h[n]$, te oni postaju $x[k]$ i $h[k]$.
2. Napravi se zrcaljenje funkcije $h[k]$ te ona postaje $h[-k]$.
3. Vrši se pomak funkcije $h[-k]$ za određeni n u desno, te ona postaje $h[n-k]$.

Duljina izlaznog signala se definira kao zbroj duljine dvaju ulaznih signala $x[n]$ i $h[n]$, umanjenih za 1.

Nakon provođenja prethodnih koraka signali $x[k]$ i $h[n-k]$ se mogu zapisati u tablicu 5.1. gdje svaki kvadratić predstavlja vrijednost diskretnog signala. Prazni kvadratići definiraju da je vrijednost signala 0.

- Prvi redak sastoji se od diskretnih trenutaka, radi lakše orijentacije pri računanju.
- Drugi redak sastoji se od signala $x[k]$.
- Treći redak od signala $h[-k]$.
- Ostali redovi sastoje se od signala $h[n-k]$ koji se pomiće **u desno za jedan uzorak** sve dok se ne dosegne duljina izlaznog signala.

Za primjer se može uzeti da je:

$$x[n] = \{\underline{5}, 0, 2\} \tag{5.3}$$

$$h[n] = \{\underline{-2}, 4, 0, -3\} \tag{5.4}$$

Ako se primjene prethodno(gore) uspostavljena pravila ona se umjesto crtanja grafova mogu staviti u tablicu 5.1.

	k	-3	-2	-1	0	1	2	3	4	5	6
	$x[k]$				5	0	2				
$h[n-k]$	$h[-k]$	-3	0	4	-2						
	$h[1-k]$		-3	0	4	-2					
	$h[2-k]$			-3	0	4	-2				
	$h[3-k]$				-3	0	4	-2			
	$h[4-k]$					-3	0	4	-2		
	$h[5-k]$						-3	0	4	-2	

Tablica 5.1. Prikaz postupka konvolucije

Za izračun izlaznog signala $y[n]$ dovoljno je gledati signale u crveno obojanoj rešetci.

Za $n = 0$ vrijedi:

$y[0] = x[0] \cdot h[0] + x[1] \cdot h[0-1] + x[2] \cdot h[0-2]$

$y[0] = x[0] \cdot h[0] + x[1] \cdot h[-1] + x[2] \cdot h[-2]$

$y[0] = 5 \cdot (-2) + 0 \cdot 0 + 2 \cdot 0 = -10$

Za $n = 1$ vrijedi:

$y[1] = x[0] \cdot h[1-0] + x[1] \cdot h[1-1] + x[2] \cdot h[1-2]$

$y[1] = x[0] \cdot h[1] + x[1] \cdot h[0] + x[2] \cdot h[-1]$

$y[1] = 5 \cdot 4 + 0 \cdot (-2) + 2 \cdot 0 = 20$

Za $n = 2$ vrijedi:

$$y[2] = x[0] \cdot h[2-0] + x[1] \cdot h[2-1] + x[2] \cdot h[2-2]$$

$$y[2] = x[0] \cdot h[2] + x[1] \cdot h[1] + x[2] \cdot h[0]$$

$$y[2] = 5 \cdot 0 + 0 \cdot 4 + 2 \cdot (-2) = -4$$

Za $n = 3$ vrijedi:

$$y[3] = x[0] \cdot h[3-0] + x[1] \cdot h[3-1] + x[2] \cdot h[3-2]$$

$$y[3] = x[0] \cdot h[3] + x[1] \cdot h[2] + x[2] \cdot h[1]$$

$$y[3] = 5 \cdot (-3) + 0 \cdot 0 + 2 \cdot 4 = -7$$

Za $n = 4$ vrijedi:

$$y[4] = x[0] \cdot h[4-0] + x[1] \cdot h[4-1] + x[2] \cdot h[4-2]$$

$$y[4] = x[0] \cdot h[4] + x[1] \cdot h[3] + x[2] \cdot h[2]$$

$$y[4] = 5 \cdot 0 + 0 \cdot (-3) + 2 \cdot 0 = 0$$

Za $n = 5$ vrijedi:

$$y[5] = x[0] \cdot h[5-0] + x[1] \cdot h[5-1] + x[2] \cdot h[5-2]$$

$$y[5] = x[0] \cdot h[5] + x[1] \cdot h[4] + x[2] \cdot h[3]$$

$$y[5] = 5 \cdot 0 + 0 \cdot 0 + 2 \cdot (-3) = -6$$

Izlazni signal $y[n]$ se konačno može zapisati u obliku:

$$y[n] = \{\underline{-10}, 20, -4, -7, 0, -6\} \tag{5.5}$$

Duljina ulaznog signala $x[n]$ iznosi 3, dok duljina $h[n]$ iznosi 4. Njihov zbroj, koji iznosi 7, se umanji za 1 i kao što se vidi daje duljinu izlaznog signala $y[n]$ koja iznosi 6.

6. FILTER POMIČNE SREDNJE VRIJEDNOSTI

Filtar pomične srednje vrijednosti daje izlazni signal $y[n]$ kao srednju vrijednost N uzoraka ulaznog signala.

Jednadžba filtera pomične srednje vrijednosti je:

$$y[n] = \frac{1}{N}\sum_{j=0}^{N-1} x[n+j] \tag{6.1}$$

Gdje je:

$y[n] -$ izlazni filtrirani signal

$x[n] -$ ulazni signal

$N -$ broj ulaznih uzoraka

Ako je primjerice $N = 4$ onda za $n = 10$ jednadžba je:

$$y[10] = \frac{x[10] + x[11] + x[12] + x[13]}{4} \tag{6.2}$$

Filtar se također može zapisati i u rekurzijskom obliku:

Pogledajmo niz:

$$y[40] = x[37] + x[38] + x[39] + x[40] + x[41] + x[42]x + x[43] \tag{6.3}$$

$$y[41] = x[38] + x[39] + x[40] + x[41] + x[42] + x[43]x + x[44] \tag{6.4}$$

Može se vidjeti da se točke od $x[38]$ do $x[43]$ moraju zbrojiti dva puta, za $y[40]$ i $y[41]$.

Sada se jednadžba (6.3) i (6.4) može napisati u obliku:

$$y[41] = y[40] + x[44] - x[37] \tag{6.5}$$

Na temelju jednadžbe (6.5) može se zapisati opći oblik:

$$y[n] = y[n-1] + x[n+p] - x[n-q] \tag{6.6}$$

Te vrijedi:

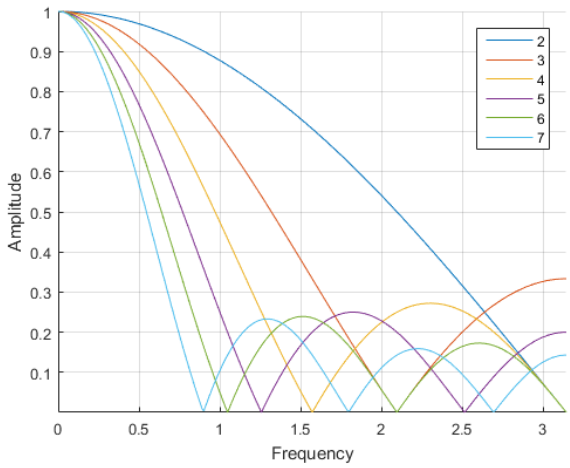
$$p = \frac{N-1}{2} \tag{6.7}$$

$$q = p + 1 \tag{6.8}$$

Filtar pomične srednje vrijednosti je najlakši filtar po definiciji za shvaćanje, što se može vidjeti iz gore navedenih formula. Filtar služi za glađanje signala i reduciranje šuma.

Filtar pomične srednje vrijednosti se ponaša kao nisko propusni filtar, ali iznimno loših karakteristika. Gušenje u nepropusnom pojasu mu je iznimno loše i prijelazni pojas je iznimno širok. Filtar je dobar za signale kodiranje u vremenskom području, dok posjeduje iznimno loše performanse za odvajanje različitih frekvencijskih pojaseva.

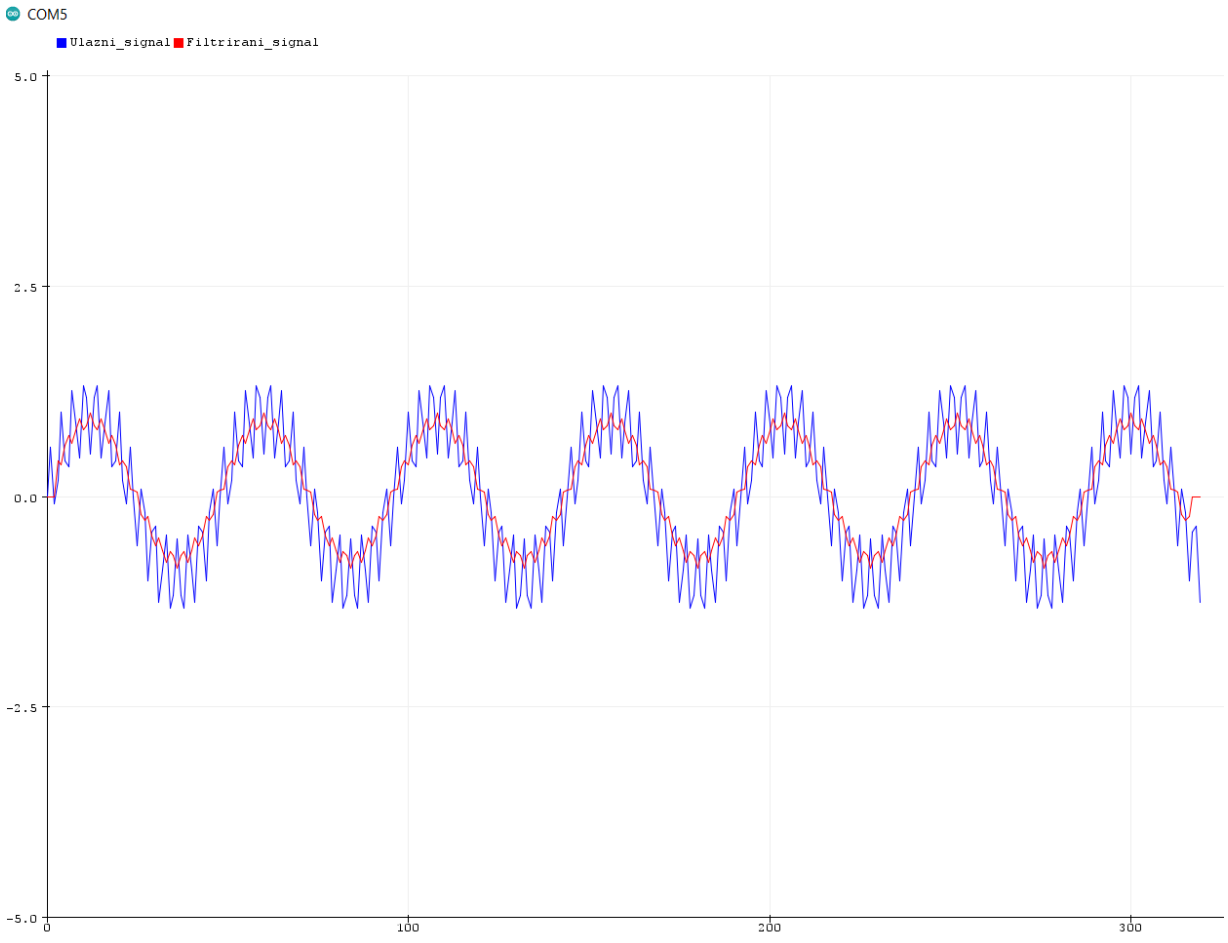
Povećanjem N, odnosno odabirom većeg reda poboljšavaju se performanse filtera što se može vidjeti na slici 6.1.



Slika 6.1. Frekvencijski odziv filtra pomične srednje vrijednosti

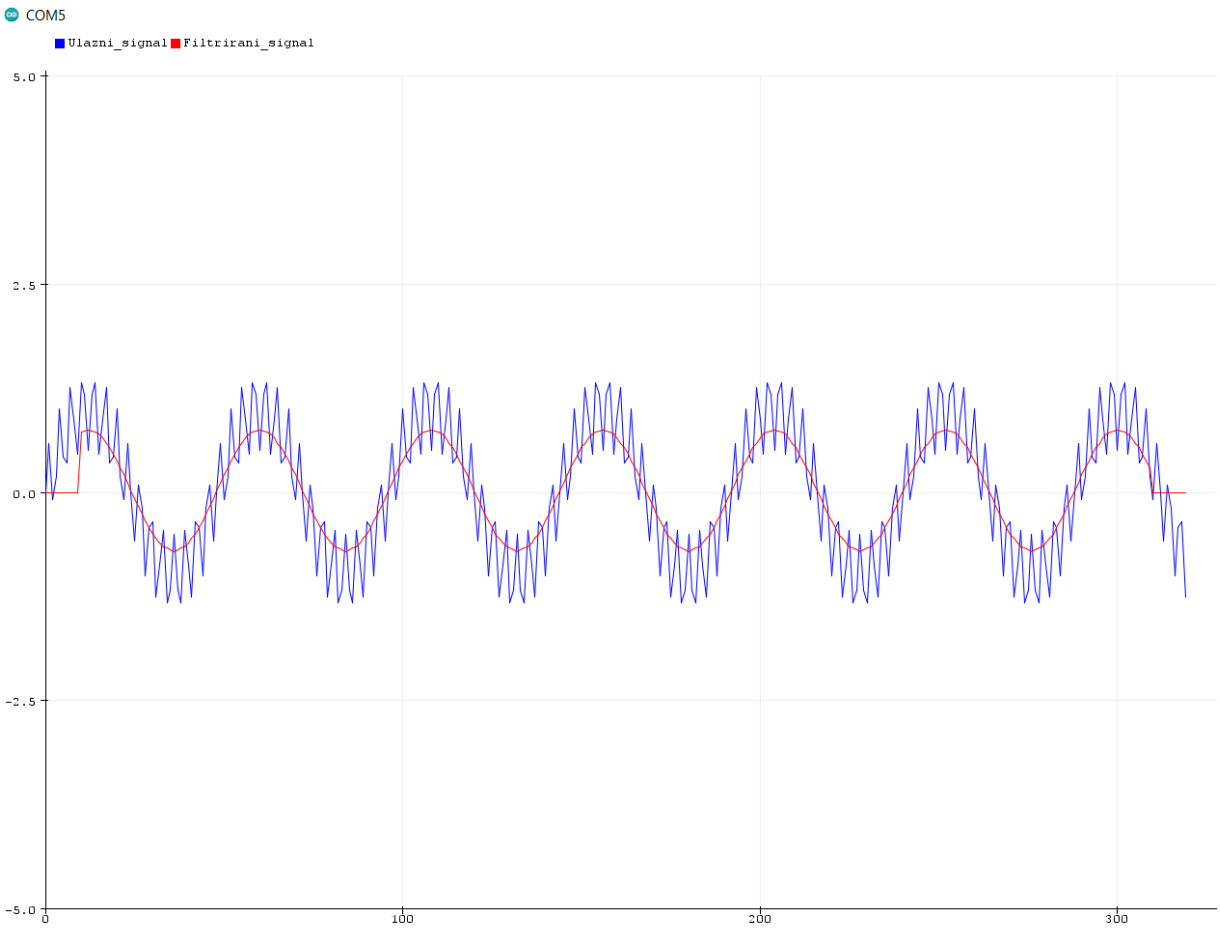
Na slici 6.2. se može vidjeti izlazni signal gdje je $N = 6$.

Da se primjetiti određeno valovanje u izlaznom filtriranom signalu.



Slika 6.2. Izlazni signal pri $N = 6$

Na slici 6.3. se može vidjeti izlazni signal za $N = 20$. Valovanje je gotovo nestalo u odnosu na prethodni filter gdje je $N = 6$.



Slika 6.3. Izlazni signal pri $N = 20$

7. FIR filteri

FIR filteri su filteri konačnog impulsnog odziva čiji je odziv konačnog trajanja, odnosno vrijednost impulsnog odziva padne na 0 u određenom trenutku n.

Izlazni signal $y[n]$ kod FIR filtera se dobije tako da se ulazni signal $x[n]$ konvoluira s impulsnim odzivom $h[n]$:

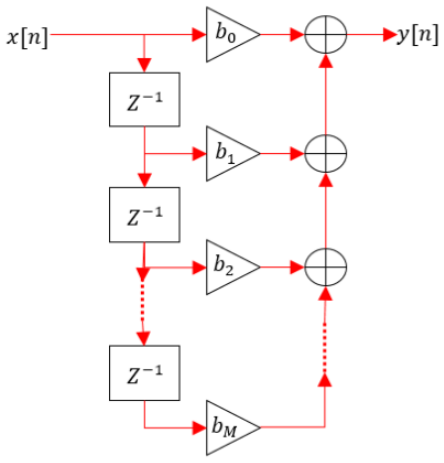
$$y[n] = \sum_{n=0}^{M-1} x[n] * h[n] \tag{7.1}$$

Odnosno kada se prethodna jednadžba(7.1) sredi i raspiše kao što je to objašnjeno u poglavlju 5. KONVOLUCIJA onda vrijedi:

$$y[n] = \sum_{k=0}^{M-1} x[k]h[n-k] = \sum_{k=0}^{M-1} b_k x[n-k] \tag{7.2}$$

Oznaka b_k se još naziva koeficijent ulaznih vrijednosti(engl. *feedforward coefficients*).
Kod FIR filtera impulsni odziv $h[n]$ se još naziva **filter kernel ili jezgra filtera**.
Ako se žele dobiti željene karakteristike FIR filtera potrebno je ispravno kreirati kernel filtera.

