5/23/2017

****

**Курсов проект по Комуникационни вериги на тема: „Проектиране на филтри“**

**Изготивил**:

**Специалност**: Телекомуникации

**Курс:** 2

**Група**:

**Ръководител:**......................................

(Златка Вълкова – Джарвис)

**Съдържание:**

*Утежняване на габарита на ЛФ ....................................стр.2*

*Преобразуване на ЛФ в РФ...............................................стр.3*

*Апроксимация.....................................................................стр.4*

*Криви на затихване...........................................................стр.5*

*ПНД и ФЧХ........................................................................стр.6*

*Схемна реализация............................................................стр.8*

*Филтриране на синусоидални сигнали............................стр.10*

*Програмен код...................................................................стр.15*

*Списък с използвани съкращения....................................стр.19*

*Списък с използвана литература...................................стр.20*

**Утежняване на габарита на лентовия филтър**

За да бъде апроксимиран правилно лентовият филтър трябва да „утежним“ габарита му. За тази цел трябва затихването в двете ленти на задържане да бъде еднакво. По-долу може да видите габарита на зададения лентов филтър и след това на утежнения.

**Преобразуване от лентов към режекторен филтър**

Лентовият филтър трябва да го превърнем в режекторен. Габарита на режекторния филтър ще изглежда по следния начин:

Полученият режекторен филтър го представяме като ниско- и високочестотен филтър:

**Апроксимиране на ниско- и високочестотен филтър**

За да апроксимираме правилно двата филтъра, които, свързани паралелно, образуват режекторен, ще използваме апроксимацията на Бътъруърд, тъй като тя е и максимално-плоска, и полиномна класическа апроксимация.

Използваме следните команди за двата филтъра:

1. [N,Wn]=buttord(Wp,Ws,Rp,Rs)- *връща най-ниския възможен ред N с гранична честота Wp*

2. [Nz,Dz]=butter(N,Wn)- *връща полином в числителя и полином в знаменателя*

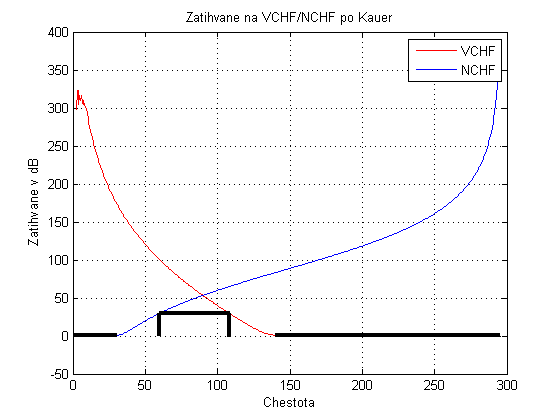
Получаваме следните предавателни функции съответно за нискочестотния и високочестотния филтър:

HLP(z)= *:* 1.0е-003 \* ;

HHP(z)=

**Криви на затихване**

На следващата фигура са изобразени кривите на затихване на нискочестотния и високочестотния филтър заедно с габарита на режекторния филтър:



За граничните честоти на нискочестотния и високочестотния филтър получихме съответно Wn1=0.1017 и Wn=0.4537.

За да се проследят стойностите на граничните честоти след апроксимацията трябва да ги денормираме горните и получаваме:

Wp1=30.0015 Hz и Wp=133.8415 Hz

Използвани команди:

1. [H,w]=freqz(Nz,Dz)-*изчислява предавателната функция и кръговата честота*
2. m=abs(H) - *изчислява амплитудно-честотната характеристика в линейни единици*
3. md=-20\*log10(m) - *изчислява затихването в dB*
4. figure(1) - *нова фигура*
5. plot(w1\*Fs/(2\*pi),md) - *чертае затихването на нискочестотния филтър*
6. hold on - *задържане на фигура, с цел наслагване на друга*

За изобразяване на габаритите на режекторния филтър използваме командите:

