**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

**Навчально-науковий інститут Інформаційно-діагностичних систем**

**Кафедра біокібернетики та аерокосмічної медицини**

**Лабораторна робота №3**

З дисципліни: «Оброблення біомедичних сигналів»

Виконав

Студент групи БМ-462

Шелемба П.В.

Перевірив:

Гордєєв А.Д.

Київ-2017р

GitHub: https://github.com/MrShelemba/lab3

function lab03\_main

%=== Дисципліна:Основи обробки біомедичної інформації ===

%--- Лабораторна робота #3 ФІЛЬТРАЦІЯ БІОСИГНАЛІВ ФІЛЬТРАМИ З НІХ

%

% Використовуйте файли даних:

% ecg105.txt - сигнал ЕКГ

% ecg2x60.dat - сигнал ЕКГ з мережевою перешкодою частотою 60 Гц

%

%----------------------------------------------------------

clear, clc, close all

disp('Лабораторна робота #3')

disp('ФІЛЬТРАЦІЯ БІОСИГНАЛІВ ФІЛЬТРАМИ З НІХ')

disp('Виконав: Шелемба П.В., група БМ-462 ННІІДС')

%=== Завдання #1.1 ===

% Синтезування смугового фільтру 2-го порядку

fs = 200;

r = 0.6;

fc = 20; % центральна частота

phi = 2\*pi.\*fc/fs;

b = [1 0 -1]; % чисельник CФ

a = 1-2\*r\*cos(phi)\*r^2; % знаменник

%=== Завдання #1.2 ===

% АЧХ та ФЧХ смугового фільтру 2-го порядку

n = 512; % кількість точок, що розраховуються

figure(1)

[h, f] = freqz(b, a, n);

mag = abs(h);

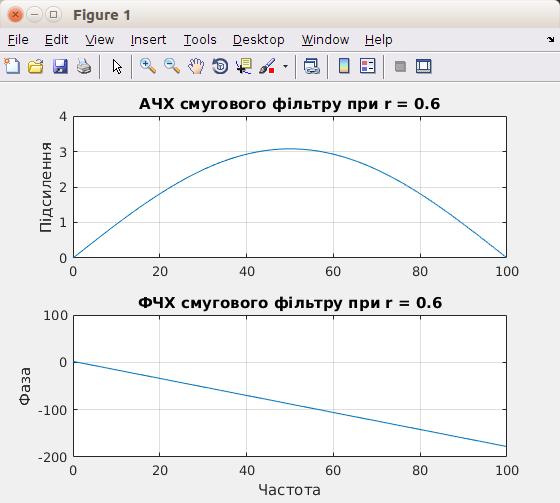
phase = angle(h)\*180/pi;

subplot(2, 1, 1); plot(f/(2\*pi)\*fs, mag), grid on;

title('АЧХ смугового фільтру при r = 0.6'); ylabel('Підсилення');

subplot(2, 1, 2); plot(f/(2\*pi)\*fs, unwrap(phase)), grid on;

title('ФЧХ смугового фільтру при r = 0.6'); xlabel('Частота'); ylabel('Фаза');



% Обчислення нулів та полюсів фільтру

disp('Нулі смугового фільтру при r = 0.6:');

x = roots(b);

disp(x);

disp('Полюси смугового фільтру при r = 0.6:');

y = poly(a);

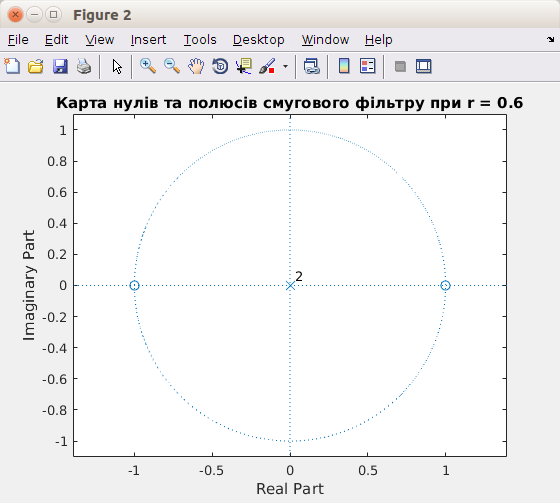
disp(y);

% Карта нулів та полюсів фільтру

figure(2)

z = zplane(b);

title('Карта нулів та полюсів смугового фільтру при r = 0.6');



%=== Завдання #1.3 ===

% r = 0.7

r1 = 0.7;

b1 = [1 0 -1];

a1 = 1-2\*r1\*cos(phi)\*r1^2;

figure(3)

[h1, f1] = freqz(b1, a1, n);

mag2 = abs(h1);

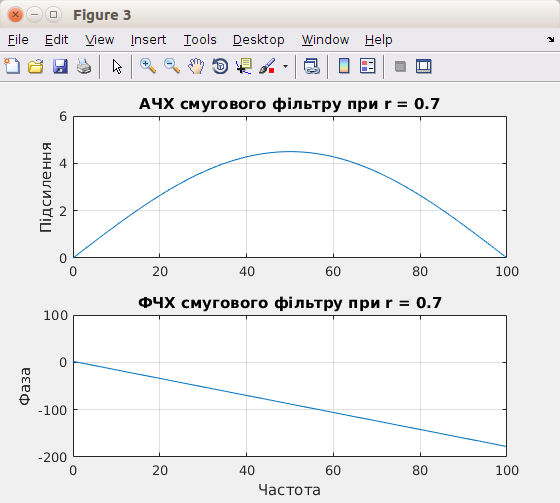
phase2 = angle(h1)\*180/pi;

subplot(2, 1, 1); plot(f1/(2\*pi)\*fs, mag2), grid on;

title('АЧХ смугового фільтру при r = 0.7'); ylabel('Підсилення');

subplot(2, 1, 2); plot(f1/(2\*pi)\*fs, unwrap(phase2)), grid on;

title('ФЧХ смугового фільтру при r = 0.7'); xlabel('Частота'); ylabel('Фаза');



