**Комуникационни вериги**

Курсов проект

Тема:“Проектиране на филтри“

**Студент:** **Факултет:** Факултет по телекомуникации

**Специалност:** Телекомуникации

**Курс:** 2

**Група:**

**Факултетен №:**

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

/доц.д-р Златка Вълкова-Джарвис/

**Съдържание:**

*Утежняване на габарита на зададения цифров лентов филтър* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 3

*Разлагане на лентов филтър на съставящите го нискочестотен и високочестотен филтър и техните габарити* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_4

*Апроксимиране на получените нискочестотен и високочестотен филтър.Предавателни функции.* \_\_\_\_4

*Криви на затихване на нискочестотен и високочестотен, заедно с габарита на лентов филтър.Честоти на задържане.*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_7

*Полюсно-нулеви диаграми и фазово-честотни характеристики на нискочестотен и високочестотен филтър.* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_9

*Схема на цифров филтър, съставен чрез нискочестотен и високочестотен филтър, реализирани по паралелна реализация.*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_11

*Филтриране на синусоидални сигнали.*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_13

*Списък с използвани съкращения.*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_19

*Съставен код на Matlab*.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*20*

*Използвани литературни източници.*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_24

***Утежняване на габарита на зададения цифров лентов 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 се виждат габаритите на зададения лентов филтър и на вече утежнения:

***Разлагане на лентов филтър на съставящите го нискочестотен и високочестотен филтър и техните габарити***

***Апроксимиране на получените нискочестотен и високочестотен филтър.Предавателни функции.***

По задание трябва цифровият лентов филтър да бъде апроксимиран с равновълнова неполиномна класическа апроксимация.

Равновълнов характер в лентата на пропускане имат апроксимациите по Кауер и Чебишев. От друга страна неполиномни са апрокцимациите по Кауер и инверсна по Чебишев. Най-подходящата апроксимация за нашия филтър е тази по Кауер, тъй като е неполиномна от една страна и с равновълнов характер в лентата на пропускане от друга страна.

При нискочестотния филтър имаме гранична честота Wp1=920 Hz при затихване 0.5 dB и честота на задържане Ws 1=1040 Hz при затихване 30 dB.

Дадена ни е честота на дискретизация Fs=4.42 kHz. За да бъде апроксимиран нискочестотния филтър честотите Wp1 и Ws1 трябва да бъдат нормирани. Нормирането се изисква, тъй като цифровите филтри съществуват в честотния диапазон от 0 до честотата на Найкуист – Fs/2, която е половината от честотата на дискретизация. Нормирането се извършва именно с тази честота, която в нашия случай е *: Fn= 4 420/2=2 210 Hz .*

Въвеждаме тези параметри в командния прозорец на програмния продукт *Matlab*. Имайки предвид нормирането по честота въвеждаме честотите, нормирани с Fs/2. След въвеждането им преминаваме към някои команди:

1. [N1,Wn1]=ellipord(Wp1,Ws1,Rp,Rs) – *изчислява реда N и коригираната гранична честота Wn за този ред на цифров филтър по Кауер*
2. [Nz1,Dz1]=ellip(N1,Rp,Rs,Wn1) – *връща полином в числителя и полином в знаменателя на предавателната функция на нискочестотен филтър*

Първата команда връща резултат N1=5 и Wn1=0.4163. Втората команда извежда коефициентите на полиномните функции в числителя и знаменателя, които са:

*Числител* : [ 0.0968 0.1070 0.1956 0.1956 0.1070 0.0968]

*Знаменател*: [ 1.0000 -1.5137 2.2362 -1.5777 0.8688 -0.2148]

Оттук получаваме следната предавателна функция за получения нискочестотен филтър:

HLP(z)=

По същия начин работим и за високочестотния филтър. При него имаме гранична честота Wp=760 Hz при затихване 0.5 dB и честота на задържане Ws=460 Hz при затихване 30 dB.

