ФГБОУ ВО “Чувашский государственный университет им. И. Н. Ульянова” Факультет: ИВТ

Кафедра: Вычислительной техники

Предмет: Теория цифровых сигналов

Лабораторная работа №2

Проектирование, реализация и исследование характеристик БИХ-фильра

Вариант 6

Выполнил: студент группы ИВТ-41-20

Галкин Дмитрий

Проверила: Аркадьевна А.А.

Чебоксары 2023 г.

**Цель работы:** Получение опыта проектирования и программной реализации цифровых фильтров.

**Задание 1:**

* В системе Matlab составить программу расчета БИХ(КИХ)-фильтра.
* Построить частотные характеристики фильтра (АЧХ, ФЧХ, ЛАЧХ, характеристика групповой задержки)
* Построить импульсную и временные характеристики
* Получить реакцию фильтра на синусоидальный входной сигнал с частотами из полосы пропускания или полосы непропускания
* Получить коэффициенты последовательной и параллельной (только для БИХ-фильтра) форм реализации фильтра

**Задание 2:**

* Программно реализовать на языке высокого уровня фильтр в прямой, канонический (БИХ-фильтр), последовательной и параллельной (БИХ-фильтр) формах при представлении коэффициентов разностного уравнения в форматах
  + с плавающей точкой
  + с фиксированной точкой (16 бит)
* Получить реакцию фильтра на единичный импульс, единичный ступенчатый и синусоидальный сигналы

**Решение в Matlab:**

% Задаем характиристики фильтра

% Частота дискретизации

Fd = 1000

% Частота среза полосы пропускания и непропускания

fp = 220

fs = 150

% Величины затухания

Rp = 0.2000

Rs = 55

% Нормализуем частоты

FdN = Fd / 2

FdN = 500

Wp = fp / FdN

Wp = 0.4400

Ws = fs / FdN

Ws = 0.3000

% Расчет фильра

% Вычисляем порядок фильтра и частоту среза

[n, Wn] = buttord(Wp, Ws, Rp, Rs)

n = 17

Wn = 0.4122

Порядок фильтра: 17

Частота среза: 0.4122

% Вычисляем коэффициенты фильтра

[b, a] = butter(n, Wn, ‘high’)

Коэффициенты b: [0.0005, -0.0085, 0.0681, -0.3404, 1.1914, -3.0976, 6.1952, -9.7354, 12.1692,

-12.1692, 9.7354, -6.1952, 3.0976, -1.1914, 0.3404, -0.0681, 0.0085, -0.0005]

Коэффициенты a: [1.0000, -3.2134, 6.9987, -10.4919, 12.3903, -11.6302, 8.9979, -5.7521 3.0712, -1.3647, 0.5037, -0.1527, 0.0375, -0.0073, 0.0011, -0.0001, 0.0000, -0.0000]

% Построение частотных характеристик

% Вычисляем частотную характеристику

[h, w] = freqz(b, a)

% Задаем сетку частот

ff = w/pi\*FdN;

%Построение АЧХ

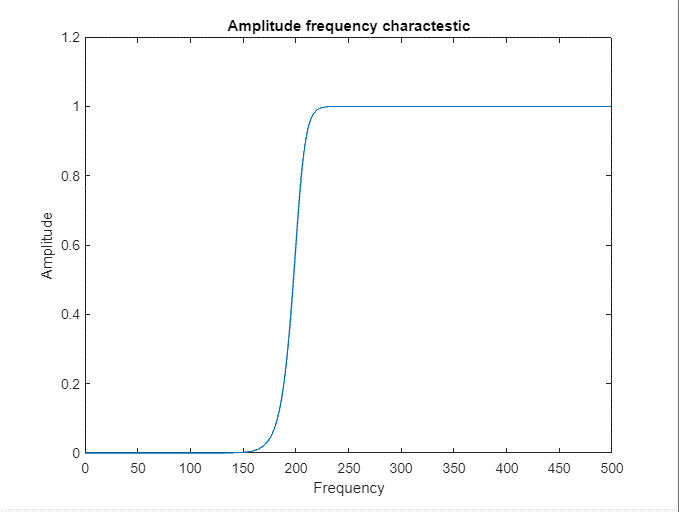
figure(1)

plot(ff, abs(h))

title('Amplitude frequency charactestic')

xlabel('Frequency')

ylabel('Amplitude')