% Обчислення нулів та полюсів фільтру

disp('Нулі смугового фільтру при r = 0.7:');

x1 = roots(b1);

disp(x1);

disp('Полюси смугового фільтру при r = 0.7:');

y1 = poly(a1);

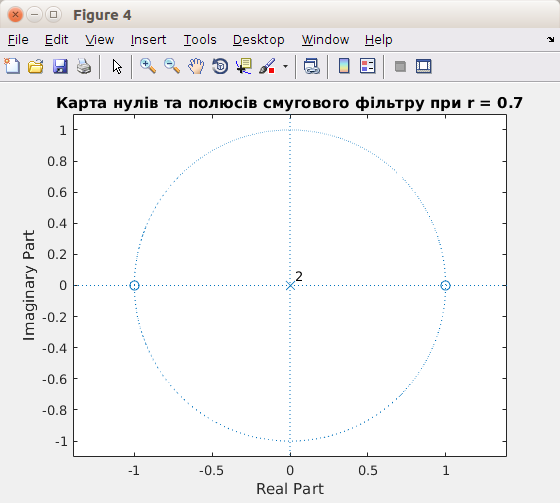
disp(y1);

% Карта нулів та полюсів фільтру

figure(4)

z1 = zplane(b1);

title('Карта нулів та полюсів смугового фільтру при r = 0.7');



% r = 0.9

r2 = 0.9;

b2 = [1 0 -1];

a2 = 1-2\*r2\*cos(phi)\*r2^2;

figure(5)

[h2, f2] = freqz(b2, a2, n);

mag3 = abs(h2);

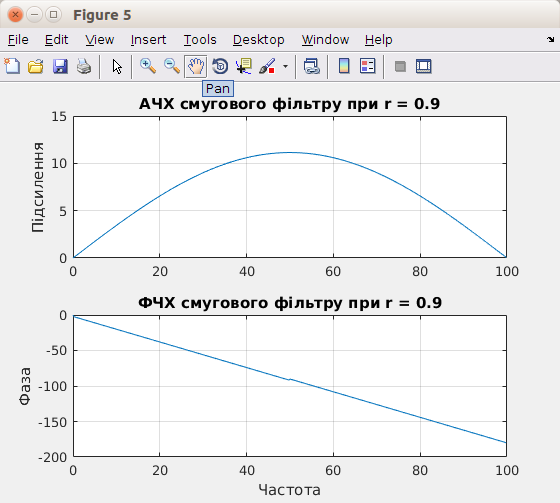
phase3 = angle(h2)\*180/pi;

subplot(2, 1, 1); plot(f2/(2\*pi)\*fs, mag3), grid on;

title('АЧХ смугового фільтру при r = 0.9'); ylabel('Підсилення');

subplot(2, 1, 2); plot(f2/(2\*pi)\*fs, unwrap(phase3)), grid on;

title('ФЧХ смугового фільтру при r = 0.9'); xlabel('Частота'); ylabel('Фаза');



% Обчислення нулів та полюсів фільтру

disp('Нулі смугового фільтру при r = 0.9:');

x2 = roots (b2);

disp(x2);

disp('Полюси смугового фільтру при r = 0.9:');

y2 = poly (a2);

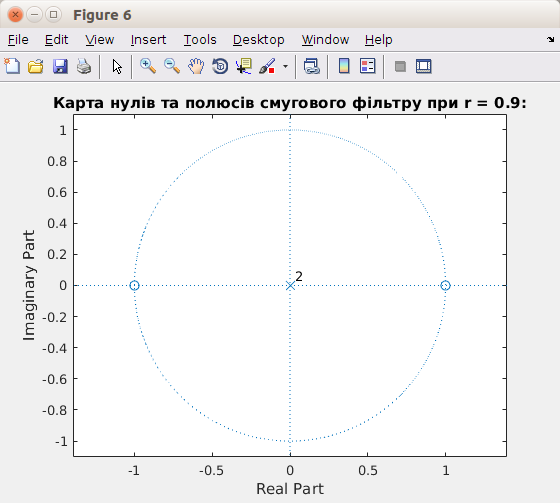
disp(y2);

% Карта нулів та полюсів фільтру

figure(6)

z2 = zplane(b2);

title('Карта нулів та полюсів смугового фільтру при r = 0.9:');



%=== Завдання #1.4 ===

%Визначення добротності

%при r=0.6 y=3.075 0.7y=2.1525

xd1=24.61;

xd2=75.39;

%при r=0.7 y=4.494 0.7y=3.1458 x1=24.61 x2=75.39

%при r=0.9 y=11.14 0.7y=7.798 x1=24.61 x2=75.39

disp('Добротність смугового фільтру:');

Q=fc/(xd2-xd1);

disp(Q);

%=== Завдання #1.5 ===

%Побудува графіків перехідних процесів

figure(7)

subplot(3,1,1); stepz(b, a);

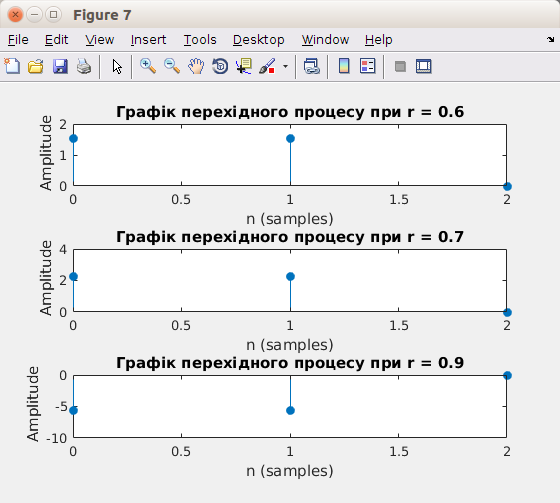
title('Графік перехідного процесу при r = 0.6');

subplot(3,1,2); stepz(b1, a1);

title('Графік перехідного процесу при r = 0.7');

subplot(3,1,3); stepz(b2, a2);

title('Графік перехідного процесу при r = 0.9');