Следва въвеждането на команди 1. и 2. като добиват нов вид за високочестотния филтър:

1. [N,Wn]=ellipord(Wp,Ws,Rp,Rs,’high’) – *изчислява реда N и коригираната гранична честота Wn за този ред на цифров филтър по Кауер*
2. [Nz,Dz]=ellip(N,Rp,Rs,Wn,’high’) – *връща полином в числителя и полином в знаменателя на предавателната функция на високочестотен филтър*

На параметрите на нискочестотния филтър ще добавяме символ **1** (Nz1,Dz1,Wn1 и т.н), за да можем да ги различаваме от тези на високочестотния филтър.

Първата команда ни извежда резултат N=4 и Wn=0.3439, а резултатът на втората команда е:

*Числител* : [ 0.2889 -0.9288 1.3090 -0.9288 0.2889 ]

*Знаменател*: [ 1.0000 -1.1333 1.2592 -0.3910 0.1828]

Оттук получаваме предавателната функция за получения високочестотен филтър:

HHP(z)=

***Криви на затихване на нискочестотен и високочестотен, заедно с габарита на лентов филтър.Честоти на задържане и гранични честоти.***

В точка „Апроксимиране на получените нискочестотен и високочестотен филтър.Предавателни функции. “ за граничните честоти на нискочестотния и високочестотния филтър получихме съответно Wn1=0.4163 и Wn=0.3439.

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

Wp1=920.023 Hz и Wp=760.019 Hz

Това са същност първоначалните честоти съответно на нискочестотния и на високочестотния филтър, които както забелязваме са променени незначително.

Преминаваме към затихванията на нискочестотния и високочестотни филтър, както и габаритите на лентовия филтър. За тях са ни нужни следните команди.Прилагаме ги поотделно и за двата филтъра.

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*

Затихването на нискочестотния филтър е в син цвят.

Като за високочестотния филтър ще изглежда така : plot(w\*Fs/(2\*pi),md,‘r’) – неговото затихване ще бъде с червен цвят.

1. hold on - *задържане на фигура, с цел наслагване на друга*

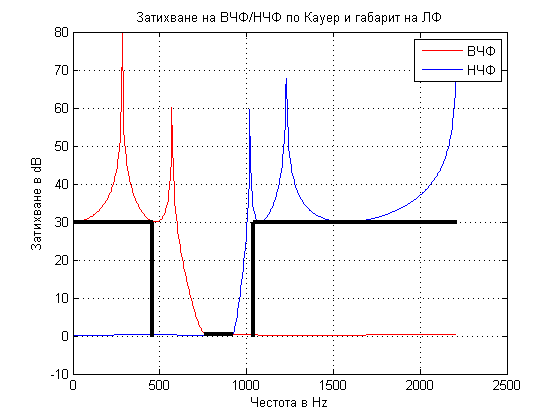
За изобразяване на габаритите на лентовия филтър използваме командите:

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 лентовият филтър в* ***черен цвят***

Ето го и резултата от всички команди:

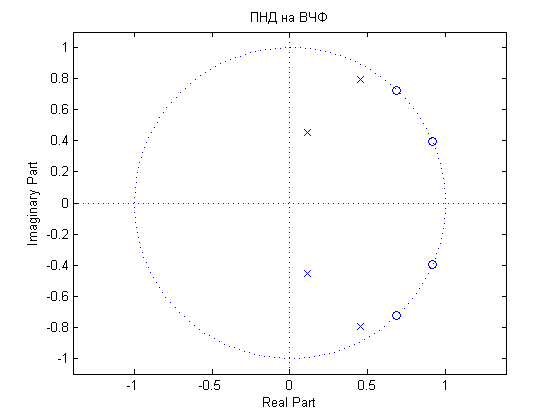
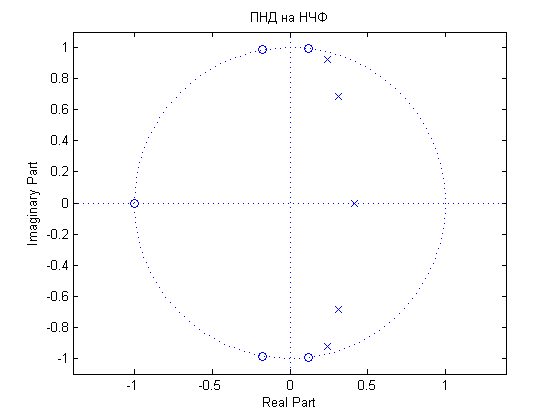