%Построение ФЧХ

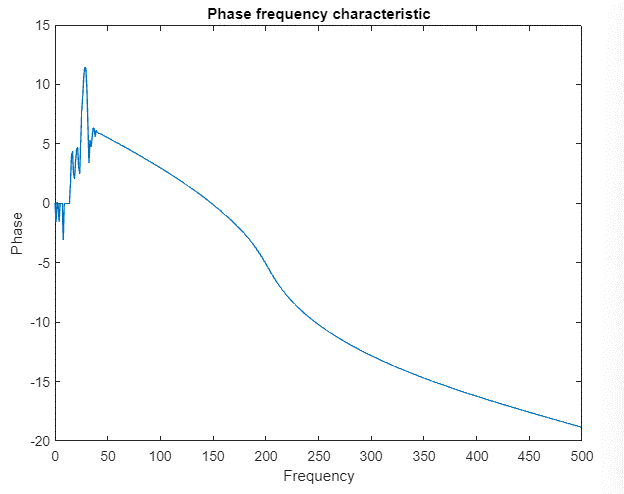
figure(2)

plot(ff, unwrap(angle(h)))

title('Phase frequency characteristic')

xlabel('Frequency')

ylabel('Phase')



%Построение ЛАЧХ

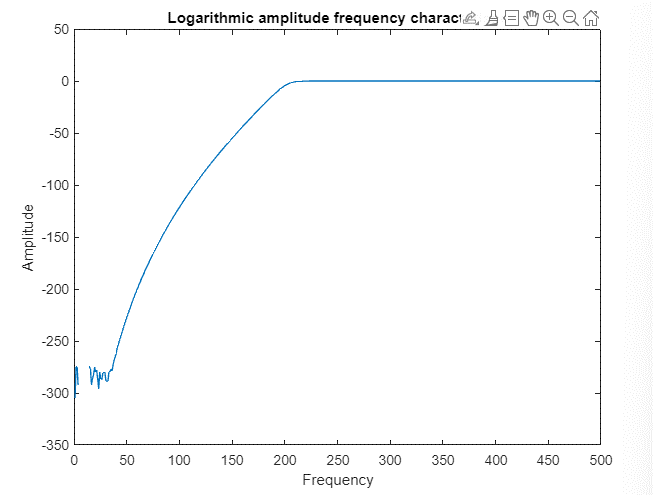
figure(3)

plot(ff, 20 \* log10(abs(h)))

title('Logarithmic amplitude frequency characteristic')

xlabel('Frequency')

ylabel('Amplitude')



% Построение групповой задержки

figure(4)

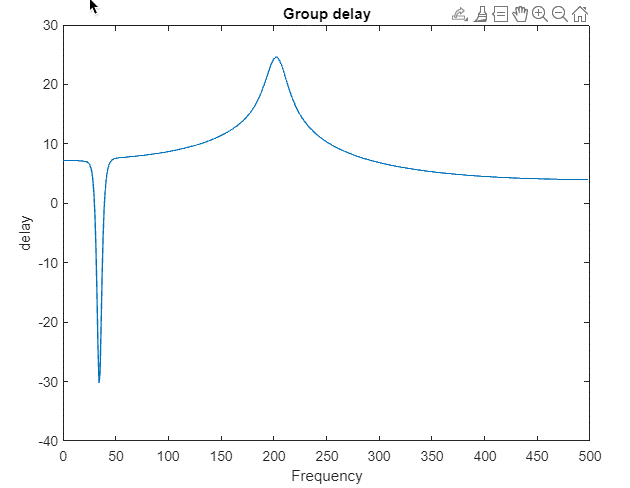
gd = **grpdelay(b, a);**

**plot(ff, gd);**

**xlabel('Frequency');**

**ylabel(**'delay');

title('Group delay');

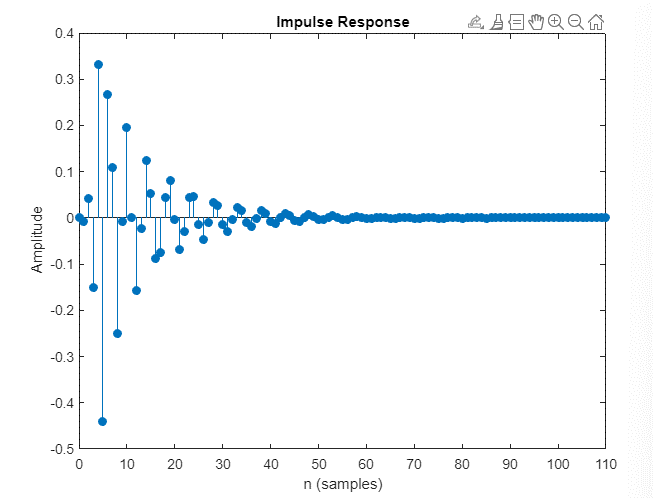


% Построение импульсной и временной характеристик

% Импульсная характеристика

figure(5)