%=== Завдання #2.1 ===

% Фільтрація ЕКГ при r = 0.6

r3 = 0.6;

b3 = [1 0 -1];

a3 = 1-2\*r3\*cos(phi)\*r3^2;

ecg = load('ecg105.txt'); % сигнал ЕКГ

ecgd1 = detrend(ecg);

ecgf1 = filter(b3, a3, ecgd1);

t1 = (0:length(ecgf1)-1)/fs;

figure(9)

subplot(2, 1, 1); plot(t1, ecg), grid on;

title('Сигнал з шумом');

xlim([0 2]);

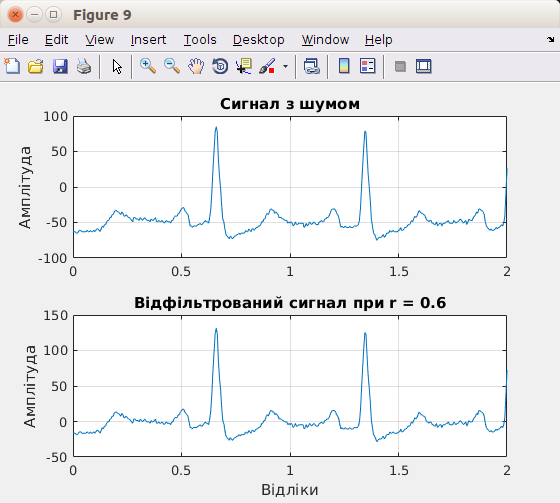
ylabel('Амплітуда');

subplot(2, 1, 2); plot(t1, ecgd1); grid on;

title('Відфільтрований сигнал при r = 0.6');

xlim([0 2]);

xlabel('Відліки'); ylabel('Амплітуда');



%=== Завдання #2.2 ===

% Фільтрація ЕКГ при r = 0.7 та r = 0.8

% r = 0.7

r4 = 0.7;

b4 = [1 0 -1];

a4 = 1-2\*r4\*cos(phi)\*r4^2;

ecgf2 = filter(b4, a4, ecgd1);

t2 = (0:length(ecgf2)-1)/fs;

figure(10)

subplot(2, 1, 1); plot(t2, ecgd1), grid on;

title('Сигнал з шумом');

xlim([0 2]);

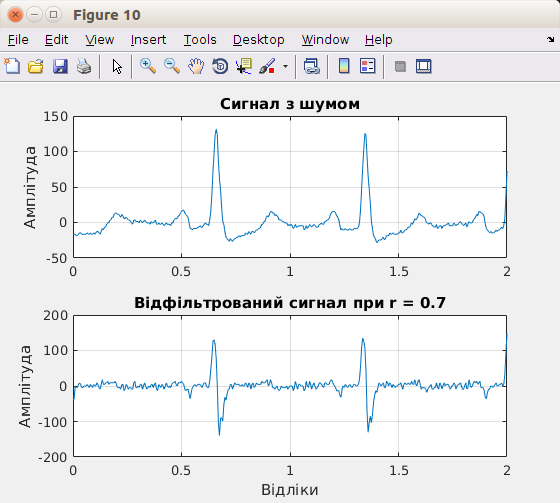
ylabel('Амплітуда');

subplot(2, 1, 2); plot(t2, ecgf2), grid on;

title('Відфільтрований сигнал при r = 0.7');

xlim([0 2]);

xlabel('Відліки'); ylabel('Амплітуда');



% r = 0.8

r5 = 0.8;

b5 = [1 0 -1];

a5 = 1-2\*r5\*cos(phi)\*r5^2;

ecgf3 = filter(b5, a5, ecgd1);

t3 = (0:length(ecgf3)-1)/fs;

figure(11)

subplot(2, 1, 1); plot(t3, ecgd1), grid on;

title('Сигнал з шумом');

xlim([0 2]);

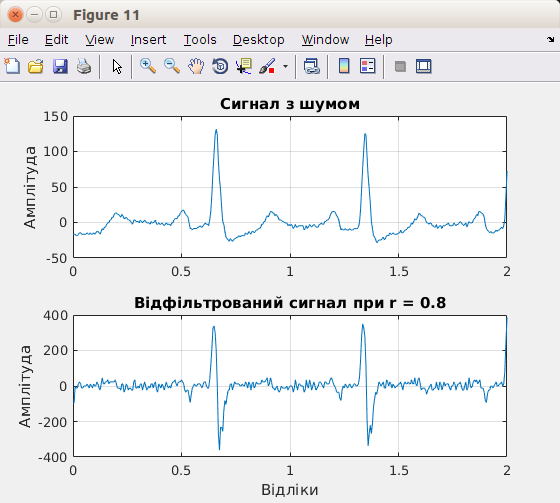
ylabel('Амплітуда');

subplot(2, 1, 2); plot(t3, ecgf3), grid on;

title('Відфільтрований сигнал при r = 0.8');

xlim([0 2]);

xlabel('Відліки'); ylabel('Амплітуда');



%Графіки результатів фільтрації

figure(12)

subplot(2, 1, 1); plot(t2, ecgf2); grid on;

title('Відфільтрований сигнал при r = 0.7');

xlim([0 2]);

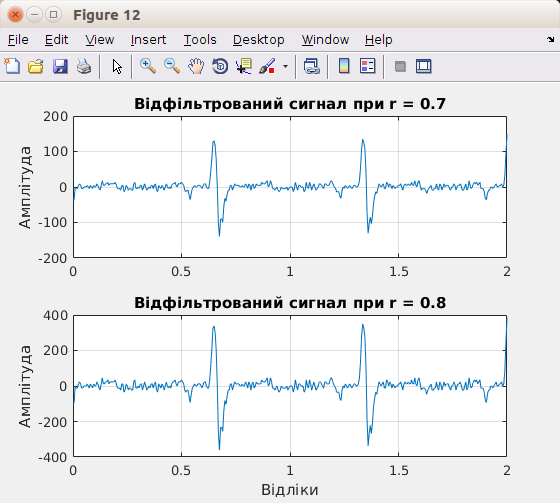
ylabel('Амплітуда');

subplot(2, 1, 2); plot(t3, ecgf3); grid on;

title('Відфільтрований сигнал при r = 0.8');

xlim([0 2]);

xlabel('Відліки'); ylabel('Амплітуда');