От получената графика установяваме, че граничните честоти не са се изменили или дори да са се изменили, разликите са незначителни, както вече разгледахме в началото на точката.

***Полюсно-нулеви диаграми и фазово-честотни характеристики на нискочестотен и високочестотен филтър.***

За изчертаване на полюсно-нулева диаграма използваме командата zplane(Nz1,Dz1) за нискочестотния филтър и zplane(Nz,Dz) за високочестотния.

А ето го и резултата:



Тук идва на помощ командата [z,p,k]=ellip(N1,Rp,Rs,Wn1) за нискочестотния и [z,p,k]=ellip(N,Rp,Rs,Wn) за високочестотния филтър, с която ще получим числените стойности на полюсите и нулите. А те са :

* **за нискочестотния филтър:**

*нули:* -1.0000 полюси: 0.2381+j0.9261

0.1214+j0.9926 0.2381-j0.9261

0.1214-j0.9926 0.3128+j0.6873

-0.1744+j0.9847 0.3128-j0.6873

-0.1744-j0.9847 0.4120

* **за високочестотен филтър:**

нули: 0.9194+j0.3933 полюси: 0.1159+j0.4548

0.9194-j0.3933 0.1159-j0.4548

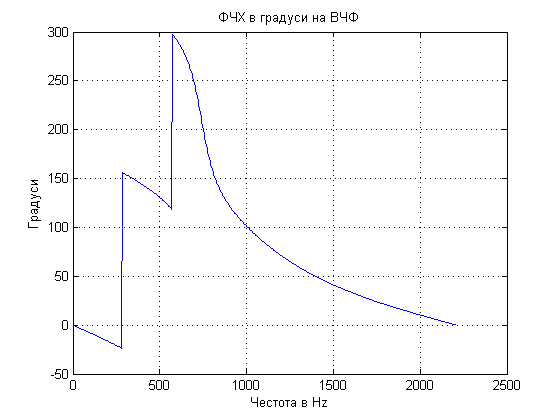
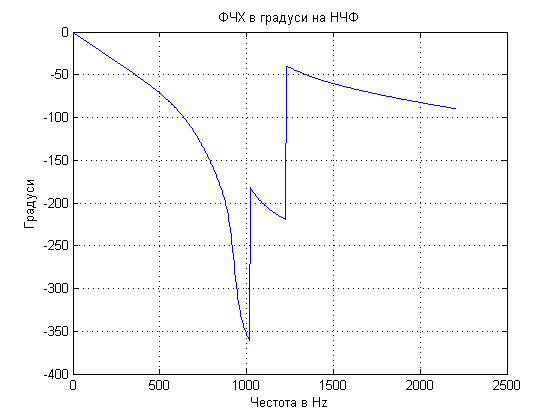
0.6884+j0.7253 0.4507+j0.7916

0.6884-j0.7253 0.4507-j0.7916

Следва да изчертаем фазово-честотните на ниско- и високочестотния филтър с помощта на командите:

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 цифров филтър, съставен чрез нискочестотен и високочестотен филтър, реализирани по паралелна реализация***

За реализирането на филтрите по паралелна реализация ще използваме предавателните функции на филтрите, тъй като командата, реализираща паралелната реализация изисква въвеждане коефициентите на числителя и знаменателя, а именно :

**b**=[…] – *въвеждане коефициенти на числителя*

**а**=[…]- *въвеждане коефициенти на знаменателя*

След като сме въвели коефициентите **b** и **а** можем да приложим и командата за паралелна реализация:

[C,B,A]=dir2par(b1,a1) – *преобразуване на директна реализация в паралелна на нискочестотния филтър*

[C,B,A]=dir2par(b,a) – *преобразуване на директна реализация в паралелна на високочестотния филтър*

С получените коефициенти C,B и А за ниско- и високочестотния филтър можем да съставим схемите, реализирани по паралелна реализация.