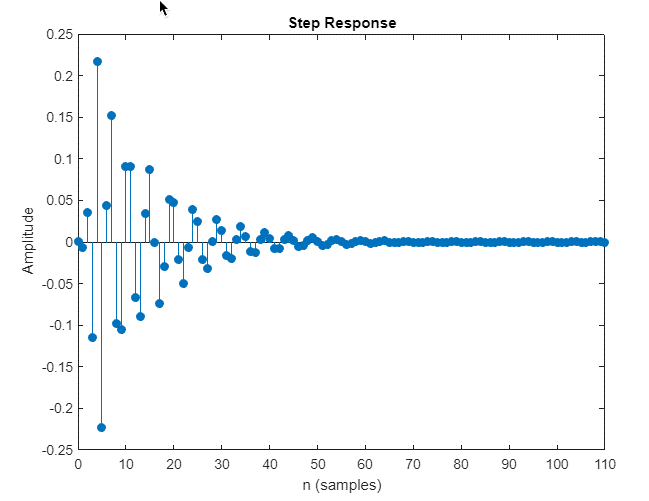
impz(b,a);



% Временная характеристика

figure(6)

stepz(b,a);



% Вычисление и построение реакции фильтра

% на синусоидальный сигнал в полосе пропускания и непропускания

% a) в полосе пропускания

nn = (0:99)

f1 = 300

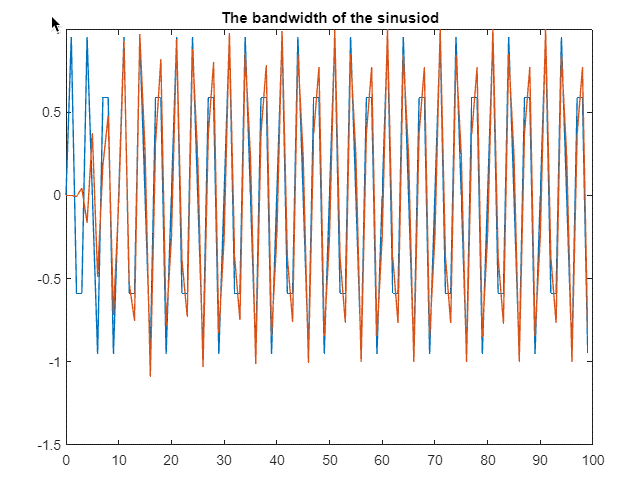
x1 = sin(2\*pi\*f1\*nn/Fd)

y1 = filter(b, a, x1)

figure(7)

plot(nn, x1, nn, y1)

title('The bandwidth of the sinusiod')



% б) в полосе непропускания

f2 = 800

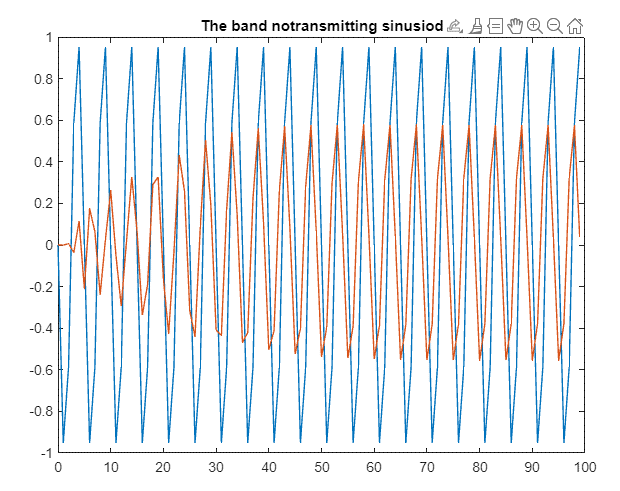
x1 = sin(2\*pi\*f2\*nn/Fd)

y1 = filter(b, a, x1)

figure(8)

plot(nn, x1, nn, y1)

title('The band notransmitting sinusiod')