%=== Завдання #3.1 ===

% Дослідження властивостей режекторного НІХ-фільтру

r6 = 0.8;

phi1 = 110\*pi/180;

phi2 = 130\*pi/180;

ax = [1-2\*r6\*cos(phi1)\*r6^2] ;

ax2 = [1-2\*r6\*cos(phi2)\*r6^2];

a6 = conv(ax, ax2);

b6 = [1 1 1];

% Передавальна функція

disp('Передавальна функція режекторного НІХ-фільтру:');

H1 = filt(b6, a6) % передавальна функція

%=== Завдання #3.2 ===

% АЧХ та ФЧХ смугового режекторного НІХ-фільтру

figure(12)

[h4, f4] = freqz(a6, b6, n);

mag4 = abs(h4);

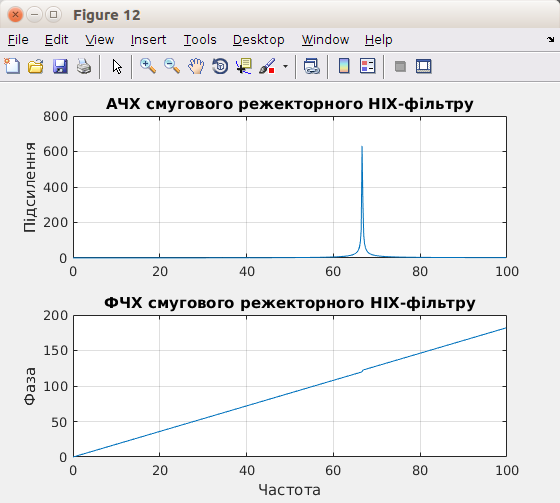
phase4 = angle(h4)\*180/pi;

subplot(2, 1, 1); plot(f4/(2\*pi)\*fs, mag4), grid on;

title('АЧХ смугового режекторного НІХ-фільтру'); ylabel ('Підсилення');

subplot(2, 1, 2); plot(f4/(2\*pi)\*fs, unwrap(phase4)), grid on;

title('ФЧХ смугового режекторного НІХ-фільтру'); xlabel('Частота'); ylabel('Фаза');



% Обчислення нулів та полюсів фільтру

disp('Нулі смугового режекторного НІХ-фільтру:');

x3 = roots(b);

disp(x3);

disp('Полюси смугового режекторного НІХ-фільтру:');

y3 = poly(a);

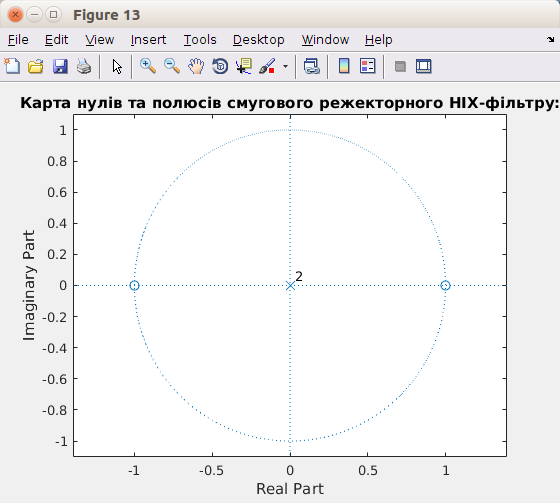
disp(y3);

% Карта нулів та полюсів фільтру

figure(13)

z3 = zplane(b,a);

title('Карта нулів та полюсів смугового режекторного НІХ-фільтру:');



%=== Завдання #3.3 ===

% Порівняння АЧХ і ФЧХ режекторних НІХ і СІХ-фільтрів

figure(14)

subplot(4, 1, 1); plot(f4/(2\*pi)\*fs, mag4), grid on;

title('АЧХ режекторного НІХ-фільтру'); ylabel('Підсилення');

subplot(4, 1, 2); plot(f4/(2\*pi)\*fs, unwrap(phase4)), grid on;

title('ФЧХ режекторного НІХ-фільтру'); xlabel('Частота'); ylabel('Фаза');

b7 = [1,0.618, 1]; % коефіцієнти різницевого рівняння

a7 = 1;

[h5, f5] = freqz(b7, a7, n, fs);

mag5 = abs(h5);

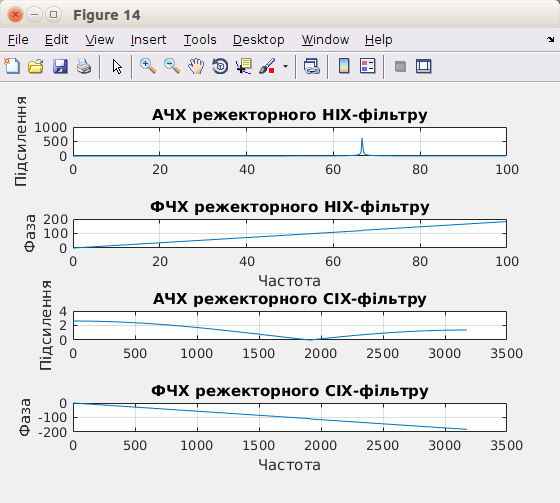
phase5 = angle(h5)\*180/pi;

subplot(4, 1, 3); plot(f5/(2\*pi)\*fs, mag5), grid on;

title('АЧХ режекторного СІХ-фільтру'); ylabel('Підсилення');

subplot(4, 1, 4); plot(f5/(2\*pi)\*fs, unwrap(phase5)), grid on;

title('ФЧХ режекторного СІХ-фільтру'); xlabel('Частота'); ylabel('Фаза');