За да работят като лентов филтър, получените ниско- и високочестотен филтър трябва да се свържат последователно.Схемата на лентовия филтър ще изглежда по следния начин:

***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.

Входните сигнали, които ще дефинираме изглеждат така :

* **sin(2π800n)** – тук участва честота, избрана от лентата на пропускане (800 Hz) и очакваме сигналът да бъде пропуснат
* **sin(2π300n)** – избираме честота от лентата на задържане (300 Hz); тук сигналът не би трябвало да се пропусне

Използваните команди в тази точка са:

К=4420 – *брой точки за бързо преобразувание на Фурие;*

n=(0:99)/Fs – *определяне на времева област;*

x=sin(2.\*pi\*800.\*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;*

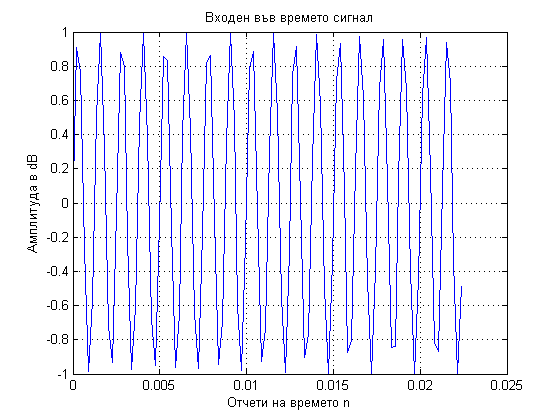
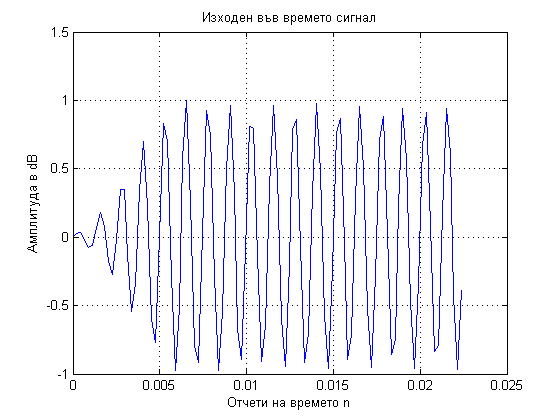
Px=fft(x,K1) - *изчисляване не бързото преобразувание на Фурие на изходен сигнал* ***х*** *в* ***К*** *точки;*

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

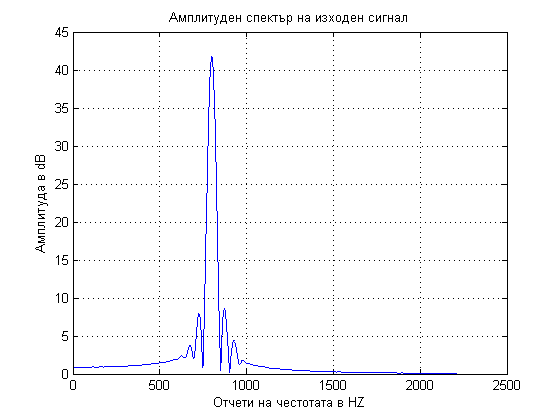
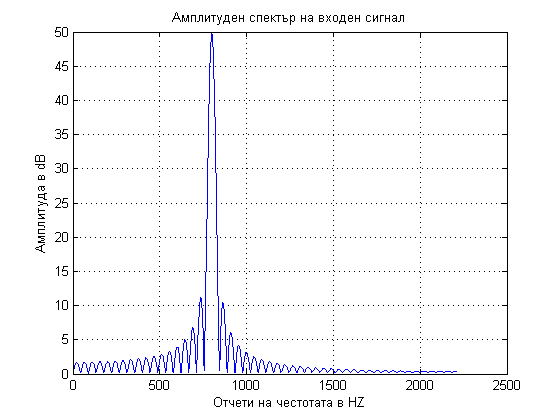
f=(0:(length(Px)-1)/2) – *създава вектор на честота с дължина Fs/2;*