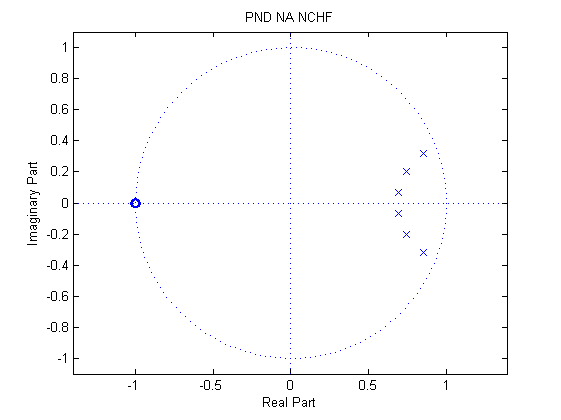
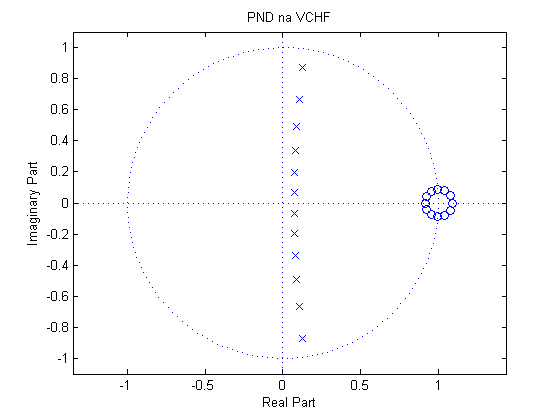
x=linspace(x1,x2), *като в скобите записваме стойностите на ЛЗ/ЛП по* ***х***

y=linspace(y1,y2), *като в скобите записваме стойностите на ЛЗ/ЛП по* ***у***

line(x,y,’Color’,’black’, ‘LineWidth’,3) – *чертаем габарита на режекторния филтър в* ***черен цвят***

**Полюсно-нулеви диаграми и фазово-честотни характеристики**

С помощтна на командата zplane(Nz,Dz) чертаем полюсно-нулеви диаграми съответно на нискочестотния и високочестотния филтър:

С командата [z,p,k]=butter(N,Wn) “Matlab” извежда следните числени стойности на полюсите и нулите на предавателните функции :

- **нискочестотен филтър**:

**Полюси:** 0.6925 + 0.0691i **Нули:** -1

0.6925 - 0.0691i -1

0.7439 + 0.2027i -1

0.7439 - 0.2027i -1

0.8536 + 0.3178i -1

0.8536 - 0.3178i -1

**- високочестотен филтър**:

**Полюси:** 0.1283 + 0.8688i **Нули:** 1

0.1283 - 0.8688i 1

0.1051 + 0.6631i 1

0.1051 - 0.6631i 1

0.0904 + 0.4899i 1

0.0904 - 0.4899i 1

0.0731 + 0.0652i 1

0.0731 - 0.0652i 1

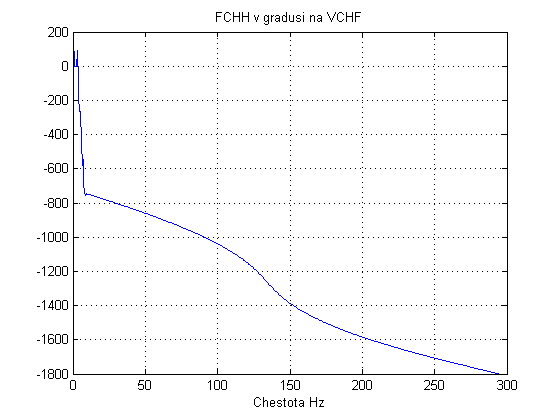
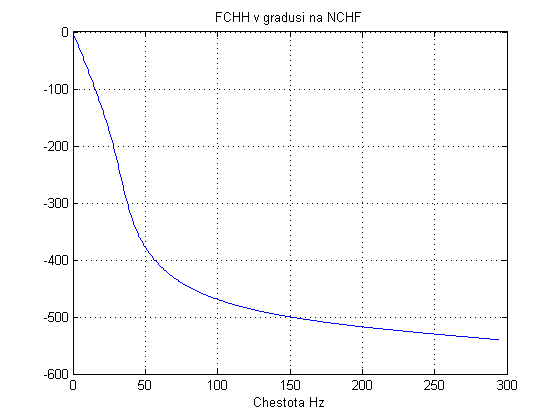
0.0757 + 0.1978i 1

0.0757 - 0.1978i 1

0.0811 + 0.3374i 1

0.0811 - 0.3374i 1

Фазово-честотни характеристики съответно на нискочестотния и на високочестотния филтър:



Команди:

1.fi=unwrap(angle(H)) – *изчислява ФЧХ в rad*

2.plot(w\*Fs/(2\*pi),fi\*180/pi) - *изчертаване на ФЧХ в градуси, с логаритмична скала на кръговата честота.*

**Схема на режекторния филтър**

Нека първо да представим схемите на нискочестотния и високочестотния филтър, които после ще свържем паралелно, за да се получи режекторен филтър. Преди това обаче ще трябва да преобразуваме предавателните функции на филтрите.