%=== Завдання #3.4 ===

% Фільтрацію сигналу ЕКГ(файл ecg2x60.dat) режекторним фільтром

ecg2 = load('ecg2x60.dat'); % сигнал ЕКГ

ecgd2 = detrend(ecg2);

ecgf4 = filter(b6, a6, ecgd2);

t4 = (0:length(ecgf4)-1)/fs;

figure(15)

subplot(2, 1, 1); plot(t4, ecgd2), grid on;

title('Сигнал з мережевою перешкодою 60 Гц');

xlim([1 2]);

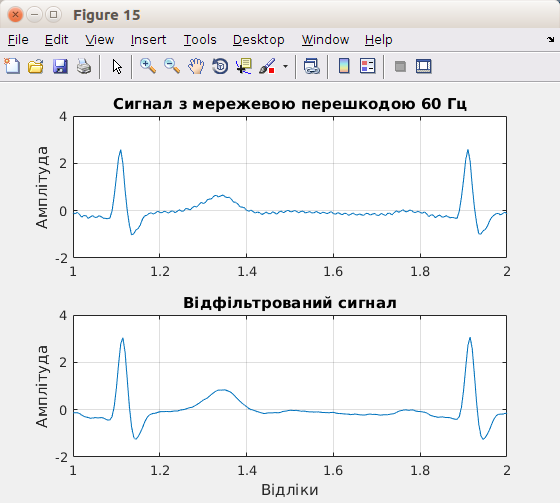
ylabel('Амплітуда');

subplot(2, 1, 2); plot (t4, ecgf4), grid on;

title('Відфільтрований сигнал');

xlim([1 2]);

xlabel('Відліки'); ylabel('Амплітуда');



%=== Завдання #4.1 ===

% АЧХ і ФЧХ цифрових інтеграторів

% Інтегрування методом прямокутників

fs = 300; % частота дискретизації

T = 1/fs;

bi1 = [T T]; % чисельник ПФ

ai1 = [1 -1]; % знаменник ПФ

% Інтегрування методом трапецій.

bi2 = [1 1]\*T/2;

ai2 = [1 -1];

% Інтегрування методом парабол (Сімпсона).

bi3 = [1 4 1]\*T/3;

ai3 = [1 0 -1];

[hi1, wi1] = freqz(bi1, ai1, n);

magi1 = abs(hi1);

phasei1 = angle(hi1)\*180/pi;

[hi2, wi2] = freqz(bi2, ai2, n);

magi2 = abs(hi2);

phasei2 = angle(hi2)\*180/pi;

[hi3, wi3] = freqz(bi3, ai3, n);

magi3 = abs(hi3);

phasei3 = angle(hi3)\*180/pi;

figure(16)

subplot(3, 1, 1); plot(wi1/(2\*pi)\*fs, magi1), grid on;

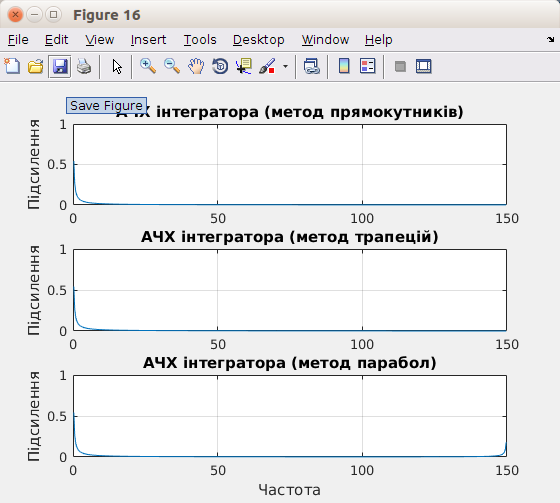
title('АЧХ інтегратора (метод прямокутників)'); ylabel('Підсилення');

subplot(3, 1, 2); plot(wi2/(2\*pi)\*fs, magi2), grid on;

title ('АЧХ інтегратора (метод трапецій)'); ylabel ('Підсилення');

subplot(3, 1, 3); plot(wi3/(2\*pi)\*fs, magi3), grid on;

title ('АЧХ інтегратора (метод парабол)'); xlabel ('Частота'); ylabel ('Підсилення');



figure(17)

subplot(3, 1, 1); plot(wi1/(2\*pi)\*fs, unwrap(phasei1)), grid on;

title('ФЧХ інтегратора (метод прямокутників)'); ylabel('Фаза');

subplot(3, 1, 2); plot(wi2/(2\*pi)\*fs, unwrap(phasei2)), grid on;

title('ФЧХ інтегратора (метод трапецій)'); ylabel('Фаза');

subplot(3, 1, 3); plot(wi3/(2\*pi)\*fs, unwrap(phasei3)), grid on;

title('ФЧХ інтегратора (метод парабол)'); xlabel('Частота'); ylabel('Фаза');



%=== Завдання #4.2 ===

% Обчислення нулів та полюсів інтеграторів

% метод прямокутників

disp('Нулі інтегратора (метод прямокутників):');

xi1 = roots(bi1);

disp(xi1);

disp('Полюси інтегратора (метод прямокутників):');

yi1 = poly(ai1);

disp(yi1);

%метод трапецій

disp('Нулі інтегратора (метод трапецій):');

xi2 = roots(bi2);

disp(xi2);

disp('Полюси інтегратора (метод трапецій):');

yi2 = poly(ai2);

disp(yi2);

%метод Сімпсона

disp('Нулі інтегратора (метод парабол):');

xi3 = roots(bi3);

disp(xi3);

disp('Полюси інтегратора (метод парабол):');

yi3 = poly(ai3);

disp(yi3);