Пропускаме през лентовия филтър синусоидален сигнал с честота **800 Hz** .

* **времева област**

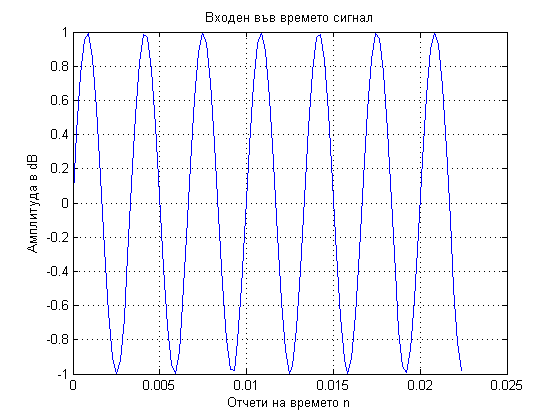
 

* **честотна област в К=4420 точки**



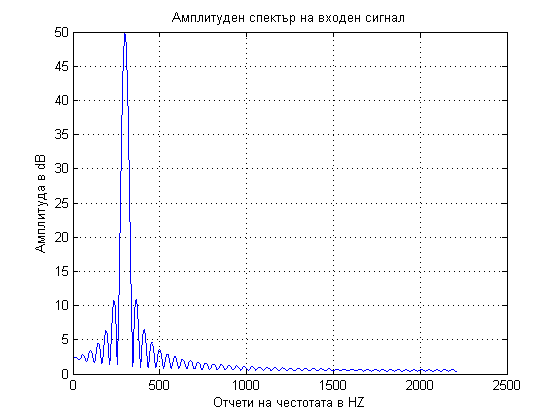
Сега пропускаме сигналът с честота **300 Hz**.

* **времева област**





* **честотна област в К=4420 точки**





Получените графики в честотна област по-горе доказват, че филтъра работи правилно. За да бъде филтриран един сигнал, амплитудата му на изхода трябва да е понижена поне 2 пъти.

При сигнала с честота 800 Hz (избран от лентата на пропускане) наблюдаваме незначително затихване. Амплитудата му от 50 dB намаля до 42 dB на изхода – сигналът се пропуска.

При сигнала с честота 300 Hz (избран от лентата на задържане) пропускане не се осъществява. Тук амплитудата от 50 dB спада до 0.45 dB – разликата е много голяма. На графиката на спектъра на изходния сигнал се наблюдава съставка с по-голяма амплитуда от останалите, но не достатъчна, за да бъде пропуснат сигнала. Това е съставката именно в граничната честота на нискочестотния филтър, който съставя лентовия ни филтър.

**Списък с използвани съкращения:**

* ПНД – полюсно-нулева диаграма
* НЧФ – нискочестотен филтър
* ВЧФ – високочестотен филтър
* ЛЗ – лента на задържане
* ЛП – лента на пропускане

**Съставен код на Matlab:**

clear all

close all

clc

Fn=2210;

Fs=4420;

Rp=0.5;Rs=30;

K=1024;

K1=4420;

Wp=760/Fn; %ВЧФ

Ws=460/Fn;

[N,Wn]=ellipord(Wp,Ws,Rp,Rs);

[Nz,Dz]=ellip(N,Rp,Rs,Wn,'high')