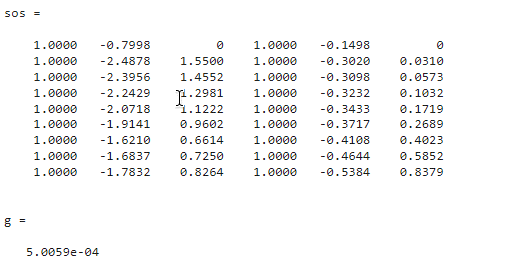


% Получаем коэффициенты последовательной и

% параллельной форм реалезации фильтра

% Построение последовательной формы фильтра

[sos, g] = tf2sos(b, a)



% Построение параллельной формы фильтра

[r p k] = residuez(b, a)

Bi=[ ; ]

Ai = [ ; ]

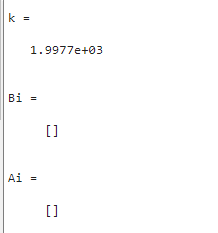
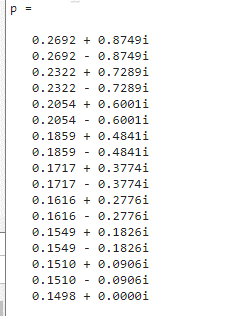
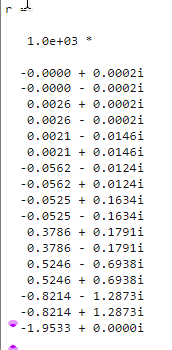
for i = 1: 2: floor(n/2)\*2

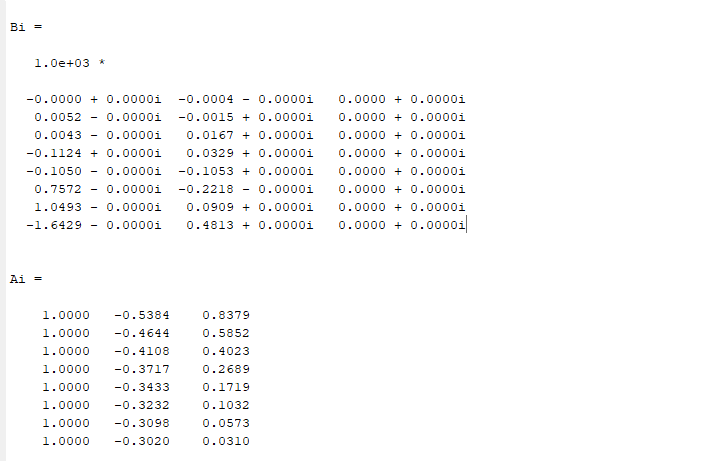
[bi ai] = residuez(r(i : i+1), p(i : i+1), 0)

Bi((i + 1)/2, : ) = bi

Ai((i + 1)/2, : ) = ai

End





**Текст программы в MATLAB**

* С плавающей точкой
  + Прямая форма

Fd = 1000

fp = 220

fs = 150

Rp = 0.2

Rs = 55

FdN = Fd / 2

Wp = fp / FdN

Ws = fs / FdN

[n, Wn] = buttord(Wp, Ws, Rp, Rs)

[b, a] = butter(n, Wn, 'high')

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

[h,w]=freqz(b,a);

ff = w/pi\*FdN;

h1 = impz(b,a, 100);

% figure(6)

h2 = stepz(b,a, 100);

nn = (0:99)

f1 = 300

x1 = sin(2\*pi\*f1\*nn/Fd)

y1 = filter(b, a, x1)

f2 = 100

x2 = sin(2\*pi\*f2\*nn/Fd)

y2 = filter(b, a, x2)

[sos, g] = tf2sos(b, a)

[r p k] = residuez(b, a)

Bi=[ ; ]

Ai = [ ; ]

for i = 1: 2: floor(n/2)\*2

[bi ai] = residuez(r(i : i+1), p(i : i+1), 0)

Bi((i + 1)/2, : ) = bi

Ai((i + 1)/2, : ) = ai

end

global a;

global b;

global oldx;

global oldy;

max = 100;

nn=0:max-1;

x = eye(1,100);

[s t] = size(b);

oldx = zeros(1,t);

oldy = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterDirect(x(temp));

end

figure(10);

plot(nn,h1,'r',nn,y,'g');

f = 300;

max = 100;

nn=0:max-1;

x=sin(2\*pi\*(f/Fd)\*nn);

[s t] = size(b);

oldx = zeros(1,t);

oldy = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterDirect(x(temp));

end

figure(11);

plot(nn,x,'r',nn,y,'g',nn,y1,'b');

x = ones(1, 100);