Blok shema FIR filtera se nalazi na slici 7.1.:

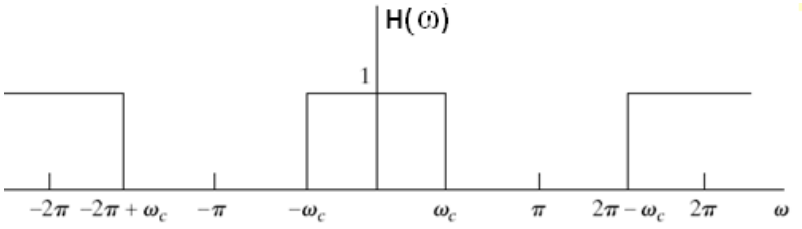


Slika 7.1. Blok shema FIR filtera

Na blok shemi 7.1. ,a i u prethodnoj jednadžbi (7.2) se vidi da FIR filter koristi samo trenutne i prethodne vrijednosti ulaznog signala x[n], te nema nikakvu povratnu vezu koja je povezana sa izlaznim signalom y[n]. Manjak povratne veze s izlazom čini filter stabilnim, odnosno ne može se dogoditi da u toku rada filter postane nestabilan.

Za dizajn FIR filtera potrebno je naći inverznu diskretnu Fourierovu transformaciju (IDFT) frekvencijske karakteristike nisko propusnog filtra.

Idealni nisko propusni filter je filter gdje sve frekvencije veće od definirane granične frekvencije ω_c budu eliminirane, dok sve frekvencije manje od ω_c budu propuštene. Frekvencijska karakteristika nisko propusnog filtera se može vidjeti na slici 7.2.



Slika 7.2. Idealna nekauzalna frekvencijska karakteristika nisko propusnog filtra

Pošto se postupak konvolucije ulaznog signala $x[n]$ i impulsnog odziva(filter kernela) $h[n]$ vrši u diskretnom vremenu n, a ne frekvencijskoj domeni potrebno je naći prijenosnu funkciju nisko propusnog filtera u diskretnom vremenu n.

Postupak pronalaska se vrši pomoći inverzne diskretne furijerove transformacije (IDFT):

$$x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X\left(e^{j\omega}\right) e^{j\omega n} d\omega \tag{7.3}$$

$$h_{sinc}[n] = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} 1 \cdot e^{j\omega n} d\omega \tag{7.4}$$

$$h_{sinc}[n] = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} e^{j\omega n}\, d\omega \tag{7.5}$$

$$h_{sinc}[n] = \frac{1}{2\pi}\frac{e^{j\omega n}}{jn}\Bigg|_{-\omega_c}^{\omega_c} \tag{7.6}$$

$$h_{sinc}[n] = \frac{1}{2\pi jn}\left[e^{j\omega_c n} - e^{-j\omega_c n}\right] \tag{7.7}$$

Primjenom trigonometrijskog identita $sin(\omega n) = \frac{1}{2j}(e^{j\omega n} - e^{-j\omega n})$ na izraz (7.7):

Dobija se:

$$h_{sinc}[n] = \frac{\sin(\omega_c n)}{\pi n} \tag{7.8}$$

Dodavanjem ω_c u brojnik i nazivnik:

$$h_{sinc}[n] = \frac{\omega_c}{\omega_c}\frac{\sin(\omega_c n)}{\pi n} = \frac{\omega_c}{\pi}\frac{\sin(\omega_c n)}{\omega_c n} \tag{7.9}$$

Dobija se sinc funkcija:

$$h_{sinc}[n] = \frac{\omega_c}{\pi} \cdot sinc(\omega_c n) \tag{7.10}$$

Raspisivanjem $\omega_c = 2\,\pi f$ u brojniku i skraćivanjem s π konačno se dobija funkcija povoljna za korištenje:

$$h_{sinc}[n] = 2f \cdot sinc(\omega_c n) \tag{7.11}$$

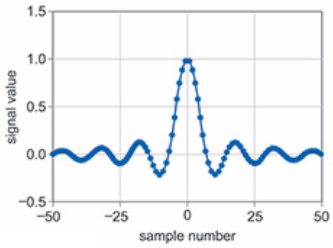
Gdje je:

$$f = \frac{f_c}{f_s} \tag{7.12}$$

f_c - granična frekvencija

f_s – frekvencija uzorkovanja

$Sinc$ funkcija izgleda kao na slici 7.3:



Slika 7.3. Prikaz diskretne sinc funkcije

Funkcija $sinc$ ima dva problema:

1. Njene vrijednosti u nijednom trenutku ne padaju na 0 nego istitravaju u beskonačnost.

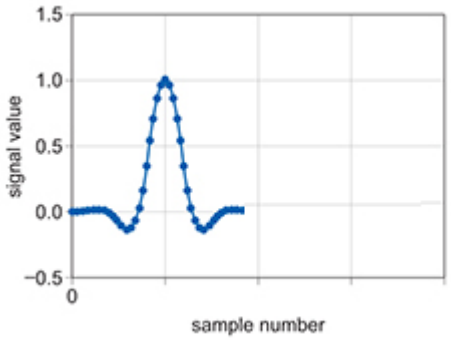
2. Funkcija se proteže od $-\infty$ do $+\infty$. Ovo je problem jer funkcija mora imati konačni broj uzoraka zbog toga jer sam impulsni odziv mora imati konačan broj uzoraka te njen početak treba biti u $n = 0$.

Da bi se prvi problem riješio odabire se određeni broj M koji predstavlja broj uzoraka oko glavne latice te se ostale točke skrate na 0.

Na ovakav način se dobija takozvana **skraćena $\boldsymbol{sinc}$ funkcija** (engl. *truncated sinc function*).

Da bi se drugi problem riješio moraju se svi uzorci pomaknuti u desno za M/2 zato da bi svi indeksi n bili pozitivni, odnosno da bi se kretali od 0 do M zbog toga što određeni programski jezici(npr. C++) ne mogu raditi s negativnim indeksima, a i sam rad s negativnim indeksima može biti problematičan.

Na slici 7.4. se može vidjeti skraćena sinc funkcija koja je pomaknuta za M/2.

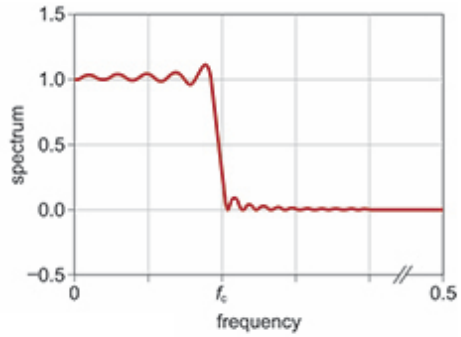


Slika 7.4. Skraćena sinc funkcija

Kada se sinc funkcija pomakne za $\frac{M}{2}$ dobije se:

$$h_{sinc}[n] = \frac{\sin\left(\omega\left(n - \frac{M}{2}\right)\right)}{\pi\left(n - \frac{M}{2}\right)} \tag{7.13}$$

Ovako nastali filter kernel i dalje nije idealan za konvoluciju s ulaznim signalom zbog toga što u propusnom pojasu ima veliko valovanje, a u nepropusnom pojasu ima lošije slabljenje što se može vidjeti na slici 7.5.



Slika 7.5. Frekvencijska karakteristika sinc funkcije

Da bi se valovanje u propusnom pojasu i loše slabljenje u nepropusnom pojasu riješili skraćena sinc funckija se može pomnožiti s jednim od dva prozora, Blackman ili Hamming. Množenjem s jednim od tih prozora se znatno smanjuje se valovanje u propusnom području i poboljšava se slabljenje u nepropusnom području.

Jednadžba za Hammmingov prozor je:

$$w[n] = 0.54 - 0.46cos\frac{2\pi n}{M} \tag{7.14}$$

Jednadžba za Blackmanov prozor je:

$$w[n] = 0.42 - 0.5cos\left(\frac{2\pi n}{M}\right) + 0.08cos\left(\frac{4\pi n}{M}\right) \tag{7.15}$$

Množenjem $h_{sinc}[n]$ funkcije sa funkcijom prozora $w[n]$ dobija se filter kernel $h[n]$, odnosno impulsni odziv filtra $h[n]$ što se može vidjeti na jednadžbi (7.16).

$$h[n] = h_{sinc}[n] \cdot w[n] \tag{7.16}$$

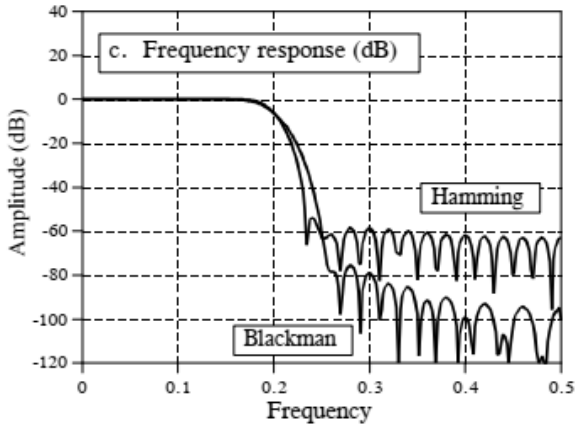
Filter kernel sa Hammingovim prozorom sada izgleda:

$$h[n] = \frac{\sin\left(\omega\left(n - \frac{M}{2}\right)\right)}{\pi\left(n - \frac{M}{2}\right)} \cdot \left(0.54 - 0.46cos\left(\frac{2\pi n}{M}\right)\right) \tag{7.17}$$

Filter kernel sa Blackmanovim prozorom sada izgleda:

$$h[n] = \frac{\sin\left(\omega\left(n - \frac{M}{2}\right)\right)}{\pi\left(n - \frac{M}{2}\right)} \cdot \left(0.42 - 0.5cos\left(\frac{2\pi n}{M}\right) + 0.08cos\left(\frac{4\pi n}{M}\right)\right) \tag{7.18}$$

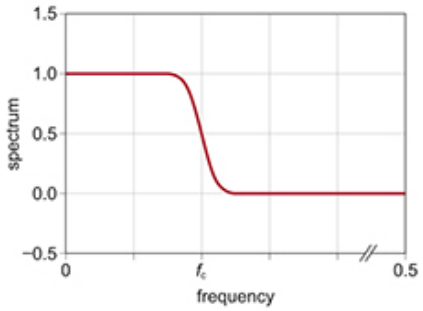
Hammingov i Blackmanov prozor imaju različite frekvencijske karakteristike, no obe daju iznimno dobro gušenje u nepropusnom pojasu. Hammingov prozor ima uži prijelazni pojasa, ali slabije gušenje u nepropusnom pojasu, dok Blackmanov prozor ima širi prijelazni pojas, ali bolje gušenje u nepropusnom pojasu. Njihove frekvencijske karakteristike se mogu vidjeti na slici 7.6.



Slika 7.6. Frekvencijske arakteristike Blackmanovog i Hammingovog prozora

Na testiranim signalima primjena oba prozora ne pokazuje nikakvu vidljivu razliku, te je za potrebe realizacije filtera korišten Blackmanov prozor.

Frekvencijska karakteristika filter kernela na slici 7.7. nakon množenja h_{sinc} s prozorom $w[n]$ pokazuje da je valovanje u propusnom pojasu smanjeno i da je poboljšano slabljenje u nepropusnom području.



Slika 7.7. Frekvencijska karakteristika filter kernela nakon množenja s prozorom $w[n]$

Kada se kreirao filter kernel, potrebno je, kao i u svemu treba odrediti njegovu duljinu.

Duljina filter kernela h[n] se odabire prema izrazu:

$$L = \frac{4}{BW} \tag{7.19}$$

Gdje BW predstavlja **normaliziranu** širinu prijelaznog pojasa.
Odabere li se uzak BW duljina kernela će biti veća, ali će računalo trebati provesti više računskih operacija. Što je BW manji(uži) filter se time približava idealnoj karakteristici, ali se time povećava duljina kernela. Valja paziti da duljina kernela ne prijeđe duljinu ulaznog signala zato jer mogu nastati problemi s filtriranjem signala točnije unosi se takozvano grupno kašenje. Što je dulji filter kernel, to je veće grupno kašenjenje.

BW se računa prema izrazu:

$$BW = \frac{B}{fs} \tag{7.20}$$

B predstavlja širinu pojasa u Hz-ima, dok je fs frekvencija uzorkovanja u Hz-ima.

Primjerice, ako želimo izračunati širinu prijelaznog pojasa kod nisko propusnog filtera od 1200 Hz i frekvencija uzorkovanja je 48000 Hz, onda je prema jednadžbi:

$$BW = \frac{1200\ Hz}{48000\ Hz} = 0.025 \tag{7.21}$$

Primjenom izraza 7.19. dobijamo duljinu kernela:

$$L = \frac{4}{BW} = \frac{4}{0.025} = 160 \tag{7.22}$$

7.1. Nisko propusni FIR filter

Prethodno je ustanovljena prijenosna funkcija nisko propusnog filtra h_{sinc} te prozorske jednazdžbe $w[n]$. Njihovim množenjem se dobija filter kernel(impulsni odziv) $h[n]$ nisko propusnog filtra.

Konvolucijom ulaznog signala $x[n]$ i filter kernela(impulsnog odziva) $h[n]$ dobija se filtrirani izlazni signal $y[n]$.

Za kreiranje nisko propusnog filtra potrebno je samo izračunati normaliziranu graničnu frekvenciju ω_c i duljinu filter kernela(impulsnog odziva) $h[n]$.

Ako je granična frekvencija nisko propusnog filtra $f_c = 3000\ Hz$ onda se njena normalizirana frekvencija računa prema:

$$\omega_c = \frac{2\pi f_c}{f_s} \tag{7.23}$$

$$\omega_c = \frac{2 \cdot \pi \cdot 3000}{48000} = \frac{6000 \cdot \pi}{48000} \tag{7.24}$$

$$\omega_c = 0.125\pi\ \frac{rad}{uzorak}$$

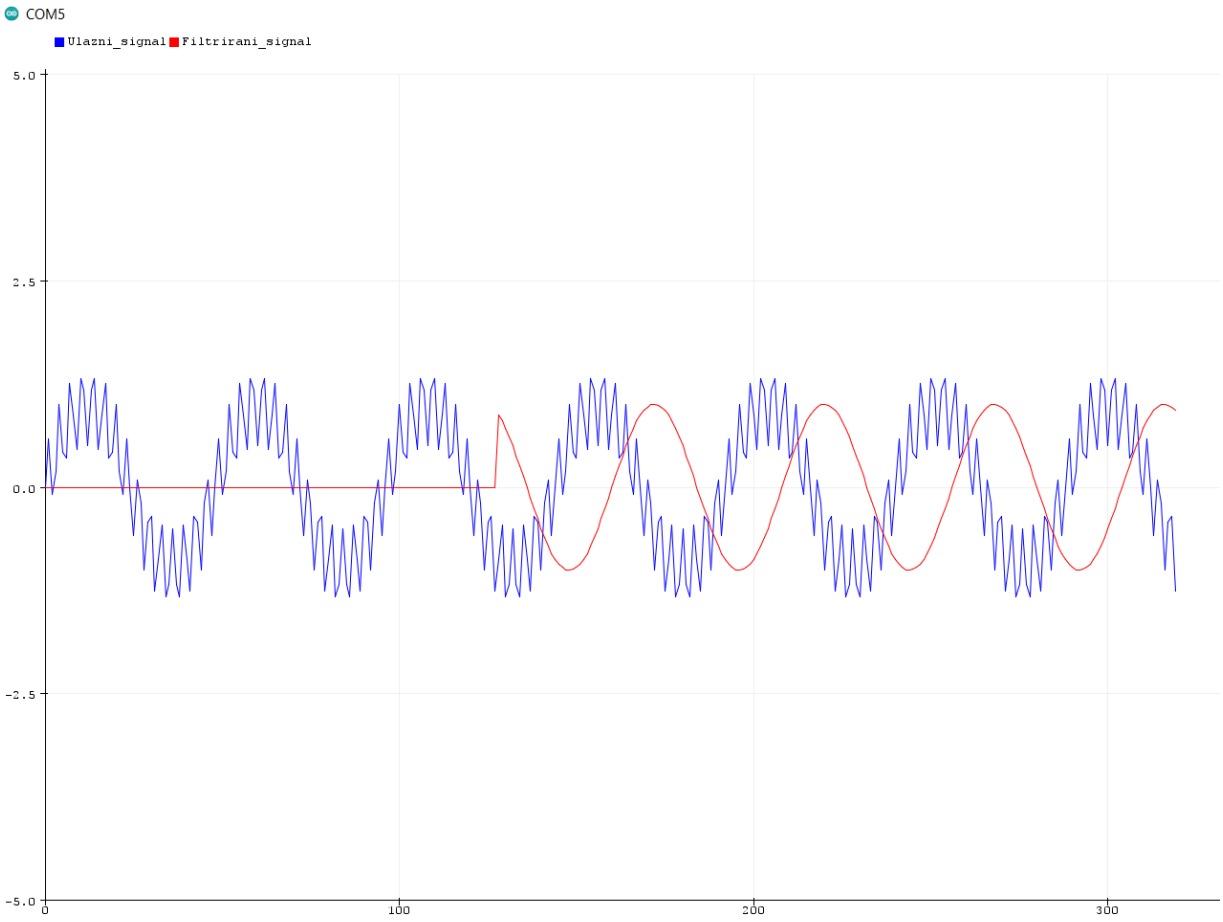
Ako želimo širinu prijelaznog pojasa od 1500 Hz onda nju računamo prema:

$$BW = \frac{B}{f_s} = \frac{1500}{48000} = 0.03125 \tag{7.25}$$

Te je dužina filter kernela(impulsnog odziva):

$$L = \frac{4}{BW} = \frac{4}{0.03125} = 128 \tag{7.26}$$

Na slici 7.8. se može vidjeti filtrirani izlazni signal u crvenoj boji i ulazni signal u plavoj boji. Izlazni signal ima određeno kašnjenje, koje otprilike odgovara duljini kernela, te se kasnije uspostavlja normalno stanje. Ulazni signal je uzorkovan s 48000 Hz i sastoji se od dvije komponente, signala od 1 kHz i signala od 15 kHz (koji se tumači kao šum).