[z,p,k]=ellip(N,Rp,Rs,Wn,'high')

Wp1=920/Fn; %НЧФ

Ws1=1040/Fn;

[N1,Wn1]=ellipord(Wp1,Ws1,Rp,Rs);

[Nz1,Dz1]=ellip(N1,Rp,Rs,Wn1)

[z,p,k]=ellip(N1,Rp,Rs,Wn1)

figure(1); %ПНД

zplane(Nz,Dz);

title('ПНД на ВЧФ');

figure(2);

zplane(Nz1,Dz1);

title('ПНД на НЧФ');

figure (3);

[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('Честота в Hz');

ylabel('Затихване в dB');

title('Затихване на ВЧФ/НЧФ по Кауер');

legend('ВЧФ','НЧФ','--','ЛФ', 'k');

grid;

hold on;

x=linspace(0,460); %габарит на ЛФ

y=linspace(30,30);

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

x=linspace(460,460);

y=linspace(30,0);

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

x=linspace(760,920);

y=linspace(0.5,0.5);

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

x=linspace(1040,1040);

y=linspace(0,30);

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

x=linspace(1040,2210);

y=linspace(30,30);

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

figure(4) %ФЧХ на ВЧФ

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

grid;

title('ФЧХ в градуси на ВЧФ');

xlabel('Честота в Hz');

ylabel('Градуси');

figure (5) %ФЧХ на НЧФ

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

grid;

title('ФЧХ в градуси на НЧФ');

xlabel('Честота в Hz');

ylabel('Градуси');

b=[0.2889 -0.9288 1.3090 -0.9288 0.2889];

a=[1 -1.1333 1.2592 -0.3910 0.1828];

[C,B,A]=dir2par(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];

[C1,B1,A1]=dir2par(b1,a1)

n=(0:99)/Fs;

x=sin(2.\*pi\*800.\*n);

figure(6);

title('dsf');

plot(n,x);

grid;

xlabel('Отчети на времето n');

ylabel('Амплитуда в dB');

title('Входен във времето сигнал');

y1=filter(Nz,Dz,x);

y=filter(Nz1,Dz1,y1);

plot(n,y);

grid;

xlabel('Отчети на времето n');

ylabel('Амплитуда в dB');

title('Изходен във времето сигнал');

Px=fft(x,K1);

px=abs(Px(1:(K1/2)));

f=(0:(length(Px)-1)/2);

plot(f,px);

grid;

xlabel('Отчети на честотата в HZ');

ylabel('Амплитуда в dB');

title('Амплитуден спектър на входен сигнал');

Yx=fft(y,K1);

yx=abs(Yx(1:(K1/2)));

plot(f,yx);

grid;

xlabel('Отчети на честотата в HZ');

ylabel('Амплитуда в dB');

title('Амплитуден спектър на изходен сигнал');

n1=(0:99)/Fs;

x1=sin(2.\*pi\*300.\*n1);

figure(7);

plot(n1,x1);

grid;

xlabel('Отчети на времето n');

ylabel('Амплитуда в dB');

title('Входен във времето сигнал');

y11=filter(Nz,Dz,x1);

y1=filter(Nz1,Dz1,y11);

plot(n1,y1);

grid;

xlabel('Отчети на времето n');

ylabel('Амплитуда в dB');

title('Изходен във времето сигнал');

Px1=fft(x1,K1);

px1=abs(Px1(1:(K1/2)));

f1=(0:(length(Px1)-1)/2);

plot(f1,px1);

grid;

xlabel('Отчети на честотата в HZ');

ylabel('Амплитуда в dB');

title('Амплитуден спектър на входен сигнал');

Yx1=fft(y1,K1);

yx1=abs(Yx1(1:(K1/2)));

plot(f1,yx1);

grid;

xlabel('Отчети на честотата в HZ');

ylabel('Амплитуда в dB');

title('Амплитуден спектър на изходен сигнал');

**Използвани литературни източници:**

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.**