[s t] = size(b);

oldx = zeros(1,t);

oldy = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterDirect(x(temp));

end

figure(12);

plot(nn,h2,'r',nn,y,'g');

f = 100;

max = 100;

nn=0:max-1;

x=sin(2\*pi\*(f/Fd)\*nn);

[s t] = size(b);

oldx = zeros(1,t);

oldy = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterDirect(x(temp));

end

figure(13);

plot(nn,x,'r',nn,y,'g',nn,y2,'b');

* + Параллельная форма

Fd = 1000

fp = 220

fs = 150

Rp = 0.2

Rs = 55

FdN = Fd / 2

Wp = fp / FdN

Ws = fs / FdN

[n, Wn] = buttord(Wp, Ws, Rp, Rs)

[b, a] = butter(n, Wn, 'high')

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

[h,w]=freqz(b,a);

ff = w/pi\*FdN;

% figure(5)

h1 = impz(b,a, 100);

% figure(6)

h2 = stepz(b,a, 100);

nn = (0:99)

f1 = 300

x1 = sin(2\*pi\*f1\*nn/Fd)

y1 = filter(b, a, x1)

f2 = 100

x2 = sin(2\*pi\*f2\*nn/Fd)

y2 = filter(b, a, x2)

[sos, g] = tf2sos(b, a)

[r p k] = residuez(b, a)

Bi=[ ; ]

Ai = [ ; ]

for i = 1: 2: floor(n/2)\*2

[bi ai] = residuez(r(i : i+1), p(i : i+1), 0)

Bi((i + 1)/2, : ) = bi

Ai((i + 1)/2, : ) = ai

end

global Ai;

global Bi;

global oldx;

global oldy;

global k;

max = 100;

nn=0:max-1;

x = eye(1,100);

[s t] = size(b);

oldx = 0;

oldy = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterParallel(x(temp));

end

figure(10);

plot(nn,h1,'r',nn,y,'g');

f = 300;

max = 100;

nn=0:max-1;

x=sin(2\*pi\*(f/Fd)\*nn);

[s t] = size(b);

oldx = 0;

oldy = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterParallel(x(temp));

end

figure(11);

plot(nn,x,'r',nn,y,'g',nn,y1,'b');

x = ones(1, 100);

[s t] = size(b);

oldx = 0;

oldy = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterParallel(x(temp));

end

figure(12);

plot(nn,h2,'r',nn,y,'g');

f = 100;

max = 100;

nn=0:max-1;

x=sin(2\*pi\*(f/Fd)\*nn);

[s t] = size(b);

oldx = 0;

oldy = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterParallel(x(temp));

end

figure(13);

plot(nn,x,'r',nn,y,'g',nn,y2,'b');

* + Последовательная форма

Fd = 1000

fp = 220

fs = 150

Rp = 0.2

Rs = 55

FdN = Fd / 2

Wp = fp / FdN

Ws = fs / FdN

[n, Wn] = buttord(Wp, Ws, Rp, Rs)

[b, a] = butter(n, Wn, 'high')

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

[h,w]=freqz(b,a);

ff = w/pi\*FdN;

% figure(5)

h1 = impz(b,a, 100);

% figure(6)

h2 = stepz(b,a, 100);

nn = (0:99)

f1 = 300

x1 = sin(2\*pi\*f1\*nn/Fd)

y1 = filter(b, a, x1)

f2 = 100

x2 = sin(2\*pi\*f2\*nn/Fd)

y2 = filter(b, a, x2)

[sos, g] = tf2sos(b, a)

[r p k] = residuez(b, a)

Bi=[ ; ]

Ai = [ ; ]

for i = 1: 2: floor(n/2)\*2

[bi ai] = residuez(r(i : i+1), p(i : i+1), 0)

Bi((i + 1)/2, : ) = bi

Ai((i + 1)/2, : ) = ai

end

global oldy;

global oldx;

global sos;

global g;

max = 100;

nn=0:max-1;

x = eye(1,100);

[s t] = size(b);

oldx = zeros(1,t);

oldy = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterSerial(x(temp));

end

figure(10);

plot(nn,h1,'r',nn,y,'g');

f = 300;

max = 100;

nn=0:max-1;

x=sin(2\*pi\*(f/Fd)\*nn);