Slika 7.8. Nisko propusni filter granične frekvencije 3000 Hz s duljinom kernela 128

7.2. Visoko propusni FIR filter

Za nisko propusni filter trebalo je računati diskretnu inverznu furijerovu transformaciju frekvencijske karakteristike nisko propusnog filtra odnosno naći h_{sinc}. Po istoj logici bi vrijedilo pravilo i za preostale filtere, ali kod FIR filtra to je nije potrebno **već se iz nisko propusnog filtera kreiraju preostali filteri.**

Postupak kojim se nisko propusni filteri pretvaraju u visokopropusni filter se naziva **spektralna inverzija i sastoji se od dva koraka**:

1. Tako da se promijeni predznak svake vrijednosti filter kernela $h[n] = -h[n]$
2. Sredina filter kernela se poveća za 1 odnosno vrijedi $h\left[\frac{M}{2}\right] = h\left[\frac{M}{2}\right] + 1$

Ako je odrezna frekvencija visokopropusnog filtera f_c**=11000 Hz** onda se njena normalizirana kružna frekvencija računa prema:

$$\omega_c = \frac{2\pi f_c}{f_s} \tag{7.27}$$

$$\omega_c = \frac{2 \cdot \pi \cdot 11000}{48000} = \frac{22000 \cdot \pi}{48000} \tag{7.28}$$

$$\omega_c = 0.45\pi \ \frac{rad}{uzorak}$$

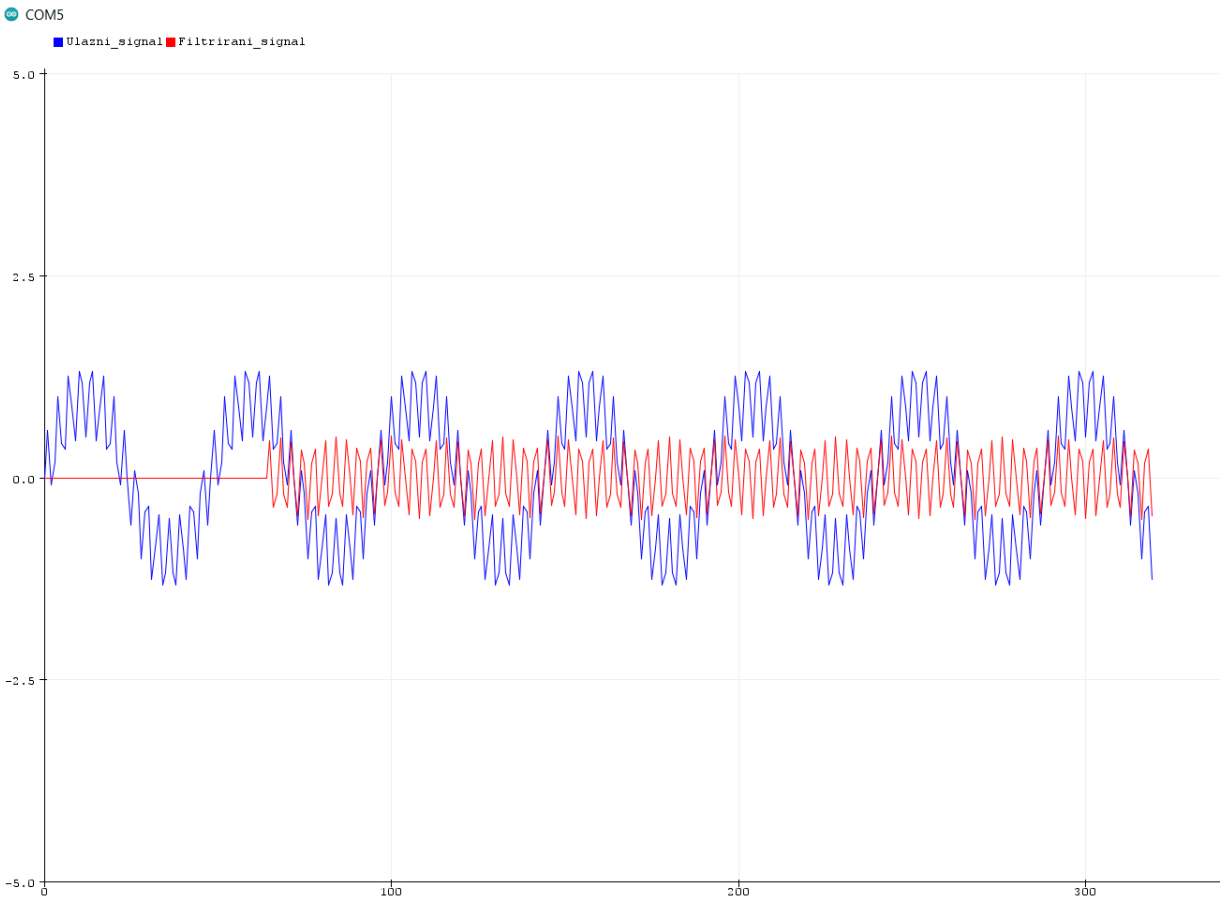
Ako želimo širinu prijelaznog pojasa od 3000 Hz onda nju računamo prema izrazu:

$$BW = \frac{B}{f_s} = \frac{3000}{48000} = 0.0625 \tag{7.29}$$

Te je dužina filter kernela(impulsnog odziva):

$$L = \frac{4}{BW} = \frac{4}{0.0625} = 64 \tag{7.30}$$

Na slici 7.9. se može vidjeti filtrirani izlazni signal u crvenoj boji i ulazni signal u plavoj boji. Izlazni signal ima određeno kašnjenje, koje otprilike odgovara duljini kernela, te se kasnije uspostavlja normalno stanje. Ulazni signal je uzorkovan s 48000 Hz i sastoji se od dvije komponente, signala od 1 kHz i signala od 15 kHz (koji se tumači kao šum).



Slika 7.9. Visoko propusni filter granične frekvencije 11000 Hz s duljinom kernela 64.

7.3. Pojasno nepropusni filter FIR filter

Pojasno nepropusni filter se kreira kao kombinacija niskopropusnog filtra i visokopropusnog filtra.

Granična frekvencija nisko propusnog filtera predstavlja **donju graničnu frekvenciju**, dok **granična frekvencija visoko propusnog filtera** predstavlja **gornju graničnu frekvenciju.**

Zbrajanjem nisko propusnog kernela i visoko propusnog filter kernela stvara se novi pojasno nepropusni filter kernel:

$$h[n] = h_{NP}[n] + h_{VP}[n] \tag{7.31}$$

Treba napomenuti da nisko i visoko propusni filteri imaju filter kernele iste duljine odnosno prijelazni im pojasevi trebaju biti iste duljine, da bi se mogli ispravno zbrojiti.

Ako je granična frekvencija nisko propusnog filtra **12000 Hz, tada je to ujedno i donja granična frekvencija**:

$$\omega_d = \frac{2\pi f_c}{f_s} \tag{7.32}$$

$$\omega_d = \frac{2 \cdot \pi \cdot 12000}{48000} = \frac{24000 \cdot \pi}{48000} \tag{7.33}$$

$$\omega_d = 0.5\pi \, \frac{rad}{uzorak}$$

Ako je granična frekvencija visoko propusnog filtra **17000 Hz**, **tada je to ujedno i gornja granična frekvencija**:

$$\omega_g = \frac{2\pi f_c}{f_s} \tag{7.34}$$

$$\omega_g = \frac{2 \cdot \pi \cdot 17000}{48000} = \frac{34000 \cdot \pi}{48000} \tag{7.35}$$

$$\omega_g = 0.7\pi \frac{rad}{uzorak}$$

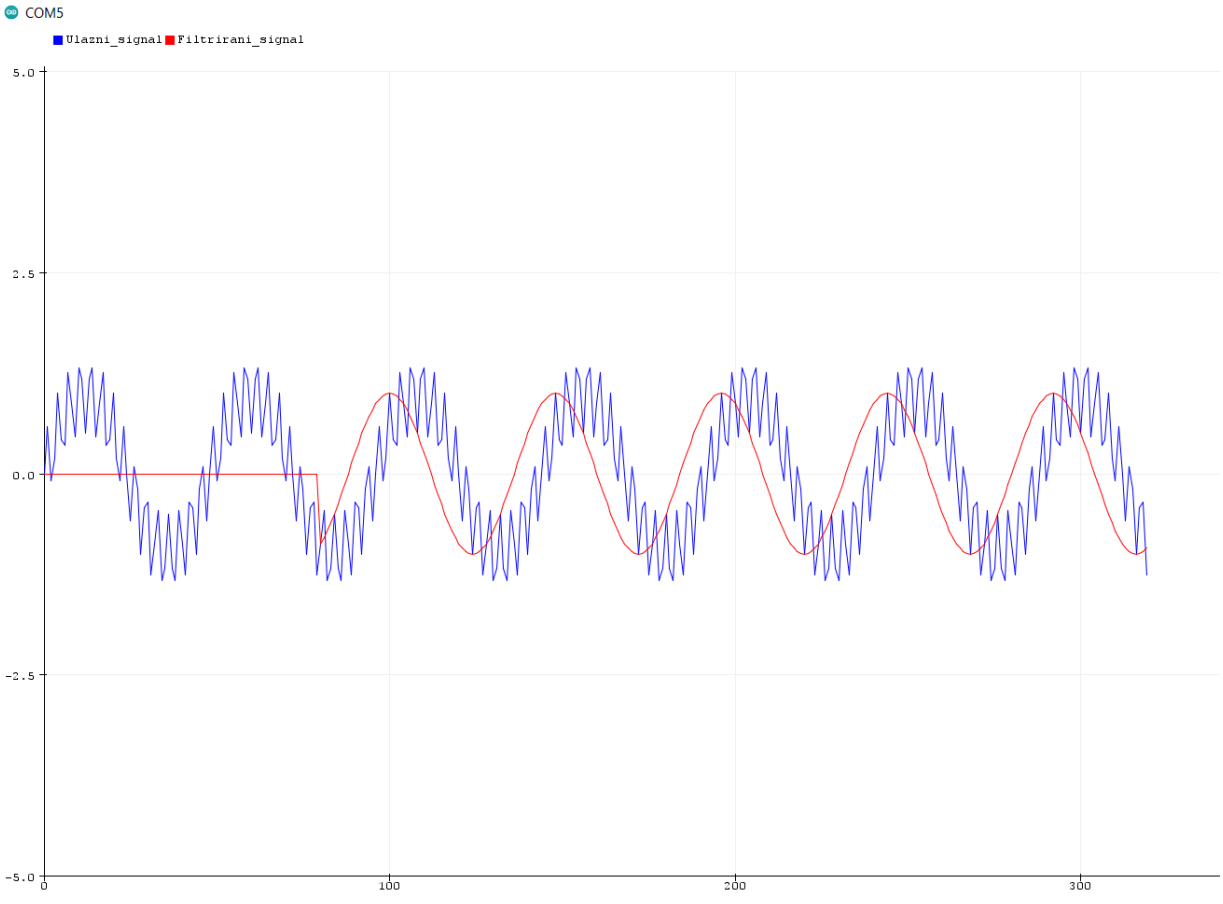
Ako želimo širinu prijelaznog pojasa od 2400 Hz kod nisko i visoko propusnog kernela onda nju računamo prema izrazu:

$$BW = \frac{B}{f_s} = \frac{2400}{48000} = 0.05 \tag{7.36}$$

Te je dužina filter kernela(impulsnog odziva):

$$L = \frac{4}{BW} = \frac{4}{0.05} = 80 \tag{7.37}$$

Na slici 7.10. se može vidjeti filtrirani izlazni signal u crvenoj boji i ulazni signal u plavoj boji. Izlazni signal ima određeno kašnjenje, koje otprilike odgovara duljini kernela, te se kasnije uspostavlja normalno stanje. Ulazni signal je uzorkovan s 48000 Hz i sastoji se od dvije komponente, signala od 1 kHz i signala od 15 kHz (koji se tumači kao šum).



Slika 7.10. Pojasno nepropusni filter od 12000 Hz do 17000 Hz s duljinom kernela 80

7.4. Pojasno propusni FIR filter

Pojasno propusni filter se isto kreira kao kombinacija nisko propusnog filtra i visoko propusnog filtra.

Granična frekvencija nisko propusnog filtera predstavlja **donju graničnu frekvenciju**, dok **granična frekvencija visoko propusnog filtera** predstavlja **gornju graničnu frekvenciju.**

Zbrajanjem nisko propusnog kernela i visoko propusnog filter kernela stvara se novi pojasno nepropusni filter kernel:

$$h_{PN}[n] = h_{NP}[n] + h_{VP}[n] \tag{7.38}$$

Sada da bi se dobio filter kernel pojasno propusnog filtera potrebno je izvršiti **spektralnu inverziju kernela $h_{PN}[n]$ od pojasno nepropusnog filtra,** te imamo pojasno propusni kernel:

$$h_{PP}[n] = -h_{PN}[n]$$

Ako je granična frekvencija nisko propusnog filtra **500 Hz, tada je to ujedno i donja granična frekvencija**:

$$\omega_c = \frac{2\pi f_c}{f_s} \tag{7.39}$$

$$\omega_c = \frac{2 \cdot \pi \cdot 500}{48000} = \frac{1000 \cdot \pi}{48000} \tag{7.40}$$

$$\omega_c = 0.02\pi \, \frac{rad}{uzorak}$$

Ako je granična frekvencija HP filtra **5000 Hz**, tada je to ujedno i gornja granična frekvencija prema:

$$\omega_c = \frac{2\pi f_c}{f_s} \tag{7.41}$$

$$\omega_c = \frac{2 \cdot \pi \cdot 5000}{48000} = \frac{10000 \cdot \pi}{48000} \tag{7.42}$$

$$\omega_c = 0.2\pi \, \frac{rad}{uzorak}$$

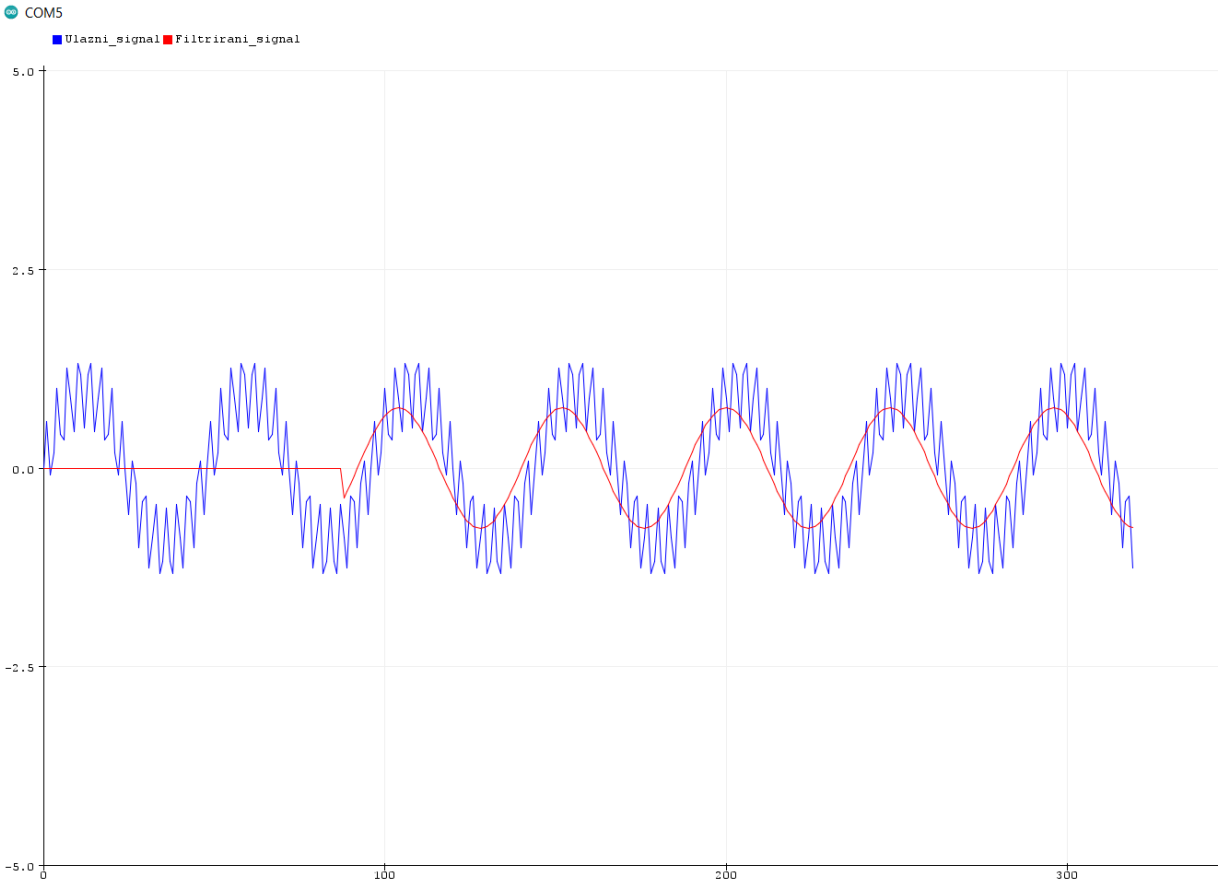
Ako želimo širinu prijelaznog pojasa od 2200 Hz onda nju računamo prema izrazu:

$$BW = \frac{B}{f_s} = \frac{2200}{48000} = 0.045 \tag{7.43}$$

Te je dužina filter kernela(impulsnog odziva):

$$L = \frac{4}{BW} = \frac{4}{0.045} = 88 \tag{7.44}$$

Na slici 7.11. se može vidjeti filtrirani izlazni signal u crvenoj boji i ulazni signal u plavoj boji. Izlazni signal ima određeno kašnjenje, koje otprilike odgovara duljini kernela, te se kasnije uspostavlja normalno stanje. Ulazni signal je uzorkovan s 48000 Hz i sastoji se od dvije komponente, signala od 1 kHz i signala od 15 kHz (koji se tumači kao šum).