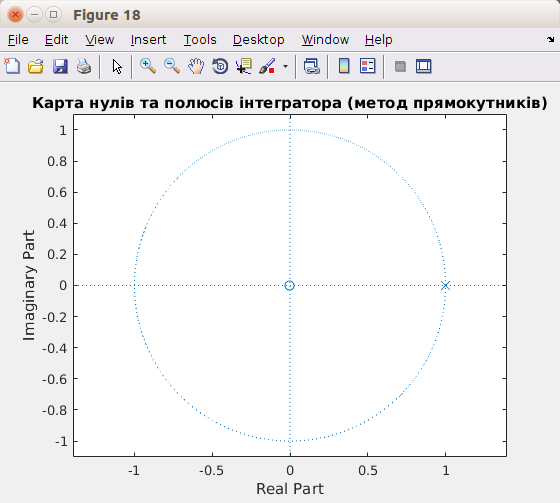
% Карти нулів та полюсів інтеграторів

% метод прямокутників

figure(18);

zi1 = zplane(bi1, ai1);

title('Карта нулів та полюсів інтегратора (метод прямокутників)');

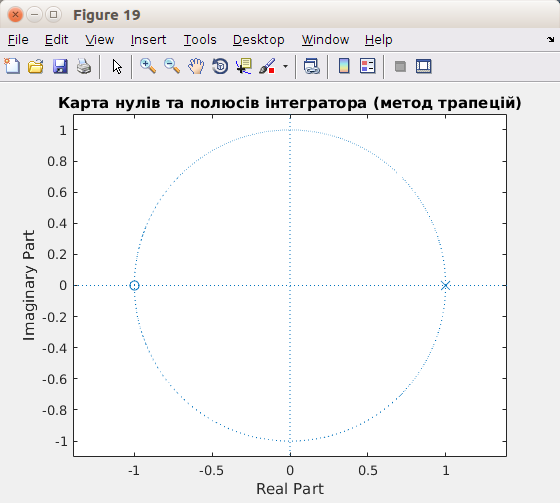


% метод трапецій

figure(19);

zi2 = zplane(bi2, ai2);

title('Карта нулів та полюсів інтегратора (метод трапецій)');

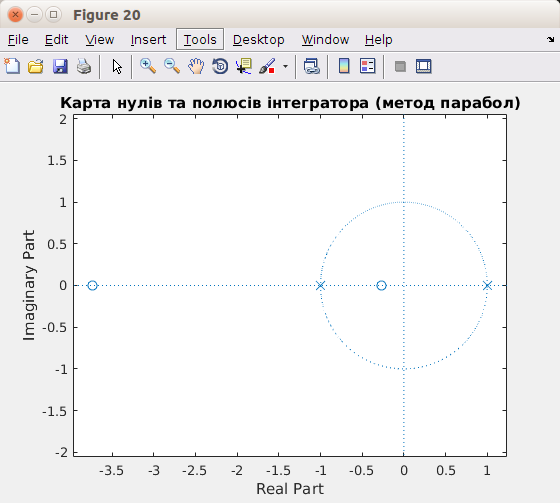


% метод парабол

figure(20);

zi3 = zplane(bi3, ai3);

title('Карта нулів та полюсів інтегратора (метод парабол)');



%=== Завдання #4.3 ===

% Обчислення абсолютної похибки АЧХ

% метод прямокутників

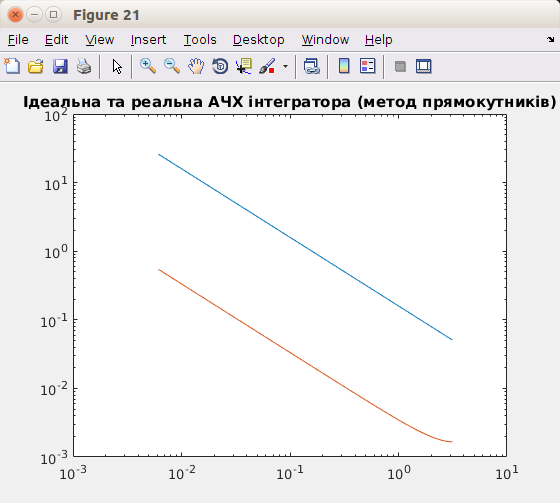
figure(21);

mag0 = 1./(2\*pi\*f); % АЧХ ідеального інтегратора

loglog(f, mag0, f, magi1);

err1 = (magi1 - mag0)\*100;

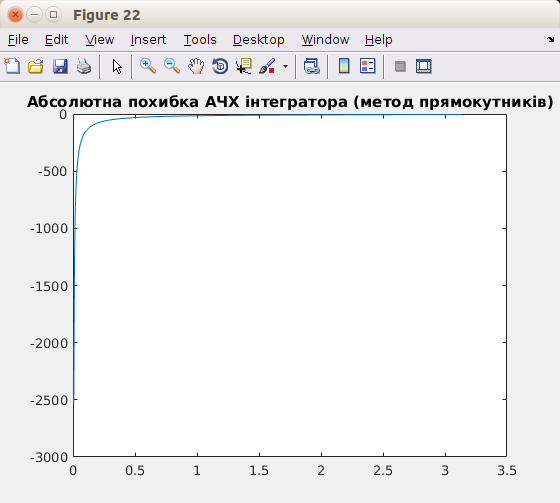
title('Ідеальна та реальна АЧХ інтегратора (метод прямокутників)');



figure(22);

plot(f, err1);

title('Абсолютна похибка АЧХ інтегратора (метод прямокутників)');



%метод трапецій

figure(23);

loglog(f, mag0, f, magi2); %логарифмічний маштаб

err2 = (magi2 - mag0)\*100;

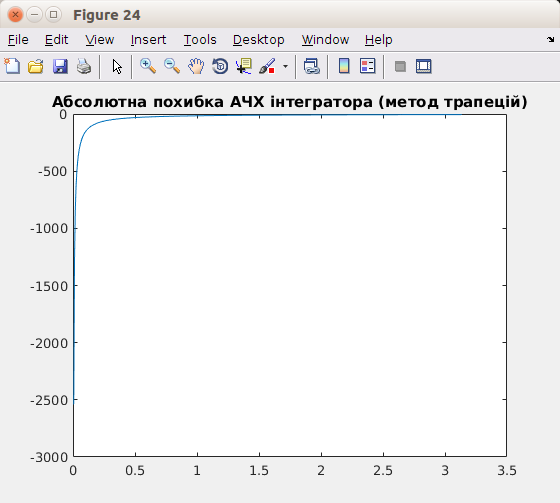
title('Ідеальна та реальна АЧХ інтегратора (метод трапецій)');



figure(24);

plot(f, err2);

title('Абсолютна похибка АЧХ інтегратора (метод трапецій)');