Сега с функцията [b0,B,A]=dir2cas(Nz,Dz) ще преобразуваме директната реализация в каскадна. Каскадно разложените предавателни функции са, както следва:

HLP(z)=

HHP(z)=

Схемите на двата филтъра ще изглеждат така:

Схема на режекторния филтър, получен чрез паралелно свързване на НЧФ и ВЧФ:

**Филтриране на синусоидални сигнали**

Случайно избрани сигнали трябва да преминат през лентовия филтър, единият от които трябва да бъде пропуснат, а другия – не. За целта избираме произволна честота от лентата на пропускане и от лентата на задържане, с които ще работим.

Работим със следните команди:

К=590 – *брой точки за бързо преобразувание на Фурие;*

n=0:1/590:1– *определяне на времева област;*

x=sin(2.\*pi\*x.\*n) – *въвеждане на входен сигнал* ***х;***

y=filter(Nz,Dz,x) – *изчисляване на изходния сигнал след филтриране на входния;*

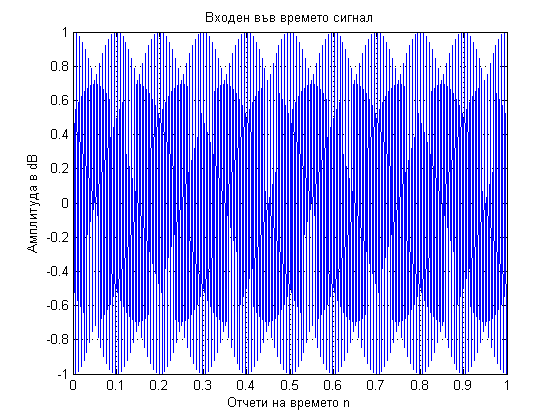
Px=fft(x,K1) – *изчисляване не бързото преобразувание на Фурие на входен сигнал* ***х*** *в* ***К*** *точки;*

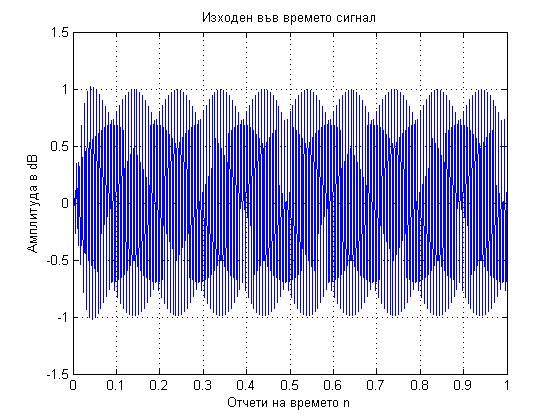
Px=fft(x,K1) - *изчисляване не бързото преобразувание на Фурие на изходен сигнал* ***х*** *в* ***К*** *точки;*

px=abs(Px(1:(K1/2))) – *изчислява амплитудния спектър;*

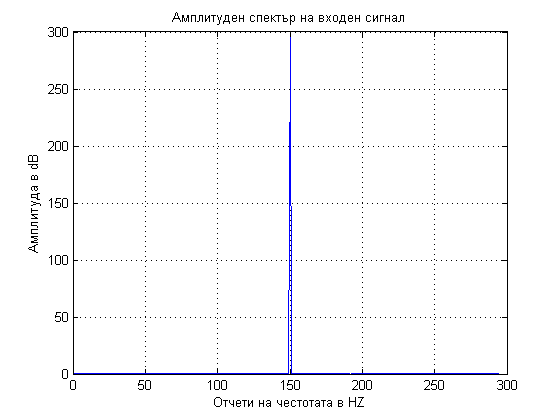
Сигнал с честота **150 Hz** (*x=sin(2\*pi\*440\*n*) съответно във времева и честотна област, на входа и на изхода:

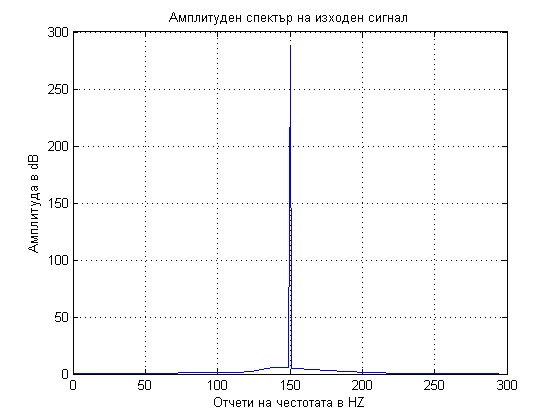
***Времева област***

**

**

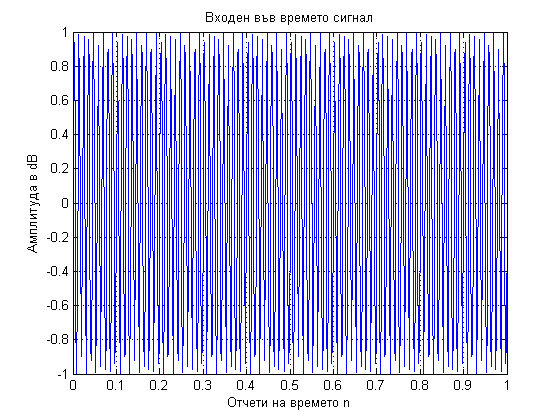
***Честотна област***

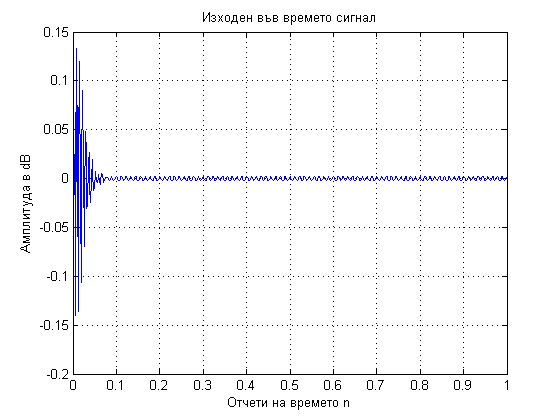
**

**

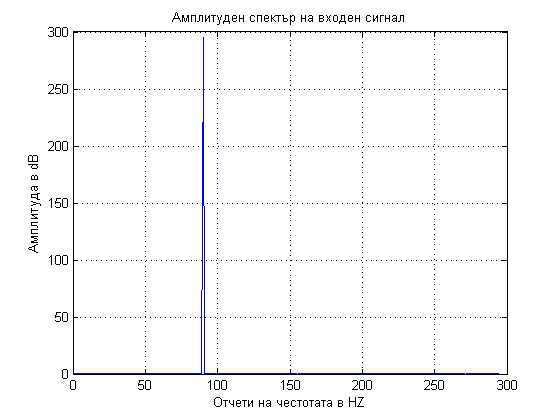
Сигнал с честота **90** Hz (*x=sin(2\*pi\*90\*n*) съответно във времева и честотна област, на входа и на изхода:

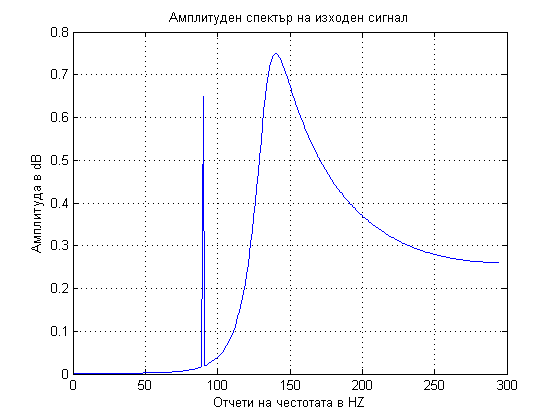
***Времева област***

**

**

***Честотна област***

**

**

От графиките се вижда, че сигналът с честота 150 Hz е затихнат незначително. Този с честота 90 Hz е филтриран, тъй като неговата амплитуда на изхода спрямо тази на входа е намалена много повече от 2 пъти.

Програмен код

clear all

close all

clc

Fn=295;

Fs=590;

Rp=1;Rs=30;

Wp=140/Fn;

Ws=108/Fn;

[N,Wn]=buttord(Wp,Ws,Rp,Rs); %VCHF

[Nz,Dz]=butter(N,Wn,'high')

[z,p,k]=butter(N,Wn,'high')

Wp1=30/Fn;

Ws1=60/Fn;

[N1,Wn1]=buttord(Wp1,Ws1,Rp,Rs); %NCHF

[Nz1,Dz1]=butter(N1,Wn1)

[z,p,k]=butter(N1,Wn1)

figure(1);

zplane(Nz,Dz);

title('PND na VCHF');

figure(2);

zplane(Nz1,Dz1);

title('PND NA NCHF');

figure (2);

[H,w]=freqz(Nz,Dz);

m=abs(H);

md=-20\*log10(m);

fi=unwrap(angle(H));

plot(w\*Fs/(2\*pi),md,'r');

hold on;