Slika 7.11. Pojasno propusni filter od 500 Hz do 5000 Hz s duljinom kernela 88

8. IIR FILTERI

IIR filteri su filteri čiji je impulsni odziv beskonačan, odnosno njegova vrijednost nikada ne padne na 0 već istitrava u beskonačnost.

Za razliku od FIR filtera vrijednosti prethodno izračunatog izlaznog signala $y[n-k]$ utječu na trenutne vrijednosti izlaznog signala $y[n]$.

Jednadžba za izračun IIR filtera je:

$$\sum_{k=0}^{N-1} a_k y[n-k] = \sum_{k=0}^{N-1} b_k x[n-k] \tag{8.1}$$

$$y[n] = \sum_{k=0}^{N-1} b_k x[n-k] - \sum_{k=1}^{N-1} a_k y[n-k] \tag{8.2}$$

Gdje su:

a_k – koeficijenti povratne veze(eng. *feedback coefficients*)

b_k – koeficijenti ulaznih vrijednosti(eng. *feedforward coefficients*)

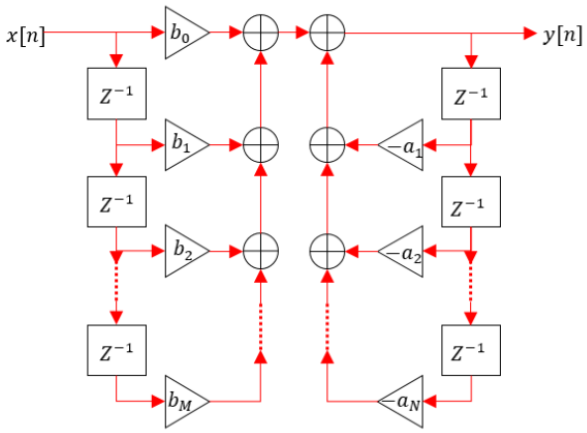
Radi lakšeg shvaćanja, kada se raspiše, jednadžba (8.2) izgleda:

$$y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] + b_3 x[n-3] + \cdots \tag{8.3}$$
$$- a_1 y[n-1] - a_2 y[n-2] - a_3 y[n-3] - \cdots$$

Iz jednadžbe (8.3) se vidi da prethodno izračunate vrijednosti izlaznog signala $y[n-1]$, $y[n-2]$, $y[n-3]$ skupa sa svojim odgovarajućim koeficijentima a_k utječu na trenutni izlazni signal $y[n]$.

Takva jednadžba se ujedno i naziva **rekurzijska jednadžba(engl. *recursion equation*)**, a IIR filteri se još nazivaju **rekurzijski filteri(engl. *recursion filters*).**
Pripadajući koeficijeniti a_k i b_k se nazivaju **rekurzijski koeficijenti(engl. *recursion coefficients*).**

Blok shema IIR filtera se može vidjeti na slici 8.1.:



Slika 8.1. Blok shema IIR filtera

Za razliku od FIR filtera, **IIR filteri nastaju transformacijom analognih filtera u digitalne filtere.**

Postupak transformacije analognih filtera u digitalne filtere je sljedeći:

1. Odabere se željena diskretna kružna frekvencija ω_d, koja se dobija poznavanjem željene granične frekvencije f_g te poznavanjem perioda T ili frekvencije uzorkovanja f_s.

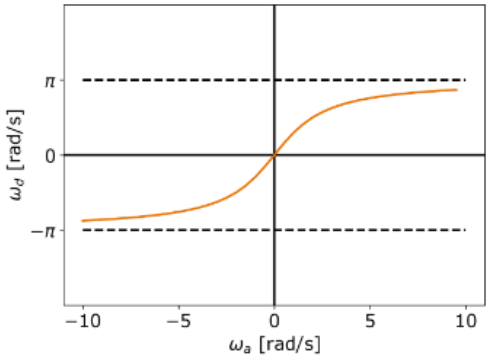
2. Potrebno je preslikati diskretnu kružnu frekvenciju ω_d, u analognu kružnu frekvenciju ω_a korištenjem izraza:

$$\omega_a = \frac{2}{T}\tan\left(\frac{\omega_d}{2}\right) \tag{8.4}$$

Ovaj postupak se naziva frekvencijsko sabijanje ili savijanje (engl. *frequency warping*).

S izrazom se analogne frekvencije ω_a koje idu od $-\infty$ do $+\infty$ usklađuju s diskretnim kružnim frekvencijama ω_d koje se kreću na opsegu od $-\pi$ do π.

Na slici 8.2. se vidi da je za manje frekvencije usklađivanje linearno, dok je za veće frekvencije usklađivanje nelinearno, te je potrebno koristiti navedeni izraz (8.4).



Slika 8.2. Frekvencijsko savijanje – odnos analogne i digitalne frekvencije

3. Vrši se transformacija prijenosne funkcije(s područje) u prijenosnu funkciju(s područje) željenog filtra pomoću prototipa filtera. Prototipi filtera su slijedeći:

- Nisko propusni filtar u nisko propusni filtar:

$$s \to \frac{s}{\omega_a} \tag{8.5}$$

- Nisko propusni filtar u visoko propusni filtar:

$$s \to \frac{\omega_a}{s} \tag{8.6}$$

- Nisko propusni filtar u pojasno propusni filtar:

$$s \to \frac{s^2 + \omega_0}{\Delta\omega \cdot s} \tag{8.7}$$

- Nisko propusni filtar u pojasno nepropusni filtar:

$$s \to \frac{\Delta\omega \cdot s}{s^2 + \omega_0} \tag{8.8}$$

Gdje je $\omega_o = \sqrt{\omega_{a1} \cdot \omega_{a2}}$ centralna frekvencija, a $\Delta\omega = \omega_{a2} - \omega_{a1}$ širina propusnog ili nepropusnog pojasa, ovisno o tipu filtera bio to pojasno propusni ili pojasno nepropusni filtar. Oznaka ω_{a1} predstavlja **donju graničnu frekvenciju**, dok oznaka ω_{a2} predstavlja **gornju graničnu frekvenciju**.

Kod nisko propusnog i visoko propusnog filtra umjesto oznake ω_a može se koristiti oznaka ω_0, obje imaju isto značenje.

Oznake ω_o i $\Delta\omega$, odnosno njihov omjer daje faktor kvalitete Q (engl. *Quality factor*):

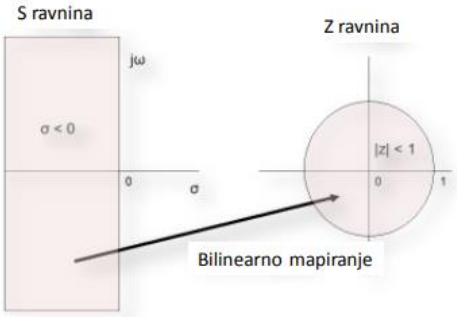
$$Q = \frac{\omega_o}{\Delta\omega} \tag{8.9}$$

Postupak se vrši tako da se s parametar u prijenosnoj funkciji zamjeni sa s parametrom željenog filtarskog prototipa.

4. Nakon dobijene prijenosne funkcije željenog filtra u s domeni, prijenosna funkcija se sada prebacuje u z domenu postupkom koji se zove bilinearna transformacija. Prebacivanje se vrši pomoću izraza:

$$s = \frac{2}{T} \cdot \frac{1 - z^{-1}}{1 + z^{-1}} = \frac{2 - 2z^{-1}}{T + Tz^{-1}} \tag{8.10}$$

Postupkom bilinearne transformacije se cijela desna strana s područja prebacuje u jedinični krug z područja. Razlog toga je što se na desnoj strani s domene nalaze polovi kojima se osigurava stabilnost određenog sustava. Bilinearnom transformacijom polovi se mapiraju unutar i na granicu jediničnog kruga u z domeni.



Slika 8.3. Bilinearna transformacija – Mapiranje s domene u z domenu

5. Kada se izvrši bilinearna transformacija prijenosnu funkciju u z domeni potrebno je svesti na oblik:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} \ldots}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} \ldots} \tag{8.11}$$

Iz oblika se mogu izvući koeficijenti a i b koji se samo uvrste u rekurzijsku jednadžbu te se na izlazu dobija filtrirani signal y[n].

Cijeli postupak će biti jasniji na slijedećim primjerima.

8.1. Nisko propusni IIR filter

Normirana prijenosna funkcija nisko propusnog filtera je

$$H(s) = \frac{1}{s+1} \tag{8.12}$$

Ako je:

- frekvencija uzorkovanja $f_s = 48000\ Hz$, odnosno period $T = 20.83\ \mu s$.
- granična frekvencija $f_g = 3000\ Hz$

Prvi korak je naći diskretnu kružnu frekvenciju ω_d.

To se obavlja prema:

$$\omega_d = \frac{2\pi f_g}{f_s}$$

$$\omega_d = \frac{2 \cdot \pi \cdot 3000}{48000} = \frac{6000 \cdot \pi}{48000} \tag{8.13}$$

$$\omega_d = 0.125\,\pi \frac{rad}{uzorak}$$

Zatim je potrebno preslikati diskretnu kružnu frekvenciju ω_d u analognu kružnu frekvenciju ω_a korištenjem izraza:

$$\omega_a = \frac{2}{T}\tan\left(\frac{\omega_d}{2}\right) = 19095.59\,\frac{rad}{s} \tag{8.14}$$

Korištenjem prethodno uspostavljenih prototipova filtra potrebno je normiranu prijenosnu funkciju nisko propusnog filtra pretvoriti u klasičnu prijenosnu funkciju nisko propusnog filtra. To se vrši tako da se izraz za prototip nisko propusnog filtra uvrsti u normiranu prijenosnu funkciju nisko propusnog filtra:

$$H(s) = \frac{1}{s+1}\bigg|_{s = \frac{s}{\omega_a}} \tag{8.15}$$

Tako se dobije prijenosna funkcija nisko propusnog filtra:

$$H(s) = \frac{\omega_a}{s + \omega_a} \tag{8.16}$$

Umjesto oznake ω_a može se koristi oznaka ω_0, te je sada prijenosna funkcija:

$$H(s) = \frac{\omega_0}{s + \omega_0} \tag{8.17}$$

Korištenjem bilinearne transformacije prijenosna funkcija se prebacuje iz s domene u z domenu:

$$H(z) = \frac{\omega_0}{\left(\frac{2}{T} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}\right) + \omega_0} = \frac{\omega_0}{\frac{2 - 2z^{-1} + \omega_0(T + Tz^{-1})}{T + Tz^{-1}}} \tag{8.18}$$

Sređivanjem se dobija:

$$H(z) = \frac{(\omega_0 T) + \omega_0 T z^{-1}}{(2 + \omega_0 T) + (\omega_0 T - 2)z^{-1}} \tag{8.19}$$

Jednadžbu (8.18) je potrebno svesti na određeni oblik:

$$H(Z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1}} \tag{8.20}$$

To se dobija tako da se brojnik i nazivnik podijele sa $(2 + \omega_0 T)$:

$$H(z) = \frac{(\frac{\omega_0 T}{2 + \omega_0 T}) + (\frac{\omega_0 T}{2 + \omega_0 T})z^{-1}}{1 + (\frac{\omega_0 T - 2}{2 + \omega_0 T})z^{-1}} \tag{8.21}$$

Iz jednadžbe (8.20) se izvuku koeficijenti b:

$$b_0 = \frac{\omega_0 T}{2 + \omega_0 T} \qquad\qquad b_1 = \frac{\omega_0 T}{2 + \omega_0 T} \tag{8.22}$$

I koeficijent a:

$$a_1 = \frac{\omega_0 T - 2}{2 + \omega_0 T} \tag{8.23}$$

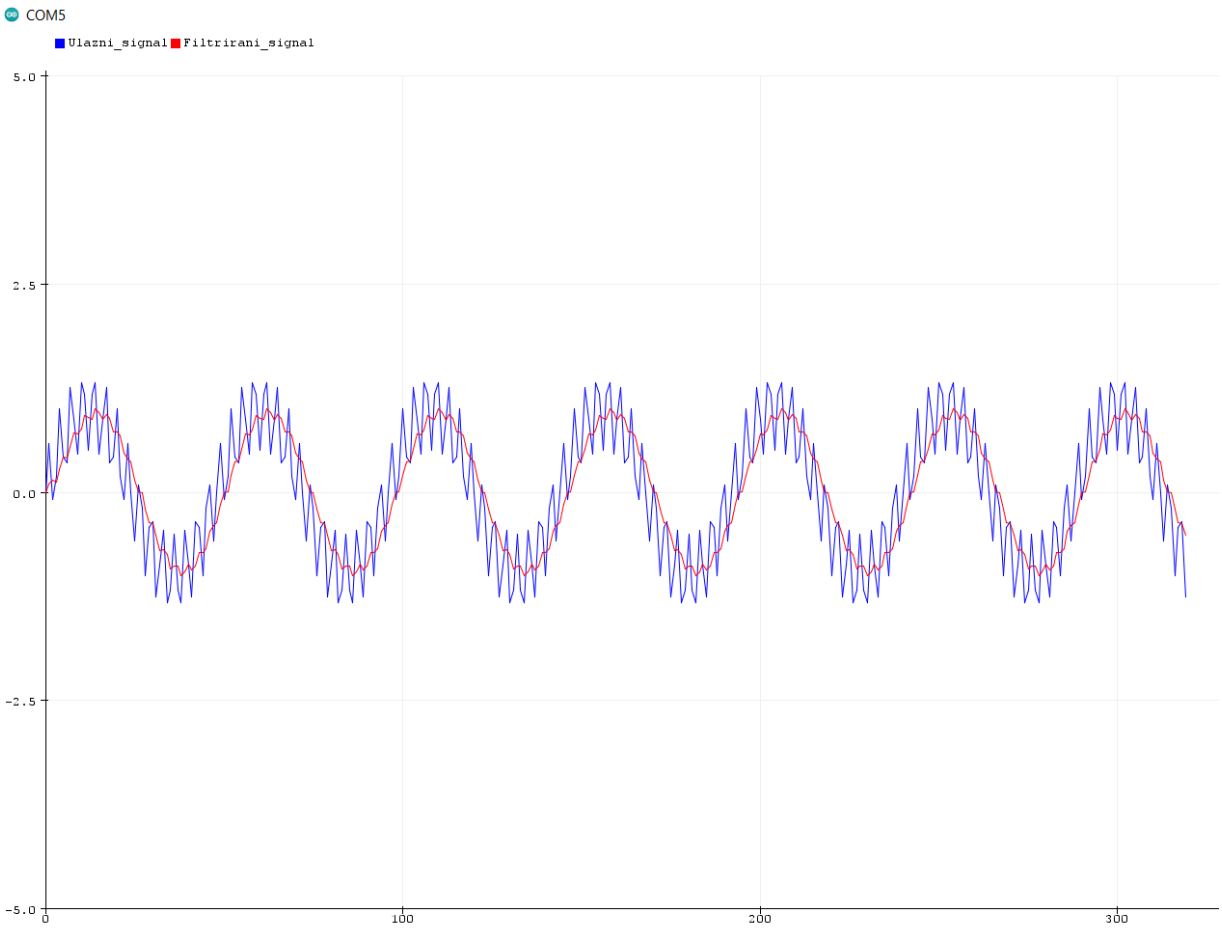
Prethodno izračunati ω_a, odnosno ω_0 i T se samo uvrste u jednadžbe (8.22) i (8.23) te se dobiju vrijednosti za koeficijene a i b.

Izračunati koeficijenti se samo uvrste u rekurzijsku jednadžbu i na izlazu se dobija filtrirani signal y[n].

$$y[n] = b_0 x[n] + b_0 x[n-1] - a_1 y[n-1] \tag{8.24}$$

$$y[n] = 0.17x[n] + 0.17x[n-1] - (-0.67)y[n-1]$$

Na slici 8.4. se može vidjeti filtrirani izlazni signal u crvenoj boji i ulazni signal u plavoj boji. Ulazni signal je uzorkovan s 48000 Hz i sastoji se od dvije komponente, signala od 1 kHz i signala od 15 kHz (koji se tumači kao šum).



Slika 8.4. Nisko propusni filter pri graničnoj frekvenciji od 3000 Hz

8.2. Visoko propusni IIR filter

Normirana prijenosna funkcija nisko propusnog filtera je

$$H(s) = \frac{1}{s+1} \tag{8.25}$$

Ako je:

- frekvencija uzorkovanja $f_s = 48000\ Hz$, odnosno period je $T = 20.83\ \mu s$.
- granična frekvencija $f_g = 13000\ Hz$

Prvi korak je naći diskretnu kružnu frekvenciju ω_d.