[s t] = size(b);

oldx = zeros(1,t);

oldy = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterSerial(x(temp));

end

figure(11);

plot(nn,x,'r',nn,y,'g',nn,y1,'b');

x = ones(1, 100);

[s t] = size(b);

oldx = zeros(1,t);

oldy = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterSerial(x(temp));

end

figure(12);

plot(nn,h2,'r',nn,y,'g');

f = 100;

max = 100;

nn=0:max-1;

x=sin(2\*pi\*(f/Fd)\*nn);

[s t] = size(b);

oldx = zeros(1,t);

oldy = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterSerial(x(temp));

end

figure(13);

plot(nn,x,'r',nn,y,'g',nn,y2,'b');

* + Каноническая форма

Fd = 1000

fp = 220

fs = 150

Rp = 0.2

Rs = 55

FdN = Fd / 2

Wp = fp / FdN

Ws = fs / FdN

[n, Wn] = buttord(Wp, Ws, Rp, Rs)

[b, a] = butter(n, Wn, 'high')

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

[h,w]=freqz(b,a);

ff = w/pi\*FdN;

% figure(5)

h1 = impz(b,a, 100);

% figure(6)

h2 = stepz(b,a, 100);

nn = (0:99)

f1 = 300

x1 = sin(2\*pi\*f1\*nn/Fd)

y1 = filter(b, a, x1)

f2 = 100

x2 = sin(2\*pi\*f2\*nn/Fd)

y2 = filter(b, a, x2)

[sos, g] = tf2sos(b, a)

[r p k] = residuez(b, a)

Bi=[ ; ]

Ai = [ ; ]

for i = 1: 2: floor(n/2)\*2

[bi ai] = residuez(r(i : i+1), p(i : i+1), 0)

Bi((i + 1)/2, : ) = bi

Ai((i + 1)/2, : ) = ai

end

global a;

global b;

global oldw;

max = 100;

nn=0:max-1;

x = eye(1,100);

[s t] = size(b);

oldw = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterCanon(x(temp));

end

figure(10);

plot(nn,h1,'r',nn,y,'g');

f = 300;

max = 100;

nn=0:max-1;

x=sin(2\*pi\*(f/Fd)\*nn);

[s t] = size(b);

oldw = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterCanon(x(temp));

end

figure(11);

plot(nn,x,'r',nn,y,'g',nn,y1,'b');

x = ones(1, 100);

[s t] = size(b);

oldw = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterCanon(x(temp));

end

figure(12);

plot(nn,h2,'r',nn,y,'g');

f = 100;

max = 100;

nn=0:max-1;

x=sin(2\*pi\*(f/Fd)\*nn);

[s t] = size(b);

oldw = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterCanon(x(temp));

end

figure(13);

plot(nn,x,'r',nn,y,'g',nn,y2,'b');

* С фиксированной точкой
  + Прямая форма

Fd = 1000

fp = 220

fs = 150

Rp = 0.2

Rs = 55

FdN = Fd / 2

Wp = fp / FdN

Ws = fs / FdN

[n, Wn] = buttord(Wp, Ws, Rp, Rs)

[b, a] = butter(n, Wn, 'high')

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

[h,w]=freqz(b,a);

ff = w/pi\*FdN;

% figure(5)

h1 = impz(b,a, 100);

% figure(6)

h2 = stepz(b,a, 100);

nn = (0:99)

f1 = 300

x1 = sin(2\*pi\*f1\*nn/Fd)

y1 = filter(b, a, x1)

f2 = 100

x2 = sin(2\*pi\*f2\*nn/Fd)

y2 = filter(b, a, x2)

[sos, g] = tf2sos(b, a)

[r p k] = residuez(b, a)

Bi=[ ; ]

Ai = [ ; ]

for i = 1: 2: floor(n/2)\*2

[bi ai] = residuez(r(i : i+1), p(i : i+1), 0)

Bi((i + 1)/2, : ) = bi

Ai((i + 1)/2, : ) = ai

end

global a;

global b;

global oldx;

global oldy;

global m;

m = 32768;

a = round(a \* m);

b = round(b \* m);

max = 100;

nn=0:max-1;

x = eye(1,100);

x = round(x \* m);

[s t] = size(b);

oldx = zeros(1,t);

oldy = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterDirectInt(x(temp));

end

figure(10);

plot(nn,h1 \* m,'r',nn,y,'g');

f = 300;

max = 100;

nn=0:max-1;

x=sin(2\*pi\*(f/Fd)\*nn);

x=round(x\*m);