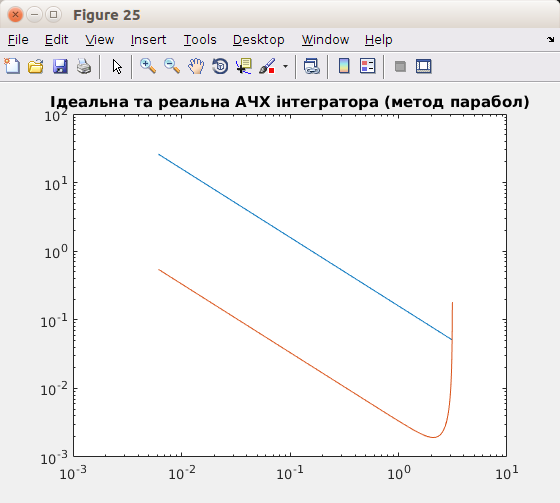
% метод Сімпсона

figure(25);

loglog(f, mag0, f, magi3);

err3 = (magi3 - mag0)\*100;

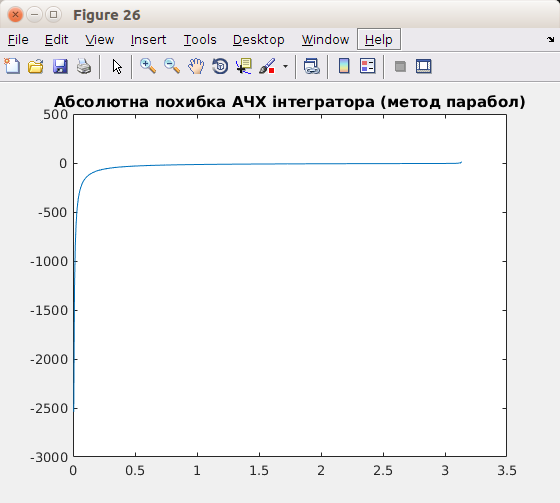
title('Ідеальна та реальна АЧХ інтегратора (метод парабол)');



figure(26);

plot(f, err3);

title('Абсолютна похибка АЧХ інтегратора (метод парабол)');



%=== Завдання #4.4 ===

% Інтегрування сигналу ЕКГ (файл ecg105.txt) інтеграторами

% метод прямокутників

ecgfi1 = filter(bi1, ai1, ecgd1);

ti1 = (0:length(ecgfi1)-1)/fs;

figure(27)

subplot(2, 1, 1); plot(ti1, ecgd1), grid on;

title('Сигнал з шумом');

xlim([0 1]);

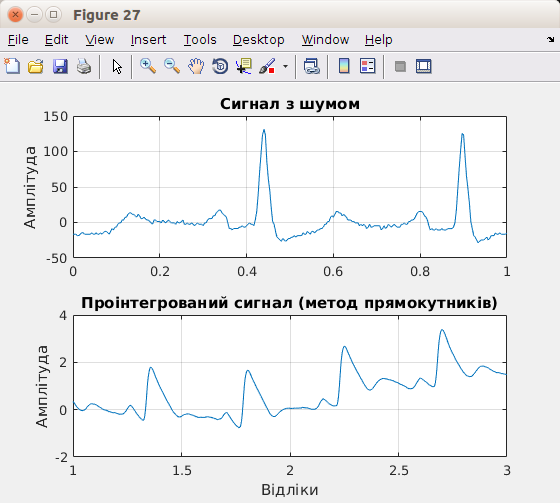
ylabel('Амплітуда');

subplot(2, 1, 2); plot(ti1, ecgfi1); grid on;

title('Проінтегрований сигнал (метод прямокутників)');

xlim([1 3]);

xlabel('Відліки'); ylabel('Амплітуда');



%метод трапецій

ecgfi2 = filter(bi2, ai2, ecgd1);

ti2 = (0:length(ecgfi2)-1)/fs;

figure(28)

subplot(2, 1, 1); plot(ti2, ecgd1), grid on;

title('Сигнал з шумом');

xlim([0 1]);

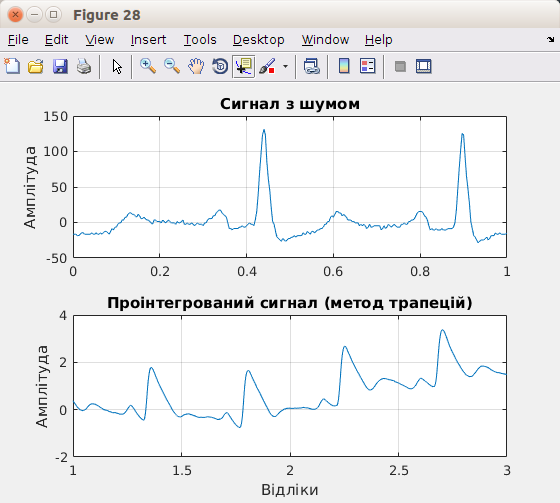
ylabel('Амплітуда');

subplot(2, 1, 2); plot(ti2, ecgfi2); grid on;

title('Проінтегрований сигнал (метод трапецій)');

xlim([1 3]);

xlabel('Відліки'); ylabel('Амплітуда');



% метод Сімпсона

ecgfi3 = filter(bi3, ai3, ecgd1);

ti3 = (0:length(ecgfi3)-1)/fs;

figure(29)

subplot(2, 1, 1); plot(ti3, ecgd1), grid on;

title('Сигнал з шумом');

xlim([0 1]);