To se obavlja prema:

$$\omega_d = \frac{2\pi f_g}{f_s}$$

$$\omega_d = \frac{2 \cdot \pi \cdot 13000}{48000} = \frac{26000 \cdot \pi}{48000} \tag{8.26}$$

$$\omega_d = 0.54\,\pi\ \frac{rad}{uzorak}$$

Zatim je potrebno preslikati diskretnu kružnu frekvenciju ω_d u analognu kružnu frekvenciju ω_a korištenjem izraza:

$$\omega_a = \frac{2}{T}\tan\left(\frac{\omega_d}{2}\right) = 109467\frac{rad}{s} \tag{8.27}$$

Korištenjem prethodno uspostavljenih prototipova filtra potrebno je normiranu prijenosnu funkciju nisko propusnog filtra pretvoriti u klasičnu prijenosnu funkciju visoko propusnog filtra. To se vrši tako da se izraz za prototip filtra uvrsti u normiranu prijenosnu funkciju nisko propusnog filtra:

$$H(s) = \frac{1}{s+1}\bigg|_{s = \frac{\omega_a}{s}} \tag{8.28}$$

Tako se dobije prijenosna funkcija visoko propusnog filtra:

$$H(s) = \frac{s}{s + \omega_a} \tag{8.29}$$

U literaturi se umjesto oznake ω_a često koristi oznaka ω_0, te je sada prijenosna funkcija:

$$H(s) = \frac{s}{s + \omega_0} \tag{8.30}$$

Korištenjem bilinearne transformacije prijenosna funkcija se prebacuje iz s domene u z domenu:

$$H(z) = \frac{\left(\frac{2}{T} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}\right)}{\left(\frac{2}{T} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}\right) + \omega_0} = \frac{\left(\frac{2}{T} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}\right)}{\frac{2 - 2z^{-1} + \omega_0(T + Tz^{-1})}{T + Tz^{-1}}} \tag{8.31}$$

Sređivanjem jednadžbe (8.31) se dobija:

$$H(z) = \frac{2 - 2z^{-1}}{(2 + \omega_0 T) + (\omega_0 T - 2)z^{-1}} \tag{8.32}$$

Jednadžbu je potrebno svesti na određeni oblik:

$$H(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1}} \tag{8.33}$$

To se dobija tako da se brojnik i nazivnik podijele sa $(2 + \omega_0 T)$:

$$H(z) = \frac{(\frac{2}{2 + \omega_0 T}) + (\frac{-2}{2 + \omega_0 T})z^{-1}}{1 + (\frac{\omega_0 T - 2}{2 + \omega_0 T})z^{-1}} \tag{8.34}$$

Iz jednadžbe 8.33 se izvuku koeficijenti b:

$$b_0 = \frac{2}{2 + \omega_0 T} \qquad\qquad b_1 = \frac{-2}{2 + \omega_0 T} \tag{8.35}$$

I koeficijent a:

$$a_1 = \frac{\omega_0 T - 2}{2 + \omega_0 T} \tag{8.36}$$

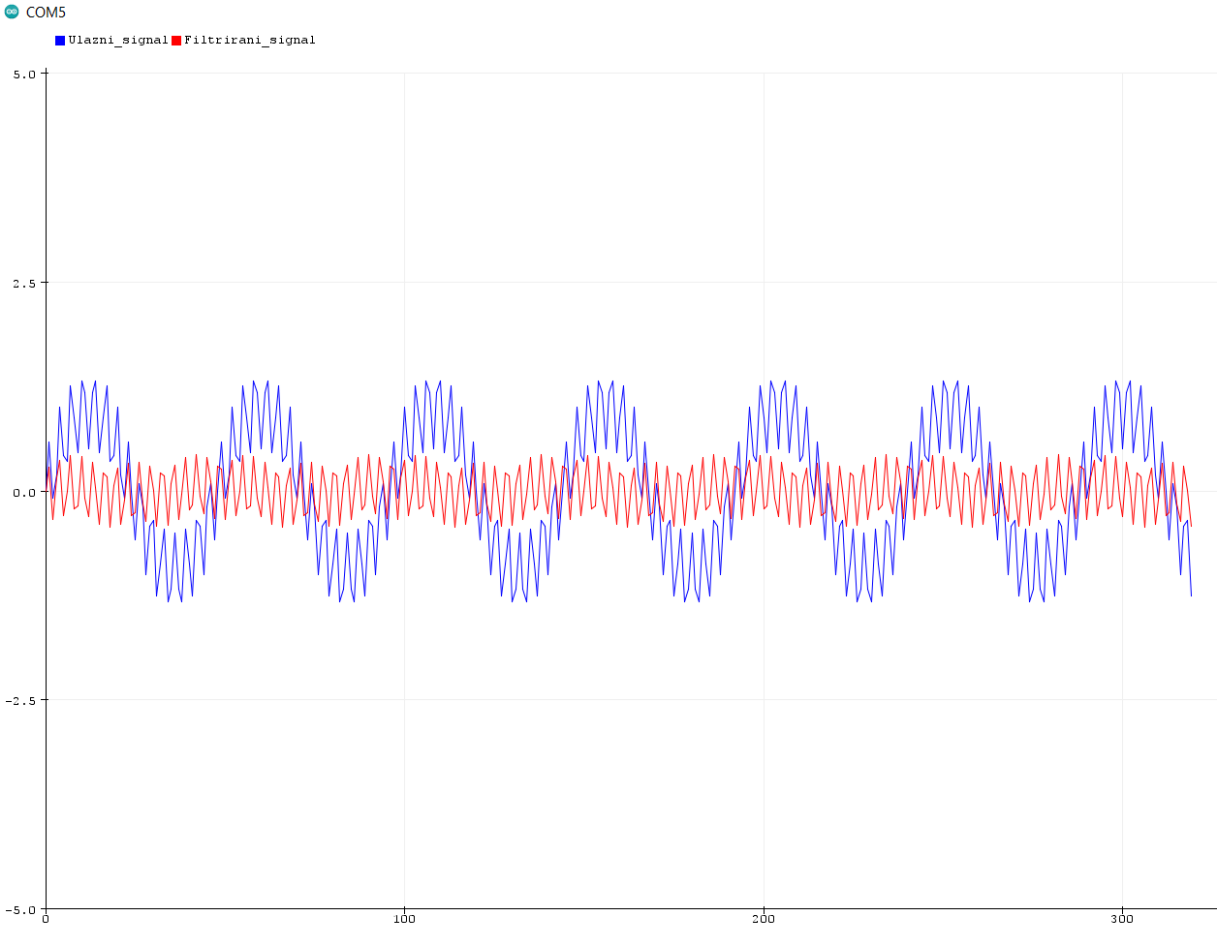
Prethodno izračunati ω_a, odnosno ω_0 i T se samo uvrste u jednadžbe (8.35) i (8.36) te se dobiju vrijednosti za koeficijene a i b.

Izračunati koeficijenti se samo uvrste u rekurzijsku jednadžbu i na izlazu se dobija filtrirani signal y[n].

$$y[n] = b_0 x[n] + b_0 x[n-1] - a_1 y[n-1] \tag{8.37}$$

$$y[n] = 0.47x[n] + (-0.47)x[n-1] - (-0.07)y[n-1]$$

Na slici 8.5. se može vidjeti filtrirani izlazni signal u crvenoj boji i ulazni signal u plavoj boji. Ulazni signal je uzorkovan s 48000 Hz i sastoji se od dvije komponente, signala od 1 kHz i signala od 15 kHz (koji se tumači kao šum).



Slika 8.5. Visoko propusni filter pri graničnoj frekvenciji od 12000 Hz

8.3. Pojasno propusni IIR filter

Normirana prijenosna funkcija nisko propusnog filtera je

$$H(s) = \frac{1}{s+1} \tag{8.38}$$

Ako je:

- frekvencija uzorkovanja $f_s = 48000\ Hz$, odnosno period je $T = 20.83\ \mu s$.
- donja granična frekvencija $f_1 = 500\ Hz$
- gornja granična frekvencija $f_2 = 1500\ Hz$

Nađe se diskretna kružna frekvencija za ω_{d1} donju i ω_{d2} za gornju graničnu frekvenciju.

To se obavlja prema:

$$\begin{aligned} \omega_{d1} &= \frac{2\pi f_g}{f_s} & \omega_{d2} &= \frac{2\pi f_g}{f_s} \\ \omega_{d1} &= \frac{2 \cdot \pi \cdot 500}{48000} = \frac{1000 \cdot \pi}{48000} & \omega_{d2} &= \frac{2 \cdot \pi \cdot 1500}{48000} = \frac{3000 \cdot \pi}{48000} \\ \omega_{d1} &= 0.02\,\pi \frac{rad}{uzorak} & \omega_{d2} &= 0.0625\,\pi \frac{rad}{uzorak} \end{aligned} \tag{8.39}$$

Zatim je potrebno preslikati diskretnu donju graničnu kružnu frekvenciju ω_{d1} u analognu kružnu donju graničnu frekvenciju ω_{a1} i gornju graničnu kružnu frekvenciju ω_{d2} u analognu kružnu gornju graničnu frekvenciju ω_{a2} prema izrazima:

$$\begin{aligned} \omega_{a1} &= \frac{2}{T}\tan\left(\frac{\omega_{d1}}{2}\right) & \omega_{a2} &= \frac{2}{T}\tan\left(\frac{\omega_{d2}}{2}\right) \\ \omega_{a1} &= 3142\frac{rad}{s} & \omega_{a2} &= 9455\frac{rad}{s} \end{aligned} \tag{8.40}$$

Nakon pronalaska analognih frekvencija nađe se se širina pojasa $\Delta\omega$ i centralna frekvencija ω_0 i faktor Q.

$$\omega_o = \sqrt{\omega_{a1} \cdot \omega_{a2}} = 5451\frac{rad}{s} \tag{8.41}$$

$$\Delta\omega = \omega_{a2} - \omega_{a1} = 6312\frac{rad}{s} \tag{8.42}$$

$$Q = \frac{\omega_o}{\Delta\omega} = 0.86 \tag{8.43}$$

Korištenjem prethodno uspostavljenih prototipova filtra potrebno je normiranu prijenosnu funkciju nisko propusnog filtra pretvoriti u klasičnu prijenosnu funkciju pojasno propusnog filtra.

To se vrši tako da se izrazi za prototip filtra uvrste u normiranu prijenosnu funkciju nisko propusnog filtra:

$$H(s) = \frac{1}{s+1}\bigg|_{s = \frac{s^2 + \omega_0}{\Delta\omega \cdot s}} \tag{8.44}$$

Tako se dobije prijenosna funkcija pojasno propusnog filtra:

$$H(s) = \frac{\Delta\omega \cdot s}{s^2 + \Delta\omega \cdot s + {\omega_0}^2} \tag{8.45}$$

Ako se primjeni relacija da je $Q = \frac{\omega_0}{\Delta\omega}$, odnosno $\Delta\omega = \frac{\omega_0}{Q}$ dobija se standardna funkcija pojasno propusnog filtra:

$$H(s) = \frac{\frac{\omega_0}{Q}s}{s^2 + \frac{\omega_0}{Q}s + {\omega_0}^2} \tag{8.46}$$

Korištenjem bilinearne transformacije prijenosna funkcija se prebacuje u z domenu:

$$H(z) = \frac{\frac{\omega_0}{Q} \cdot \left(\frac{2 - 2z^{-1}}{T + Tz^{-1}}\right)}{\frac{(2 - 2z^{-1})^2}{(T + Tz^{-1})^2} + \frac{\omega_0}{Q}\left(\frac{2 - 2z^{-1}}{T + Tz^{-1}}\right) + {\omega_0}^2} \tag{8.47}$$

Brojnik i nazivnik jednadžbe (8.47) je potrebno podijeliti s $(T + Tz^{-1})^2$ time se dobija:

$$H(z) = \frac{\frac{\omega_0}{Q} \cdot (T + Tz^{-1})(2 - 2z^{-1})}{(2 - 2z^{-1})^2 + \frac{\omega_0}{Q}(T + Tz^{-1})(2 - 2z^{-1}) + {\omega_0}^2(T + Tz^{-1})^2} \tag{8.48}$$

Sređivanjem jednadžbe (8.48) se dobije

$$H(z) = \frac{\frac{2\omega_0 T}{Q} + \left(-\frac{2\omega_0 T}{Q}\right) z^{-2}}{\left(4 + \frac{2\omega_0 T}{Q} + {\omega_0}^2 T^2\right) + (2{\omega_0}^2 T^2 - 8) z^{-1} + \left(4 - \frac{2\omega_0 T}{Q} + {\omega_0}^2 T^2\right) z^{-2}} \tag{8.49}$$

Jednadžbu (8.49) je potrebno svesti na određeni oblik:

$$H(Z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{8.50}$$

To se dobija tako da se **brojnik i nazivnik podijele** sa $\left(4 + \frac{2\omega_0 T}{Q} + {\omega_0}^2 T^2\right)$ i slijedi:

$$H(z) = \frac{\frac{\frac{2\omega_0 T}{Q}}{4 + \frac{2\omega_0 T}{Q} + {\omega_0}^2 T^2} + \left(\frac{-\frac{2\omega_0 T}{Q}}{4 + \frac{2\omega_0 T}{Q} + {\omega_0}^2 T^2}\right) z^{-2}}{1 + \left(\frac{2{\omega_0}^2 T^2 - 8}{4 + \frac{2\omega_0 T}{Q} + {\omega_0}^2 T^2}\right) z^{-1} + \left(\frac{4 - \frac{2\omega_0 T}{Q} + {\omega_0}^2 T^2}{4 + \frac{2\omega_0 T}{Q} + {\omega_0}^2 T^2}\right) z^{-2}} \tag{8.51}$$

Iz jednadžbe (8.51) se izvuku koeficijenti b:

$$b_0 = \frac{\frac{2\omega_0 T}{Q}}{4 + \frac{2\omega_0 T}{Q} + {\omega_0}^2 T^2} \qquad b_1 = 0 \qquad b_2 = \frac{-\frac{2\omega_0 T}{Q}}{4 + \frac{2\omega_0 T}{Q} + {\omega_0}^2 T^2} \tag{8.52}$$

Te koeficijenti a:

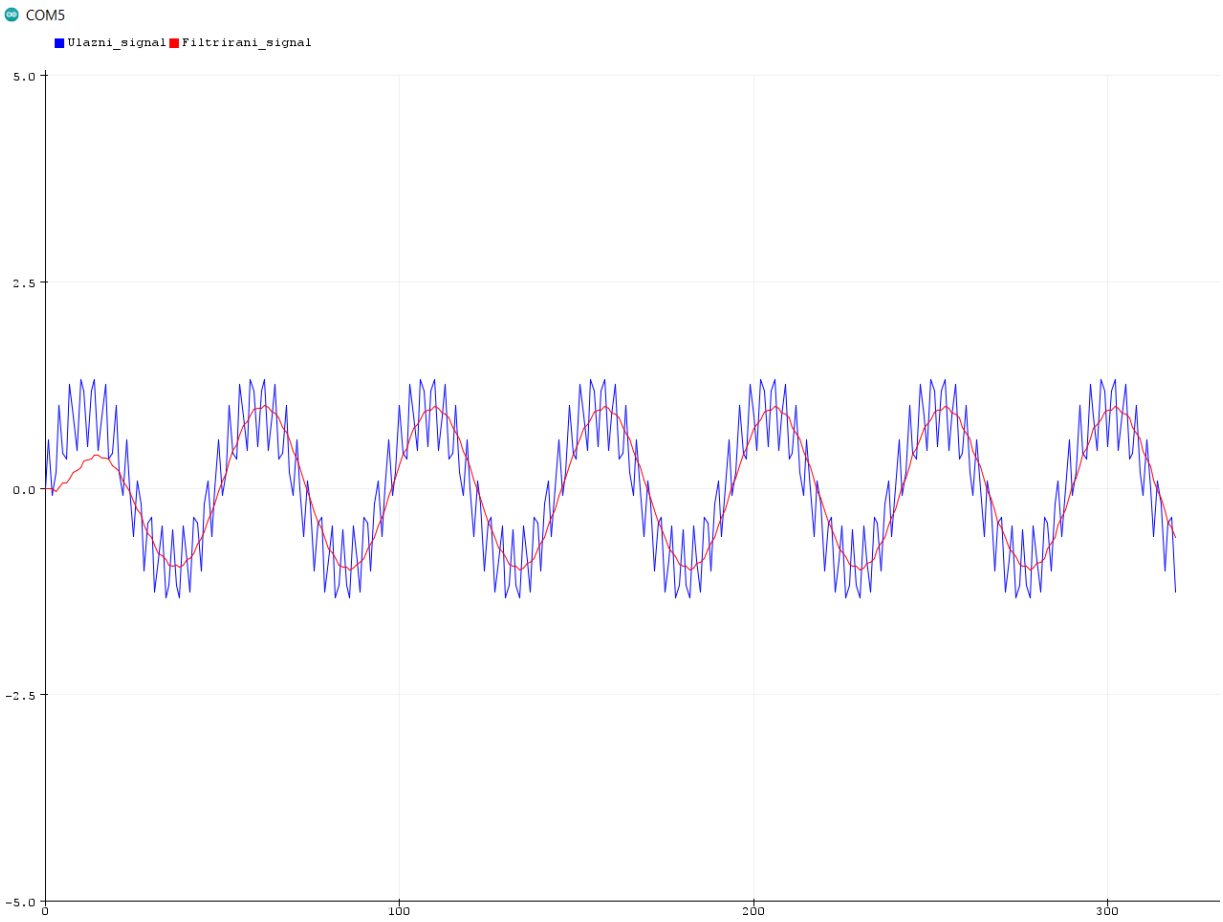
$$a_1 = \frac{2{\omega_0}^2 T^2 - 8}{4 + \frac{2\omega_0 T}{Q} + {\omega_0}^2 T^2} \qquad a_2 = \frac{4 - \frac{2\omega_0 T}{Q} + {\omega_0}^2 T^2}{4 + \frac{2\omega_0 T}{Q} + {\omega_0}^2 T^2} \tag{8.53}$$

Prethodno izračunati ω_o, Q i T se samo uvrste u jednadžbe (8.52) i (8.53) te se dobiju vrijednosti za koeficijene a i b.

$$y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] - a_1 y[n-1] - a_2 y[n-2] \tag{8.54}$$

$$y[n] = 0.06x[n] + 0x[n-1] + (-0.06)x[n-2] - (-1.86)y[n-1] - 0.88y[n-2]$$

Na slici 8.6. se može vidjeti filtrirani izlazni signal u crvenoj boji i ulazni signal u plavoj boji. Ulazni signal je uzorkovan s 48000 Hz i sastoji se od dvije komponente, signala od 1 kHz i signala od 15 kHz (koji se tumači kao šum).