[s t] = size(b);

oldx = zeros(1,t);

oldy = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterDirectInt(x(temp));

end

figure(11);

plot(nn,x,'r',nn,y,'g',nn,y1 \* m,'b');

x = ones(1, 100);

x=round(x\*m);

[s t] = size(b);

oldx = zeros(1,t);

oldy = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterDirectInt(x(temp));

end

figure(12);

plot(nn,h2 \* m,'r',nn,y,'g');

f = 100;

max = 100;

nn=0:max-1;

x=sin(2\*pi\*(f/Fd)\*nn);

x=round(x\*m);

[s t] = size(b);

oldx = zeros(1,t);

oldy = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterDirectInt(x(temp));

end

figure(13);

plot(nn,x,'r',nn,y,'g',nn,y2 \* m,'b');

* + Параллельная форма

Fd = 1000

fp = 220

fs = 150

Rp = 0.2

Rs = 55

FdN = Fd / 2

Wp = fp / FdN

Ws = fs / FdN

[n, Wn] = buttord(Wp, Ws, Rp, Rs)

[b, a] = butter(n, Wn, 'high')

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

[h,w]=freqz(b,a);

ff = w/pi\*FdN;

% figure(5)

h1 = impz(b,a, 100);

% figure(6)

h2 = stepz(b,a, 100);

nn = (0:99)

f1 = 300

x1 = sin(2\*pi\*f1\*nn/Fd)

y1 = filter(b, a, x1)

f2 = 100

x2 = sin(2\*pi\*f2\*nn/Fd)

y2 = filter(b, a, x2)

[sos, g] = tf2sos(b, a)

[r p k] = residuez(b, a)

Bi=[ ; ]

Ai = [ ; ]

for i = 1: 2: floor(n/2)\*2

[bi ai] = residuez(r(i : i+1), p(i : i+1), 0)

Bi((i + 1)/2, : ) = bi

Ai((i + 1)/2, : ) = ai

end

global Ai;

global Bi;

global oldx;

global oldy;

global k;

m = 4048;

a = round(a \* m);

b = round(b \* m);

Ai = round(Ai \* m);

Bi = round(Bi \* m);

k = round(k \* m);

max = 100;

nn=0:max-1;

x = eye(1,100);

x = round(x \* m);

[s t] = size(b);

oldx = 0;

oldy = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterParallelInt(x(temp));

end

figure(10);

plot(nn,h1 \* m,'r',nn,y,'g');

f = 300;

max = 100;

nn=0:max-1;

x=sin(2\*pi\*(f/Fd)\*nn);

x = round(x \* m);

[s t] = size(b);

oldx = 0;

oldy = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterParallelInt(x(temp));

end

figure(11);

plot(nn,x,'r',nn,y,'g',nn,y1,'b');

x = ones(1, 100);

x = round(x \* m);

[s t] = size(b);

oldx = 0;

oldy = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterParallelInt(x(temp));

end

figure(12);

plot(nn,h2 \* m,'r',nn,y,'g');

f = 100;

max = 100;

nn=0:max-1;

x=sin(2\*pi\*(f/Fd)\*nn);

x = round(x \* m);

[s t] = size(b);

oldx = 0;

oldy = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterParallelInt(x(temp));

end

figure(13);

plot(nn,x,'r',nn,y,'g',nn,y2 \* m,'b');

* + Последовательная форма

Fd = 1000

fp = 220

fs = 150

Rp = 0.2

Rs = 55

FdN = Fd / 2

Wp = fp / FdN

Ws = fs / FdN

[n, Wn] = buttord(Wp, Ws, Rp, Rs)

[b, a] = butter(n, Wn, 'high')

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

[h,w]=freqz(b,a);

ff = w/pi\*FdN;

% figure(5)

h1 = impz(b,a, 100);

% figure(6)

h2 = stepz(b,a, 100);

nn = (0:99)

f1 = 300

x1 = sin(2\*pi\*f1\*nn/Fd)

y1 = filter(b, a, x1)

f2 = 100

x2 = sin(2\*pi\*f2\*nn/Fd)

y2 = filter(b, a, x2)

[sos, g] = tf2sos(b, a)

[r p k] = residuez(b, a)

Bi=[ ; ]

Ai = [ ; ]

for i = 1: 2: floor(n/2)\*2

[bi ai] = residuez(r(i : i+1), p(i : i+1), 0)

Bi((i + 1)/2, : ) = bi

Ai((i + 1)/2, : ) = ai

end

global oldy;

global oldx;

global sos;

global g;

global m;

m = 16384;

a = round(a \* m);

b = round(b \* m);

sos = round(sos \* m);

g = round (g \* m);

max = 100;

nn=0:max-1;

x = eye(1,100);

x = round(x \* m);

[s t] = size(b);

oldx = zeros(1,t);

oldy = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterSerialInt(x(temp));

end

figure(10);

plot(nn,h1 \* m,'r',nn,y,'g');

f = 300;

max = 100;

nn=0:max-1;

x=sin(2\*pi\*(f/Fd)\*nn);

x = round(x \* m);

% ?????? ?????