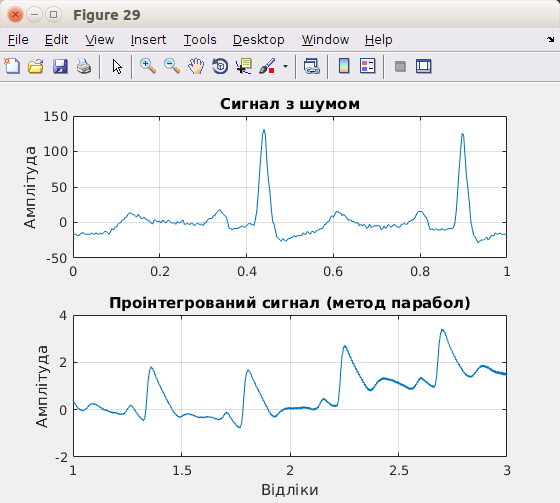
ylabel('Амплітуда');

subplot(2, 1, 2); plot(ti3, ecgfi3); grid on;

title('Проінтегрований сигнал (метод парабол)');

xlim([1 3]);

xlabel('Відліки'); ylabel('Амплітуда');

****

**Повідомлення командного вікна:**

Лабораторна робота #3

ФІЛЬТРАЦІЯ БІОСИГНАЛІВ ФІЛЬТРАМИ З НІХ

Виконав: Шелемба П.В., група БМ-462 ННІІДС

Нулі смугового фільтру при r = 0.6:

-1

1

Полюси смугового фільтру при r = 0.6:

1.0000 -0.6505

Нулі смугового фільтру при r = 0.7:

-1

1

Полюси смугового фільтру при r = 0.7:

1.0000 -0.4450

Нулі смугового фільтру при r = 0.9:

-1

1

Полюси смугового фільтру при r = 0.9:

1.0000 0.1795

Добротність смугового фільтру:

0.3939

Передавальна функція режекторного НІХ-фільтру:

H1 =

1 + z^-1 + z^-2

---------------

2.239

Sample time: unspecified

Discrete-time transfer function.

Нулі смугового режекторного НІХ-фільтру:

-1

1

Полюси смугового режекторного НІХ-фільтру:

1.0000 -0.6505

Нулі інтегратора (метод прямокутників):

-1

Полюси інтегратора (метод прямокутників):

1 0 -1

Нулі інтегратора (метод трапецій):

-1

Полюси інтегратора (метод трапецій):

1 0 -1

Нулі інтегратора (метод парабол):

-3.7321

-0.2679

Полюси інтегратора (метод парабол):

1 0 -1 0

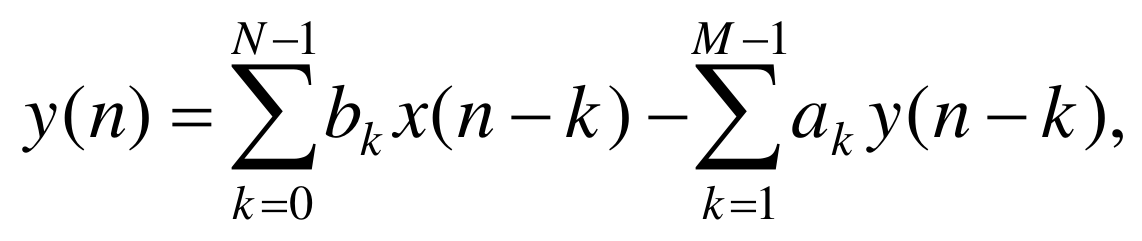
Контрольні запитання

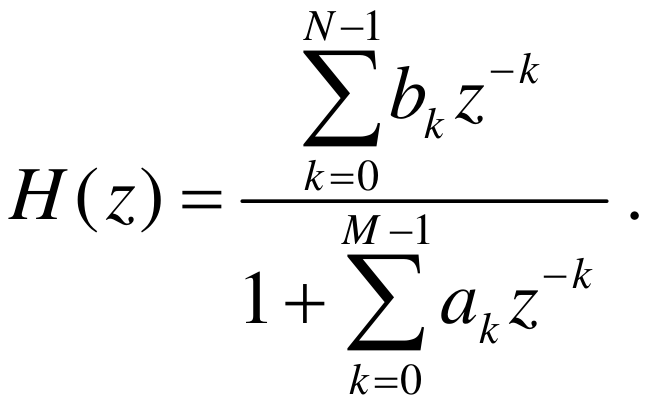
**1. Як експериментально визначити імпульсну та перехідну характеристики цифрового фільтра?**

Для цього необхідно подати на фільтр одиничний імпульс та зареєструвати його відповідь на нього.

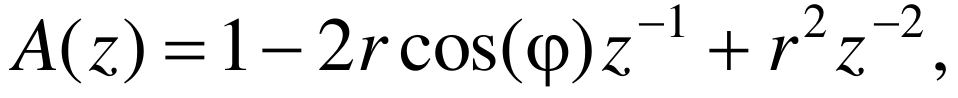
**2. Як визначити передавальну функцію цифрового НІХ-фільтра за його різницевим рівнянням?**

Якщо різницеве рівнянна записати, як:

то його передавальну функцію можна зобразити наступним виглядом:



**3. Яке перетворення пов’язує частотну та імпульсну характеристики цифрового фільтру?**



**4. За якими критеріями визначають стійкість НІХ-фільтру?**

Фільтр називається стійким, якщо при будь-яких початкових умовах реакція фільтра на будь-який обмежений вплив також обмежена. Критерієм стабільності фільтра є абсолютна збіжність відліків його імпульсного відгуку

**ВИСНОВКИ**

Виконуючи дану лабораторну роботу я дослідив властивості цифрових фільтрів з нескінченною імпульсною характеристикою за допомогою програмномного середовища MatLab. В даній роботі наведені АЧХ та ФЧХ даних фільтрів, карти нулів та полюсів, їх значення, а також порівняння з графіками АЧХ та ФЧХ фільтрів зі скінченною імпульсною характеристикою. З допомогою фільтрів з нескінченою імпульсною характеристикою проведено фільтрацію ЕКГ сигналів та наведено результати їх роботи.

Також при виконанні було засвоєно нові команди програмного продукту MatLab та можливості їх використання . Дану роботу вважаю виконаною так як було виконано поставлені задачі.