Slika 8.6. Pojasno propusni filter pri graničnim frekvencijama 500 Hz i 1000 Hz

8.4. Pojasno nepropusni IIR filter

Normirana prijenosna funkcija nisko propusnog filtera je

$$H(s) = \frac{1}{s+1} \tag{8.55}$$

Ako je:

- frekvencija uzorkovanja $f_s = 48000\ Hz$, odnosno period je $T = 20.83\ \mu s$.
- donja granična frekvencija $f_1 = 500\ Hz$
- gornja granična frekvencija $f_2 = 1500\ Hz$

Nađe se diskretna kružna frekvencija za ω_{d1} donju i ω_{d2} za gornju graničnu frekvenciju.

To se obavlja prema:

$$\begin{aligned} \omega_{d1} &= \frac{2\pi f_g}{f_s} & \omega_{d2} &= \frac{2\pi f_g}{f_s} \\ \omega_{d1} &= \frac{2 \cdot \pi \cdot 500}{48000} = \frac{1000 \cdot \pi}{48000} & \omega_{d2} &= \frac{2 \cdot \pi \cdot 1500}{48000} = \frac{3000 \cdot \pi}{48000} \\ \omega_{d1} &= 0.02\,\pi \frac{rad}{uzorak} & \omega_{d2} &= 0.625\,\pi \frac{rad}{uzorak} \end{aligned} \tag{8.56}$$

Zatim je potrebno preslikati diskretnu donju graničnu kružnu frekvenciju ω_{d1} u analognu kružnu donju graničnu frekvenciju ω_{a1} i gornju graničnu kružnu frekvenciju ω_{d2} u analognu kružnu donju graničnu frekvenciju ω_{a2} prema izrazima:

$$\begin{aligned} \omega_{a1} &= \frac{2}{T}\tan\left(\frac{\omega_{d1}}{2}\right) & \omega_{a2} &= \frac{2}{T}\tan\left(\frac{\omega_{d2}}{2}\right) \\ \omega_{a1} &= 3142\frac{rad}{s} & \omega_{a2} &= 9455\frac{rad}{s} \end{aligned} \tag{8.57}$$

Nakon pronalaska analognih frekvencija nađe se se širina pojasa $\Delta\omega$ i centralna frekvencija ω_0 i faktor Q.

$$\omega_o = \sqrt{\omega_{a1} \cdot \omega_{a2}} = 5451 \frac{rad}{s} \tag{8.58}$$

$$\Delta\omega = \omega_{a2} - \omega_{a1} = 6312 \frac{rad}{s} \tag{8.59}$$

$$Q = \frac{\omega_o}{\Delta\omega} = 0.86 \tag{8.60}$$

Korištenjem prethodno uspostavljenih prototipova filtra potrebno je normiranu prijenosnu funkciju nisko propusnog filtra pretvoriti u klasičnu prijenosnu funkciju pojasno nepropusnog filtra.

To se vrši tako da se izrazi za prototip filtra uvrste u normiranu prijenosnu funkciju nisko propusnog filtra:

$$H(s) = \frac{1}{s+1}\bigg|_{s = \frac{\Delta\omega \cdot s}{s^2 + \omega_0}} \tag{8.61}$$

Tako se dobije prijenosna funkcija pojasno nepropusnog filtra:

$$H(s) = \frac{s^2 + {\omega_0}^2}{s^2 + \Delta\omega \cdot s + {\omega_0}^2} \tag{8.62}$$

Ako se primjeni relacija da je $Q = \frac{\omega_0}{\Delta\omega}$, odnosno $\Delta\omega = \frac{\omega_0}{Q}$ dobija se standardna funkcija pojasno nepropusnog filtra:

$$H(s) = \frac{s^2 + {\omega_0}^2}{s^2 + \frac{\omega_0}{Q} s + {\omega_0}^2} \tag{8.63}$$

Korištenjem bilinearne transformacije prijenosna funkcija se prebacuje u z domenu:

$$H(z) = \frac{\frac{(2 - 2z^{-1})^2}{(T + Tz^{-1})^2} + {\omega_0}^2}{\frac{(2 - 2z^{-1})^2}{(T + Tz^{-1})^2} + \frac{\omega_0}{Q}\left(\frac{2 - 2z^{-1}}{T + Tz^{-1}}\right) + {\omega_0}^2} \tag{8.64}$$

Brojnik i nazivnik jednadžbe (8.64) je potrebno podijeliti s $(T + Tz^{-1})^2$ time se dobija:

$$H(z) = \frac{(2-2z^{-1})^2 + {\omega_0}^2(T+Tz^{-1})^2}{(2-2z^{-1})^2 + \frac{\omega_0}{Q}(T+Tz^{-1})(2-2z^{-1}) + {\omega_0}^2(T+Tz^{-1})^2} \tag{8.65}$$

Sređivanjem jednadžbe se dobije

$$H(z) = \frac{(4+{\omega_0}^2T^2) + (2{\omega_0}^2T^2-8)z^{-1} + (4+{\omega_0}^2T^2)z^{-2}}{\left(4+\frac{2\omega_0T}{Q}+{\omega_0}^2T^2\right) + (2{\omega_0}^2T^2-8)z^{-1} + \left(4-\frac{2\omega_0T}{Q}+{\omega_0}^2T^2\right)z^{-2}} \tag{8.66}$$

Jednadžbu je potrebno svesti na određeni oblik:

$$H(Z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2}}{1 + a_1z^{-1} + a_2z^{-2}} \tag{8.67}$$

To se dobija tako da se **brojnik i nazivnik podijele** sa $\left(4+\frac{2\omega_0T}{Q}+{\omega_0}^2T^2\right)$ i slijedi:

$$H(z) = \frac{\frac{(4+{\omega_0}^2T^2)}{4+\frac{2\omega_0T}{Q}+{\omega_0}^2T^2} + \frac{(2{\omega_0}^2T^2-8)}{4+\frac{2\omega_0T}{Q}+{\omega_0}^2T^2} + \left(\frac{(4+{\omega_0}^2T^2)}{4+\frac{2\omega_0T}{Q}+{\omega_0}^2T^2}\right)z^{-2}}{1 + \left(\frac{2{\omega_0}^2T^2-8}{4+\frac{2\omega_0T}{Q}+{\omega_0}^2T^2}\right)z^{-1} + \left(\frac{4-\frac{2\omega_0T}{Q}+{\omega_0}^2T^2}{4+\frac{2\omega_0T}{Q}+{\omega_0}^2T^2}\right)z^{-2}} \tag{8.68}$$

Iz jednadžbe se izvuku koeficijenti b:

$$b_0 = \frac{(4+{\omega_0}^2T^2)}{4+\frac{2\omega_0T}{Q}+{\omega_0}^2T^2} \quad b_1 = \frac{(2{\omega_0}^2T^2-8)}{4+\frac{2\omega_0T}{Q}+{\omega_0}^2T^2} \quad b_2 = \frac{(4+{\omega_0}^2T^2)}{4+\frac{2\omega_0T}{Q}+{\omega_0}^2T^2} \tag{8.69}$$

Te koeficijenti a:

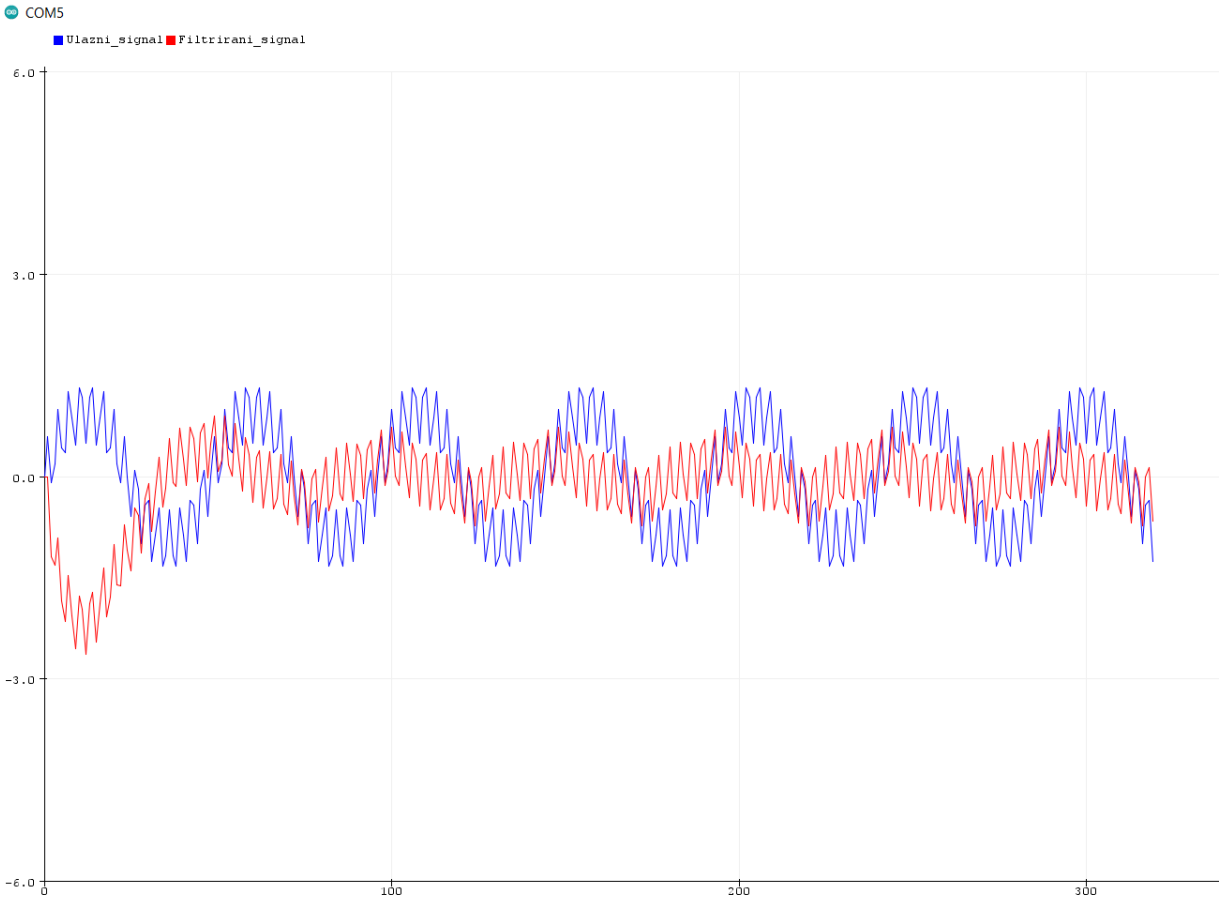
$$a_1 = \frac{2{\omega_0}^2T^2-8}{4+\frac{2\omega_0T}{Q}+{\omega_0}^2T^2} \qquad a_2 = \frac{4-\frac{2\omega_0T}{Q}+{\omega_0}^2T^2}{4+\frac{2\omega_0T}{Q}+{\omega_0}^2T^2} \tag{8.70}$$

Prethodno izračunati ω_o, Q i T se samo uvrste u jednadžbe (8.69) i (8.70) te se dobiju vrijednosti za koeficijene a i b.

$$y[n] = b_0x[n]+b_1x[n-1] + b_2x[n-2] - a_1y[n-1] - a_2y[n-2] \quad (8.71)$$

$$y[n] = 0.94x[n] + (-1.86)x[n-1] + 0.94x[n-2] - (-1.86)y[n-1] - 0.88y[n-2]$$

Na slici 8.7. se može vidjeti filtrirani izlazni signal u crvenoj boji i ulazni signal u plavoj boji. Ulazni signal je uzorkovan s 48000 Hz i sastoji se od dvije komponente, signala od 1 kHz i signala od 15 kHz (koji se tumači kao šum).



Slika 8.7. Pojasno nepropusni filter pri graničnim frekvencijama 500 Hz i 1000 Hz

8.5. Filtarske aproksimacije IIR filtera

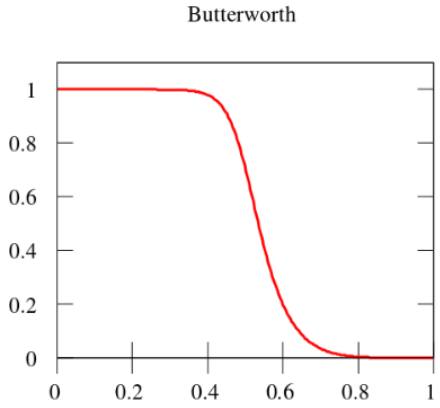
Pri dizajniranju, bilo analognih ili digitalnih filtera, nije moguće dizajnirati idealan filter, ali je takav filter moguće aproksimirati.

Idealan filter:
- nema valovanja u propusnom i nepropusnom pojasu,
- ima vrlo oštro prijelazno područje između propusnog i nepropusnog pojasa
- u propusnom pojasu propušta sve frekvencije s maksimalnim pojačanjem, dok u nepropusnom pojasu maksimalno guši sve frekvencije

Postoje 4 glavne aproksimacije idealne filtarske karakteristike:
- Butterworthova aproksimacija
- Čebiševljeva aproksimacija tipa I
- Čebiševljeva aproksimacija tipa II
- Eliptična aproksimacija

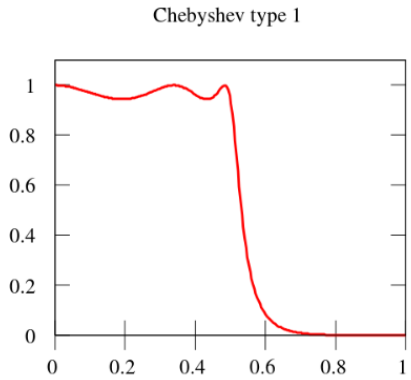
Sve navedene aproksimacije imaju svoje određene formule. Te formule po svojoj prirodi zahtjevaju dugotrajne i složene matematičke operacije, na sreću postoje određeni programski alati koji mogu dizajnirati filtar s gore navedenim aproksimacijama, odnosno nije potrebno znati matematičke izraze već je samo potrebno znati kakvo je ponašanje filtarskih aproksimacija u propusnom, nepropusnom i prijelaznom pojasu, te će ono biti pojašnjeno u daljnjem tekstu.

Butterworthov-a aproksimacija ima monotonu(ravnu) karakteristiku u propusnom i nepropusnom području, ali ima široko prijelazno područje, što se može vidjeti na slici 8.8. Ovi filteri se još nazivaju filteri s maksimalno ravnom amplitudnom karakteristikom.



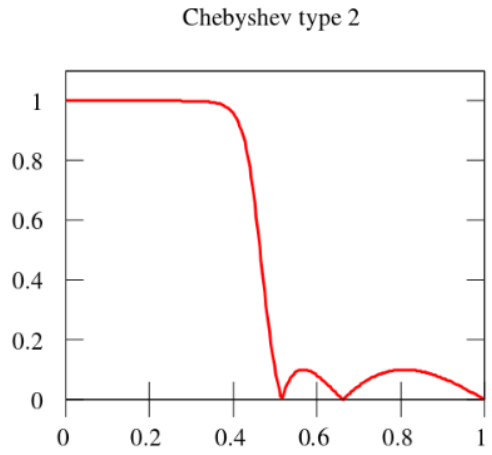
Slika 8.8. Frekvencijska karakteristika Butterworthove aproksimacije

Čebiševljeva aproksimacija tipa I ima valovitu karakteristiku u propusnom pojasu, ravnu karakteristiku u nepropusnom pojasu i ima dosta uže prijelazno području za razliku od Butterworthove aproksimacije, što se može vidjeti na slici 8.9.



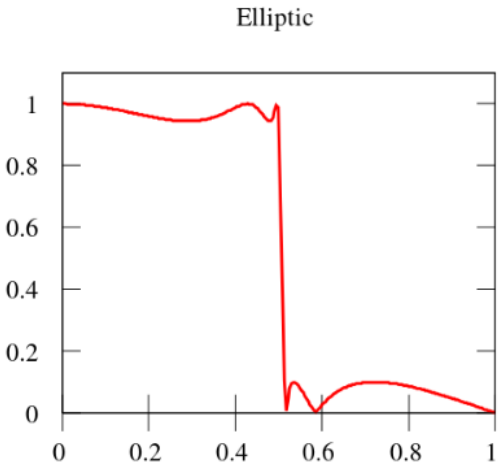
Slika 8.9. Frekvencijska karakteristika Čebiševljeve aproksimacija tipa I

Čebiševljeva aproksimacija tipa II ima valovitu karakteristiku u nepropusnom pojasu, ravnu karakteristiku u propusnom pojasu i ima također dosta uže prijelazno području za razliku od Butterworthove aproksimacije, što se može vidjeti na slici 8.10.