[H1,w1]=freqz(Nz1,Dz1);

m1=abs(H1);

md1=-20\*log10(H1);

fi1=unwrap(angle(H1));

plot(w1\*Fs/(2\*pi),md1);

xlabel('Chestota');

ylabel('Zatihvane v dB');

title('Zatihvane na VCHF/NCHF po Kauer');

legend('VCHF','NCHF','--');

grid;

hold on;

x=linspace(0,30);

y=linspace(1,1);

line(x,y,'Color','black','LineWidth',3);

x=linspace(60,60);

y=linspace(1,30);

line(x,y,'Color','black','LineWidth',3);

x=linspace(60,108);

y=linspace(30,30);

line(x,y,'Color','black','LineWidth',3);

x=linspace(108,108);

y=linspace(30,0);

line(x,y,'Color','black','LineWidth',3);

x=linspace(140,295);

y=linspace(1,1);

line(x,y,'Color','black','LineWidth',3);

figure(3) %FCHH VCHF

plot(w\*Fs/(2\*pi),fi\*180/pi);

grid;

title('FCHH v gradusi na VCHF');

xlabel('Chestota Hz');

figure (4) %FCHH NA NCHF

plot(w1\*Fs/(2\*pi),fi1\*180/pi);

grid;

title('FCHH v gradusi na NCHF');

xlabel('Chestota Hz');

b=[0.0021 -0.0250 0.1375 -0.4584 1.0314 -1.6502 1.9253 -1.6502 1.0314 -0.4584 0.1375 -0.0250 0.0021];

a=[1 -1.1072 2.1514 -1.5788 1.4434 -0.7120 0.3756 -0.1207 0.0371 -0.0071 0.0011 -0.0001];

[b0,B,A]=dirtocas(b,a)

b1=[0.0968 0.1070 0.1956 0.1956 0.1070 0.0968];

a1=[1 -1.5137 2.2362 -1.5777 0.8688 -0.2148];

[b01,B1,A1]=dirtocas(b1,a1)

n=0:1/590:1

x=sin(2\*pi\*150\*n)

y=filter(Nz,Dz,x)

figure(5);

grid;

xlabel('Отчети на времето n');

ylabel('Амплитуда в dB');

title('Входен във времето сигнал');

figure(6);

plot(n,y)

grid;

xlabel('Отчети на времето n');

ylabel('Амплитуда в dB');

title('Изходен във времето сигнал');

x1=sin(2\*pi\*90\*n)

y1=filter(Nz,Dz,x1)

figure(7);

plot(n,x1)

grid;

xlabel('Отчети на времето n');

ylabel('Амплитуда в dB');

title('Входен във времето сигнал');

figure(8);

plot(n,y1)

grid;

xlabel('Отчети на времето n');

ylabel('Амплитуда в dB');

title('Изходен във времето сигнал');

K=590

[H,w]=freqz(Nz,Dz,K/2)

[H1,w1]=freqz(Nz1,Dz1,K/2)

Px=fft(x,K)

px=abs(Px(1:(K/2)))

Py=fft(y,K)

py=abs(Py(1:(K/2)))

figure(9)

plot(w\*Fs/(2\*pi),px)

xlabel('Отчети на честотата в HZ');

ylabel('Амплитуда в dB');

title('Амплитуден спектър на входен сигнал');

grid;

figure(10)

plot(w\*Fs/(2\*pi),py)

xlabel('Отчети на честотата в HZ');

ylabel('Амплитуда в dB');

title('Амплитуден спектър на изходен сигнал');

grid;

Px1=fft(x1,K);

px1=abs(Px1(1:(K/2)))

figure(11)

Px1=fft(x1,K)

px1=abs(Px1(1:(K/2)))

plot(w\*Fs/(2\*pi),px1)

xlabel('Отчети на честотата в HZ');

ylabel('Амплитуда в dB');

title('Амплитуден спектър на входен сигнал');

grid;

figure(12)

Py1=fft(y1,K)

py1=abs(Py1(1:(K/2)))

plot(w\*Fs/(2\*pi),py1)

xlabel('Отчети на честотата в HZ');

ylabel('Амплитуда в dB');

title('Амплитуден спектър на изходен сигнал');

grid;

**Списък с използвани съкращения:**

* ПНД – полюсно-нулева диаграма
* НЧФ – нискочестотен филтър
* ВЧФ – високочестотен филтър
* ЛФ – лентов филтър
* РФ – режекторен филтър
* ФЧХ – фазово-честотна характеристика

**Използвана литературна:**

1. З.Вълкова-Джарвис, К. Николова. Комуникационни вериги - лабораторни упражнения, изд.“ЕКС-Либрис ПК“, София, 2009.

2. З.Вълкова-Джарвис, Курсов проект по дисциплината „Комуникационни вериги“, Март, 2017

**Използваният програмен продукт е Matlab, R2011b.**