Slika 8.10. Frekvencijska karakteristika Čebiševljeve aproksimacija tipa II

Eliptična aprkosimacija ima valovitu karakteristiku u propusnom pojasu, valovitu karakteristiku u nepropusnom pojasu i najuže(najstrmije) prijelazno područje, što se može vidjeti na slici 8.11.



Slika 8.11. Frekvencijska karakteristika Eliptične aproksimacije

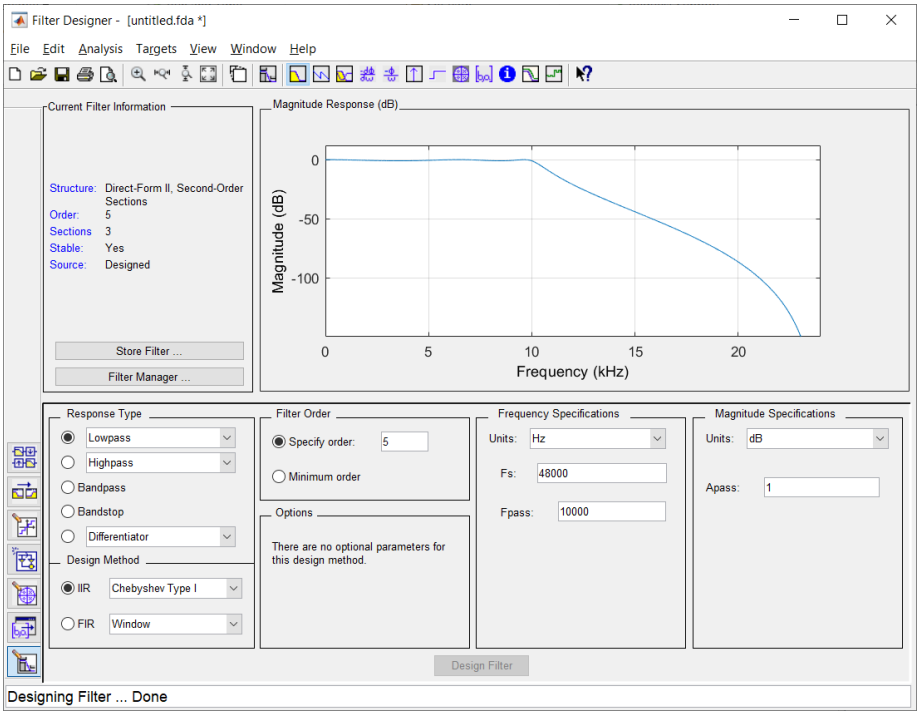
Kao što je već spomenuto, potrebno je samo poznavati njihovo ponašanje u propusnom, nepropusnom i prijelaznom pojasu, ubaciti željene parametre u program, te program izbaci filtarske koeficijente a i b.

U prethodnim poglavljima je ustanovljeno da je za željeni rad filtera potrebno samo dobiti filtarske koeficijente koji se zatim ubace u rekurzijsku jednadžbu.

Primjerice Matlab program u sebi sadrži takozvani "filter dizajner" koji se pokreće upisivanjem komande "filterDesigner".

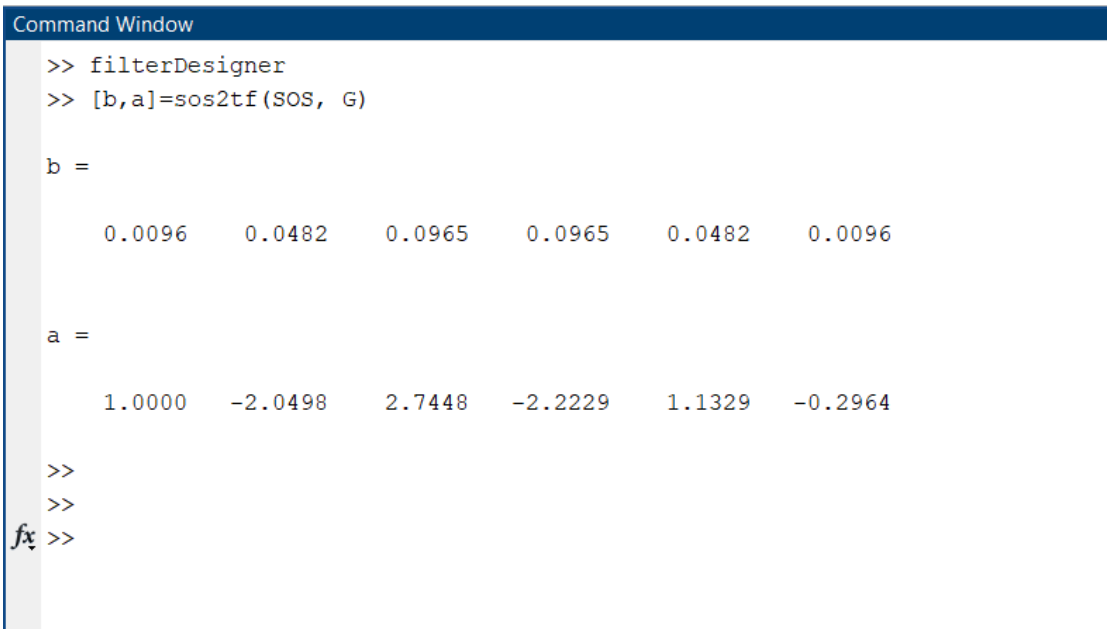
Kada se otvori alat za dizajniranje filtera u njemu postoje razne opcije, od odabira tipa filtra, do vrste filtra, odabira željenih frekvencija, pojačanja i ostalo...

Primjerice odabirom IIR filtra Čebiševljevom aproksimacijom tipa I, moguće je odabrati željeni red filtra, granične frekvencije, frekvenciju uzorkovanja, te pojačanja u propusnom i nepropusnom pojasu kao što se može vidjeti na slici 8.12.



Slika 8.12. Filter Designer u Matlabu

Odabirom željenih parametara te klikom na “Design Filter”, filter dizajner je kreirao potrebne nam filtarske koeficijente na temelju zadanih parametara.
Koeficijenti se zatim prikažu na radnoj površini kao što je prikazano na slici 8.13., te se mogu koristiti dalje u bilo kojem programskom alatu ili programskom jeziku.



Slika 8.13. Filtarski koeficijenti nastali Čebiševljevom aproksimacijom tipa I u Matlabu

8. ZAKLJUČAK

Digitalna obrada signala je neizbježan dio svakog uređaja koji se bavi bilo kakvim signalima, bilo to u telekomunikacijama, medicini, obradi zvuka ili govora te stvaranju glazbenih datoteka. Zbog sve veće dostupnosti računala digitalna obrada signala se počela sve masivnije pojavljivati te jedino što ograničava digitalni signal koji se obrađuje je njegova frekvencija uzorkovanja.
U današnje vrijeme gotovo svaki signal, u bilo kojoj upotrebi će vrlo vjerojatno biti obrađen na računalu.
Jedna od najvećih prednosti digitalne obrade signala je detekcija pogreške te njeno ispravljanje pri prijenosu, kao i digitalna kompresija.

U ovom radu je prikazan postupak dizajna filtera pomične srednje vrijednosti te digitalnih FIR i IIR filtera, ali kao što se vidi tema je iznimo složena i opsežna. FIR filteri su iznimno lagani za kreiranje, samo se primjeni postojeći algoritam na neki od programskih jezika te će filtar raditi.
IIR filter je dosta složeniji za implementaciju od FIR filtera jer je potrebno transformirati analogne filtere u digitalne, što je pokazano dosta složen proces. Srećom u moderno vrijeme većina dizajnera digitalnih filtera će koristiti odgovarajući programski alat za obradu signala te će on izbaciti potrebne koeficijente za filtraciju signala. Potrebno je samo napisati program koji će te koeficijente procesuirati i dati filtrirani signal. Neovisno o adekvatnim programskim alatima i dalje je potrebno znati određene pojmove vezane uz digitalnu obrada signala, te prednosti i mane željenog filtra, a to nijedan program ne može kompenzirati.
Od obrađenih vrsta filtera, filter pomične srednje vrijednosti je najlakši za dizajniranje jer mu je algoritam iznimno jednostavan, ali mu je uloga iznimno ograničena, samo na vremenske signale.

LITERATURA

[1] https://www.audiologs.com/ozrenbilan/sitemap.htm Pristupljeno: Lipanj 2023.

[2] https://www.audiologs.com/ozrenbilan/1_04_DSP.pdf Pristupljeno: Lipanj 2023.

[3] https://www.dspguide.com/ Pristupljeno: Lipanj 2023.

[4] https://www.gaussianwaves.com/2017/02/choosing-a-filter-fir-or-iir-understanding-the-design-perspective/ Pristupljeno: Lipanj 2023.

[5] https://thewolfsound.com/bilinear-transform/ Pristupljeno: Srpanj 2023.

[6] https://en.wikipedia.org/wiki/Prototype_filter Pristupljeno: Srpanj 2023.

[7] https://mrcet.com/downloads/digital_notes/ECE/III%20Year/DIGITAL%20SIGNAL%20PROCESSING.pdf Pristupljeno: Srpanj 2023.

[8] https://github.com/curiores/ArduinoTutorials/tree/main/BasicFilters Pristupljeno: Srpanj 2023.

[9] https://web.ece.ucsb.edu/~yoga/courses/Signals.html Pristupljeno: Srpanj 2023.

[10] https://www.mathworks.com/help/signal/ug/introduction-to-filter-designer.html Pristupljeno: Srpanj 2023.

[11] https://www.open.edu/openlearn/science-maths-technology/electronic-applications/content-section-3.7 Pristupljeno: Kolovoz 2023.

[12] https://www.allaboutcircuits.com/technical-articles/understanding-transfer-functions-for-low-pass-filters/ Kolovoz: 2023.

[13] https://users.sussex.ac.uk/~pjly20/ras100.html Pristupljeno: Kolovoz 2023.

[14] Mijat N., Električni filtri, Zagreb, 2001

[15] https://www.vyssotski.ch/BasicsOfInstrumentation/SpikeSorting/Design_of_FIR_Filters.pdf Pristupljeno: Kolovoz: 2023.

POPIS SLIKA

Slika 2.1 Analogni signal 3
Slika 2.2. Diskretni signal 4
Slika 2.3. Nastajanje diskretnog signala 5

Slika 3.1. Frekvencijska karakteristika nisko propusnog filtra 6
Slika 3.2. Frekvencijska karakteristika visoko propusnog filtra 6
Slika 3.3. Frekvencijska karakteristika pojasno propusnog filtra 7
Slika 3.4. Frekvencijska karakteristika pojasno nepropusnog filtra 7

Slika 4.1. Delta funkcija $\delta[n]$ 8
Slika 4.2. Impulsni odziv sustava 8
Slika 4.3. Prikaz konačnog impulsnog odziva 9
Slika 4.4. Beskonačni impulsni odziv 12

Slika 6.1. Frekvencijski odziv filtra pomične srednje vrijednosti 17
Slika 6.2. Izlazni signal pri $N = 6$ 18
Slika 6.3. Izlazni signal pri N=20 19

Slika 7.1. Blok shema FIR filtera 20
Slika 7.2. Idealna nekauzalna frekvencijska karakteristika nisko propusnog filtra 21
Slika 7.3. Prikaz diskretne sinc funkcije 22
Slika 7.4. Skraćena sinc funkcija 23
Slika 7.5. Frekvencijska karakteristika sinc funkcije 24
Slika 7.6. Frekvencijske arakteristike Blackmanovog i Hammingovog prozora 25
Slika 7.7. Frekvencijska karakteristika filter kernela nakon množenja s prozorom wn 26
Slika 7.8. Nisko propusni filter granične frekvencije 3000 Hz s duljinom kernela 128 28
Slika 7.9. Visoko propusni filter granične frekvencije 11000 Hz s duljinom kernela 64. 30
Slika 7.10. Pojasno nepropusni filter od 12000 Hz do 17000 Hz s duljinom kernela 80 32

Slika 7.11. Pojasno propusni filter od 500 Hz do 5000 Hz s duljinom kernela 88 34

Slika 8.1. Blok shema IIR filtera 36
Slika 8.2. Frekvencijsko savijanje – odnos analogne i digitalne frekvencije 37
Slika 8.3. Bilinearna transformacija – Mapiranje s domene u z domenu.....38
Slika 8.4. Nisko propusni filter pri graničnoj frekvenciji od 3000 Hz 41
Slika 8.5. Visoko propusni filter pri graničnoj frekvenciji od 12000 Hz 44
Slika 8.6. Pojasno propusni filter pri graničnim frekvencijama 500 Hz i 1000 Hz 48
Slika 8.7. Pojasno nepropusni filter pri graničnim frekvencijama 500 Hz i 1000 Hz.....52
Slika 8.8. Frekvencijska karakteristika Butterworthove aproksimacije 54
Slika 8.9. Frekvencijska karakteristika Čebiševljeve aproksimacija tipa I 54
Slika 8.10. Frekvencijska karakteristika Čebiševljeve aproksimacija tipa II 55
Slika 8.11. Frekvencijska karakteristika Eliptične aproksimacije 55
Slika 8.12. Filter Designer u Matlabu 56
Slika 8.13. Filtarski koeficijenti nastali Čebiševljevom aproksimacijom tipa I u Matlabu.....57

POPIS TABLICA

Tablica 5.1. Prikaz postupka konvolucije.....14

PRILOZI

PRILOG 1. Kod za FIR filter pomoću prozora

```
#define duljina_signala 320
#define duljina_kernela 89              //unesite željenu duljinu kernela,
što je veći broj prijelazni pojasi uži
#define granicna_frekvencija 3000       //unesite željenu frekvenciju ako
koristite lowpass ili high pass filter
#define donja_granicna  500              //unesite donju graničnu
frekvenciju za bandpass ili bandstop
#define gornja_granicna 5000             //unesite gornju graničnu
frekvenciju za bandpass ili bandstop
#define tip_filtera "lowpass"            //odaberite tip filtera "lowpass"
"highpass" "bandpass"  "bandstop"
#define frekvencija_uzorkovanja 48000    //frekvencija uzorkovanja

extern double ulazni_signal[duljina_signala];           //duljina ulazni
signal iz "New taba"
double filter_kernel[duljina_kernela];                  //filter kernel
double izlazni_signal[duljina_signala];                 // izlazni signal

void setup()
{
  SerialUSB.begin(9600);  //Serial.begin(9600) ako se ne koristi MKR Wifi
1010 ili Duo
  while (!SerialUSB) {}    //ukloni se ako se ne koristi MKR Wifi 1010 ili
Duo
  Serial.println("Ulazni_signal, Filtrirani_signal");

  if (tip_filtera == "lowpass" || tip_filtera == "highpass")    //za
izračun low pass ili high pass
filtera
  izracun_kernela_LP_HP (filter_kernel, duljina_kernela,
granicna_frekvencija, frekvencija_uzorkovanja);

  if (tip_filtera == "bandpass" || tip_filtera == "bandstop")   //za
izračun bandpass pass ili badnstop filtera
  izracun_kernela_BP_BS (filter_kernel, duljina_kernela, donja_granicna,
gornja_granicna, frekvencija_uzorkovanja);

  izracun_izlaza (ulazni_signal, izlazni_signal, filter_kernel,
duljina_signala, duljina_kernela);  //konvolucija kernela i ulaznog signala
  crtanje_ulaza_izlaza();   //crtanje izlaznih signala
  //crtanje_kernela ();    //filter kernel je 15 puta uvećan radi lakše
uočljivosti
}

void loop()
{
}

void izracun_kernela_BP_BS (double *kernel, int kernel_duljina, double
donja_granicna_f1, double gornja_granicna_f2, double frekv_uzorkovanja)
{
```

```
  int n;
  double f_spremnik, f1_norm, f2_norm, kernel_f1[kernel_duljina],
kernel_f2[kernel_duljina], spremnik;
  f1_norm=(donja_granicna_f1)/(frekv_uzorkovanja);  //donja granicna
frekvencija
  f2_norm=(gornja_granicna_f2)/(frekv_uzorkovanja);  // gornja granicna
frekvencija
  int brojac;

 for (brojac=1;brojac<=2;brojac++)       //brojac == 1 za low pass kernel,
brojac == 2 za high pass kernel
 {
  if (brojac == 1)
  f_spremnik=f1_norm;

  if (brojac == 2)
  f_spremnik=f2_norm;

  for (n=0;n<(kernel_duljina/2);n++)     //izraćun sinc funkcije, odnosno
izračun low pass i high pass kernela
  {
    if (n == 0)
    {
      spremnik=2*f_spremnik;
      kernel[(kernel_duljina/2)+n]=spremnik;
    }
   else
    {
      spremnik=sin(2*3.14*f_spremnik*(n))/((3.14)*(n));
      kernel[(kernel_duljina/2)+n]=spremnik;
      kernel[(kernel_duljina/2)-n]=spremnik;
    }
  }

   for (n=0;n<kernel_duljina;n++)   //izraćun kernela blackmanovim prozorom
   {
    kernel[n]=kernel[n]*((0.42)-
(0.5*cos((2*3.14*n)/kernel_duljina))+(0.08*cos((4*3.14*n)/kernel_duljina)))
;  //izraćun kernela blackmanovim prozorom
   }

   double suma=0;                  //normalizacija pojačanja, ako ovog nema
signal će samo biti uvećan
   for(n=0;n<kernel_duljina;n++)
   suma=suma+kernel[n];

   for(n=0;n<kernel_duljina;n++)
   {
    kernel[n]=(kernel[n]/suma);

    if (brojac == 1)
    kernel_f1[n]=kernel[n];

    if (brojac == 2)
    kernel_f2[n]=kernel[n];
   }
```

```
  }
  for(n=0;n<kernel_duljina;n++)
  {kernel_f2[n]=-(kernel_f2[n]);}         //pretvaranje drugog low pass
kernela gornje granične frekvencije u high pass kernel preko spektralne
inverzije
  kernel_f2[kernel_duljina/2]=kernel_f2[kernel_duljina/2]+1;

  for(n=0;n<kernel_duljina;n++)
  kernel[n]=kernel_f1[n]+kernel_f2[n];      //stvaranje band stop kernela
kao zbroj prvog low pass kernela i high pass kernela


  if (tip_filtera == "bandpass")
  {
    for(n=0;n<kernel_duljina;n++)   //stvaranje band pass kernela
spektralnom inverzijom band stop kernela
    {
     kernel[n]=-(kernel[n]);
    }
     kernel[kernel_duljina/2]=kernel[kernel_duljina/2]+1;
  }
}

void izracun_kernela_LP_HP (double *kernel, int kernel_duljina, double
granicna_frekv, double frekv_uzorkovanja)
{

 int n;
 double f_norm;
 double spremnik;
 f_norm=(granicna_frekv/frekv_uzorkovanja);    //granicna frekvencija

  for (n=0;n<(kernel_duljina/2);n++)                 //izračun low pass ili
high pass kernela
  {
    if (n == 0)
    {
      spremnik=2*f_norm;
      kernel[(kernel_duljina/2)+n]=spremnik;
    }
     else
    {
      spremnik=sin(2*3.14*f_norm*(n))/((3.14)*(n));
      kernel[(kernel_duljina/2)+n]=spremnik;
      kernel[(kernel_duljina/2)-n]=spremnik;
    }
  }
  for (n=0;n<kernel_duljina;n++)   //izraćun kernela blackmanovim prozorom
  {
   kernel[n]=kernel[n]*((0.42)-
(0.5*cos((2*3.14*n)/kernel_duljina))+(0.08*cos((4*3.14*n)/kernel_duljina)))
;
  }

  double suma=0;                    //normalizacija pojačanja, ako ovog
nema signal će samo biti uvećan
```

```
  for(n=0;n<kernel_duljina;n++)
  suma=suma+kernel[n];
  for(n=0;n<kernel_duljina;n++)
  kernel[n]=(kernel[n]/suma);
  if (tip_filtera == "highpass")     //spektralna inverzija low pass
filtera ako se traži high pass filter
  {
    for(n=0;n<kernel_duljina;n++)
    {
     kernel[n]=-(kernel[n]);
    }
     kernel[kernel_duljina/2]=kernel[kernel_duljina/2]+1;
  }
}
void izracun_izlaza (double *ulaz, double *izlaz, double *kernel, int
signal_duljina, int kernel_duljina)
{
  int n,k;
   for (n=0;n<signal_duljina;n++)
   {
    izlaz[n]=0;
   }

    for(n=kernel_duljina;n<signal_duljina;n++)
    {
      for(k=0;k<kernel_duljina;k++)
      {
       izlaz[n] = izlaz[n]+(ulaz[n-k]*kernel[k]);   //cijela funkcija
provodi konvoluciju filter kernela i ulaznog signala
      }
    }
}
void crtanje_ulaza_izlaza(void)     //iscrtavanje ulaznih i izlaznih
signala
{
 int n;
 for(n=0;n<duljina_signala;n++)
 {
  Serial.print(ulazni_signal[n]);
  Serial.print(",");
  Serial.println(izlazni_signal[n]);
  delay(10);
 }
}
void crtanje_kernela(void)     //iscrtavanje filter kernela, po potrebi se
koristi
{
 int n;
 for(n=0;n<duljina_kernela;n++)
 {
  Serial.println(filter_kernel[n]);     //filter kernel je 15 puta uvećan
radi lakše uočljivosti
  delay(10);
 }
}
```

PRILOG 2. Kod za IIR filter

```
#define duljina_signala 320                    //duljina ulaznog signala
#define frekvencija_uzorkovanja 48000          //frekvencija uzorkovanja
#define granicna_frekvencija 12000             //granicna frekvencija za low
pass i high pass filter
#define donja_granicna 500                     //donja granicna frekvencija
za bandpass i bandstop filter
#define gornja_granicna 2000                   //gornja granicna frekvencija
za bandpass i bandstop filter
#define tip_filtra "highpass"                  //za odabir unesite "lowpass"
"highpass"  "bandpass" i "bandstop"

extern double ulazni_signal[duljina_signala];   //deklaracija ulaznog
signala iz drugog taba
double izlazni_signal[duljina_signala];          //izlazni signal

void setup() {

  SerialUSB.begin(9600);  //Serial.begin(9600) ako se ne koristi MKR Wifi
1010 ili Duo

  while (!SerialUSB) {}   //ukloni se ako se ne koristi MKR Wifi 1010 ili
Duo
  Serial.println("Ulazni_signal, Filtrirani_signal");

  if(tip_filtra == "lowpass")   //funkcija za izračuna low pass filtera
  {rekurzija_LP (ulazni_signal, izlazni_signal , duljina_signala,
granicna_frekvencija, frekvencija_uzorkovanja);}

  if(tip_filtra == "highpass")   //funkcija za izračuna high pass filtera
  {rekurzija_HP (ulazni_signal, izlazni_signal , duljina_signala,
granicna_frekvencija, frekvencija_uzorkovanja);}

  if(tip_filtra == "bandpass")   //funkcija za izračuna band pass filtera
  {rekurzija_BP (ulazni_signal, izlazni_signal , duljina_signala,
donja_granicna, gornja_granicna, frekvencija_uzorkovanja);}

  if(tip_filtra == "bandstop")   //funkcija za izračuna band stop filtera
  {rekurzija_BS (ulazni_signal, izlazni_signal , duljina_signala,
donja_granicna, gornja_granicna, frekvencija_uzorkovanja);}

   crtanje_signala (ulazni_signal, izlazni_signal ,
duljina_signala);  //funkcija za iscrtavanje ulaznog i izlaznog signala
}

void loop()
{
}

void rekurzija_BP (double *ulaz, double *izlaz, int signal_duljina, double
f_donja, double f_gornja, double frekv_uzorkovanja)
{
double T, w_d, w_g, w0, B, Q, alfa, alfa2, den_koef;
T=1/(frekv_uzorkovanja);  //izračun perioda
w_d=(2/T)*tan((2*PI*f_donja*T)/2);  //donja granicna frekvencija
```

```
w_g=(2/T)*tan((2*PI*f_gornja*T)/2);  //gornja granicna frekvencija
w0=sqrt(w_d*w_g);   //omega 0
B=w_g-w_d;          //delta omega
Q=w0/B;             //faktor Q
alfa=w0*T;           //radi skraćivanja izraza
alfa2=powf(alfa,2);   //radi skraćivanja izraza
den_koef=(4+(2*alfa/Q)+alfa2);  //nazivnik koeficijenata radi lakšeg
izračuna

double b0, b1, b2, a1, a2;     //deklaracija koeficijenata za njihov
izračun
b0=((2*alfa/Q)/den_koef);
b1=0;
b2=-(2*alfa/Q)/den_koef;
a1=((2*alfa2)-8)/den_koef;
a2=(4-(2*alfa/Q)+alfa2)/den_koef;

int i;
  for (i=0;i<signal_duljina;i++)
  {
    izlaz[i]=0;
  }
  int n;
  for (n=2;n<signal_duljina;n++)
  {
  izlaz[n]=(b0*ulaz[n])+(b1*ulaz[n-1])+(b2*ulaz[n-2])-(a1*izlaz[n-1])-
(a2*izlaz[n-2]);  //rekurzijska jednadžba
  }
}

void rekurzija_LP (double *ulaz, double *izlaz, int signal_duljina, double
f_granicna, double frekv_uzorkovanja)
{
double T, w, den_koef;
T=1/(frekv_uzorkovanja);    //izračun perioda
w=(2/T)*tan((2*PI*f_granicna*T)/2);     //granicna frekvencija
den_koef=(2+(w*T));    //nazivnik koeficijenata radi lakšeg izračuna

double b0, b1, a1;       //deklaracija koeficijenata za njihov izračun
b0=((w*T)/den_koef);
b1=((w*T)/den_koef);
a1=((w*T)-2)/den_koef;

int i;
  for (i=0;i<signal_duljina;i++)
  {
    izlaz[i]=0;
  }
  int n;
  for (n=1;n<signal_duljina;n++)
  {
  izlaz[n]=(b0*ulaz[n])+(b1*ulaz[n-1])-(a1*izlaz[n-1]);   //rekurzijska
jednadžba
  }
}
```

```
void rekurzija_HP (double *ulaz, double *izlaz, int signal_duljina, double
f_granicna, double frekv_uzorkovanja)
{
double T, w, den_koef;
T=1/(frekv_uzorkovanja);    //izracun perioda
w=(2/T)*tan((2*PI*f_granicna*T)/2);  //granicna frekvencija
den_koef=(2+(w*T));   //nazivnik koeficijenata radi lakšeg izračuna

double b0, b1, a1;  //deklaracija koeficijenata za njihov izračun
b0=(2/den_koef);
b1=(-2/den_koef);
a1=((w*T)-2)/den_koef;

int i;
  for (i=0;i<signal_duljina;i++)
  {
    izlaz[i]=0;
  }
  int n;
  for (n=1;n<signal_duljina;n++)
  {
   izlaz[n]=(b0*ulaz[n])+(b1*ulaz[n-1])-(a1*izlaz[n-1]);    //rekurzijska
jednadžba
  }
}

void rekurzija_BS (double *ulaz, double *izlaz, int signal_duljina, double
f_donja, double f_gornja, double frekv_uzorkovanja)
{
double T, w_d, w_g, w0, B, Q, alfa, alfa2, den_koef;
T=1/(frekv_uzorkovanja);     //izracun perioda
w_d=(2/T)*tan((2*PI*f_donja*T)/2);  //donja granicna frekvencija
w_g=(2/T)*tan((2*PI*f_gornja*T)/2); //gornja granicna frekvencija
w0=sqrt(w_d*w_g);   //omega0
B=w_g-w_d;          //delta omega
Q=w0/B;             //faktor Q
alfa=w0*T;          //radi skraćivanja izraza
alfa2=powf(alfa,2);  //radi skraćivanja izraza
den_koef=(4+(2*alfa/Q)+alfa2); //nazivnik koeficijenata radi lakšeg
izračuna

double b0, b1, b2, a1, a2;      //deklaracija koeficijenata za njihov
izračun
b0=((4+alfa2)/den_koef);
b1=((2*alfa2)-8)/den_koef;
b2=((4+alfa2)/den_koef);
a1=((2*alfa2)-8)/den_koef;
a2=(4-(2*alfa/Q)+alfa2)/den_koef;

  int i;
  for (i=0;i<signal_duljina;i++)
  {
    izlaz[i]=0;
  }
  int n;
  for (n=2;n<signal_duljina;n++)
```

```
  {
   izlaz[n]=(b0*ulaz[n])+(b1*ulaz[n-1])+(b2*ulaz[n-2])-(a1*izlaz[n-1])-
(a2*izlaz[n-2]);  //rekurzijska jednadžba
  }
}

void crtanje_signala(double *ulaz, double *izlaz, int signal_duljina)
{
  int n;
  for(n=0;n<signal_duljina;n++)
  {
    Serial.print(ulaz[n]);
    Serial.print(",");
    Serial.println(izlaz[n]);
  }
}
```

PRILOG 3. Kod za unos vanjskih FIR koeficijenata

```
#define duljina_signala 320  //duljina signala
#define duljina_b  81         //unese se duljina b koeficijenata

extern double ulazni_signal[duljina_signala];    //ulazni signal
extern double b[duljina_b];                      //koeficijenti b iz drugog
"taba"
double izlazni_signal[duljina_signala];           //izlazni signal

void setup()
{
  SerialUSB.begin(9600);  //Serial.begin(9600) ako se ne koristi MKR Wifi
1010 ili Duo

  while (!SerialUSB) {}   //ukloni se ako se ne koristi MKR Wifi 1010 ili
Duo
  filtriranje (b, ulazni_signal, izlazni_signal);     //funkcija za
filtriranje FIR s b koeficijentima
  crtanje_signala();            //iscrtavanje ulaznog i izlaznog signala
}

void loop()
{
}

void filtriranje (double *koef_b,  double *ulaz, double *izlaz)
{
    double bcc;
    int n,k;
    for (int i=0; i<duljina_signala; i++)
    {
        izlaz[i] = 0;
    }
    for (n =duljina_b-1 ; n<duljina_signala; n++) {

        bcc = 0;     //akumulator(spremnik) pri izračunu konvolucije

        for (k = 0; k<duljina_b; k++)
        {
            bcc += koef_b[k] * ulaz[n-k];     //Izračun konvolucije
        }
        izlaz[n] = bcc;
    }
}
void crtanje_signala(void)
{
  int i;
  for(i=0;i<duljina_signala;i++){
  Serial.print(ulazni_signal[i]);
  Serial.print(",");
  Serial.println(izlazni_signal[i]);
  delay(20);

  }
}
```

PRILOG 4. Kod za unos vanjskih IIR koeficijenata

```
#define duljina_signala 320  //duljina signala
#define duljina_a_b  7         //unese se duljina b koeficijenata

extern double ulazni_signal[duljina_signala];    //ulazni signal
extern double b[duljina_a_b];              //koeficijenti b iz drugog
"taba"
extern double a[duljina_a_b];              //koeficijenti a iz drugog
"taba"
double izlazni_signal[duljina_signala];           //izlazni signal

void setup()
{
  SerialUSB.begin(9600);    //Serial.begin(9600) ako se ne koristi MKR Wifi
1010 ili Duo

  while (!SerialUSB) {}   //ukloni se ako se ne koristi MKR Wifi 1010 ili
Duo
  filtriranje (b, a, ulazni_signal, izlazni_signal);     //funkcija za
filtriranje IIR s a i b koeficijentima
  crtanje_signala();            //iscrtavanje ulaznog i izlaznog signala
}
void loop()
{
}
void filtriranje (double *koef_b, double *koef_a, double *ulaz, double
*izlaz)
{
    double bcc, acc;
    int n,k;
    for (int i=0; i<duljina_signala; i++)
    {
        izlaz[i] = 0;         //incijaliziranje izlaza na 0
    }

    for (n =duljina_a_b-1 ; n<duljina_signala; n++) {

        bcc = 0; //akumulator(spremnik) pri izračunu konvolucije
        acc=0; //akumulator(spremnik) pri izračunu rekurzije

        for (k = 0; k<duljina_a_b; k++)
        {
            bcc += koef_b[k] * ulaz[n-k];       //izračun konvolucije
        }
        for (k = 1; k<duljina_a_b; k++)
        {
            acc += (-1)*koef_a[k] * izlaz[n-k];   //izračun rekurzije
        }
        izlaz[n] = bcc+acc;   //suma od konvolucije i rekurzije
    }
}
void crtanje_signala(void)
{
  int i;
  for(i=0;i<duljina_signala;i++){
```

```
  Serial.print(ulazni_signal[i]);
  Serial.print(",");
  Serial.println(izlazni_signal[i]);
  delay(20);

  }
}
```

PRILOG 5. Kod za Filter pomične srednje vrijednosti

```
#define duljina_signala 320  //duljina signala
#define N 20          //broj uzoraka za srednju vrijednost

extern double ulazni_signal[duljina_signala]; //ulazni signal
double izlazni_signal[duljina_signala];       //izlazni signal

void setup() {
 SerialUSB.begin(9600);   //Serial.begin(9600) ako se ne koristi MKR Wifi
1010 ili Duo
 while (!SerialUSB) {}    //ukloni se ako se ne koristi MKR Wifi 1010 ili
Duo
 Serial.println("Ulazni_signal,Filtrirani_signal");

recursive_moving_average(ulazni_signal, izlazni_signal ,duljina_signala,
N);  //pozivanje filtera pomične srednje vrijednosti
 plotaj_2_signala (ulazni_signal, izlazni_signal
,duljina_signala);      //crtanje signala
}

void loop()
{
}
void plotaj_2_signala (double *ulaz ,double *izlaz ,int signal_duljina)
{
  int i;
  for (i=0;i<signal_duljina;i++)          //crtanje signala
  {
    Serial.print(ulaz[i]);
    Serial.print(",");
    Serial.println(izlaz[i]);
  }
}

void recursive_moving_average(double *ulaz, double *izlaz, int
signal_duljina, int filter_pts)
{
  int i,j;
  double spremnik;
  spremnik=0;
    for(i=0;i<filter_pts;i++)
    {
      spremnik = spremnik + ulaz[i];
    }

     izlaz[((filter_pts-1)/2)]  = spremnik/filter_pts;

      for(i= ceil(filter_pts/2); i<signal_duljina-ceil(filter_pts/2);
i++)                            //  p=(filter_pts-1)/2  q=p+1
      {
         spremnik =spremnik + ulaz[i+(filter_pts-1)/2]- ulaz[i-
(uint32_t)ceil(filter_pts/2)];                  //y[n]=y[n-1]+x[n+p]-x[n-q]
         izlaz[i]  = spremnik/filter_pts;
      }
}
```