[s t] = size(b);

oldx = zeros(1,t);

oldy = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterSerialInt(x(temp));

end

figure(11);

plot(nn,x,'r',nn,y,'g',nn,y1 \* m,'b');

x = ones(1, 100);

x = round(x \* m);

[s t] = size(b);

oldx = zeros(1,t);

oldy = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterSerialInt(x(temp));

end

figure(12);

plot(nn,h2 \* m,'r',nn,y,'g');

f = 100;

max = 100;

nn=0:max-1;

x=sin(2\*pi\*(f/Fd)\*nn);

x = round(x \* m);

[s t] = size(b);

oldx = zeros(1,t);

oldy = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterSerialInt(x(temp));

end

figure(13);

plot(nn,x,'r',nn,y,'g',nn,y2 \* m,'b');

* + Каноническая форма

Fd = 1000

fp = 220

fs = 150

Rp = 0.2

Rs = 55

FdN = Fd / 2

Wp = fp / FdN

Ws = fs / FdN

[n, Wn] = buttord(Wp, Ws, Rp, Rs)

[b, a] = butter(n, Wn, 'high')

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

[h,w]=freqz(b,a);

ff = w/pi\*FdN;

% figure(5)

h1 = impz(b,a, 100);

% figure(6)

h2 = stepz(b,a, 100);

nn = (0:99)

f1 = 300

x1 = sin(2\*pi\*f1\*nn/Fd)

y1 = filter(b, a, x1)

f2 = 100

x2 = sin(2\*pi\*f2\*nn/Fd)

y2 = filter(b, a, x2)

[sos, g] = tf2sos(b, a)

[r p k] = residuez(b, a)

Bi=[ ; ]

Ai = [ ; ]

for i = 1: 2: floor(n/2)\*2

[bi ai] = residuez(r(i : i+1), p(i : i+1), 0)

Bi((i + 1)/2, : ) = bi

Ai((i + 1)/2, : ) = ai

end

global a;

global b;

global oldw;

global m;

m = 4096;

a = round(a\*m);

b = round(b\*m);

max = 100;

nn=0:max-1;

x = eye(1,100);

x = round(x \* m);

[s t] = size(b);

oldw = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterCanonInt(x(temp));

end

figure(10);

plot(nn,h1 \* m,'r',nn,y,'g');

f = 300;

max = 100;

nn=0:max-1;

x=sin(2\*pi\*(f/Fd)\*nn);

x = round(x \* m);

[s t] = size(b);

oldw = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterCanonInt(x(temp));

end

figure(11);

plot(nn,x,'r',nn,y,'g',nn,y1 \* m,'b');

x = ones(1, 100);

x = round(x \* m);

[s t] = size(b);

oldw = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterCanonInt(x(temp));

end

figure(12);

plot(nn,h2 \* m,'r',nn,y,'g');

f = 100;

max = 100;

nn=0:max-1;

x=sin(2\*pi\*(f/Fd)\*nn);

x = round(x \* m);

[s t] = size(b);

oldw = zeros(1,t);

y = zeros(1,max);

for temp=1:max

y(temp) = filterCanonInt(x(temp));

end

figure(13);

plot(nn,x,'r',nn,y,'g',nn,y2 \* m,'b');

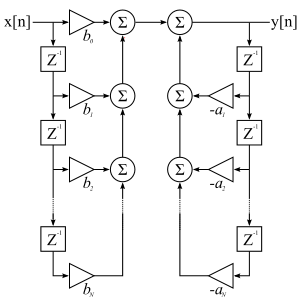
**Вывод:** изучил основные операций цифровой обработки сигналов, базирующихся на теории линейных дискретных систем и ознакомился с функциями пакета Signal Processing Toolbox.

**Ответы на вопросы**

**Фильтры:**

БИХ-фильтр (Бесконечно импульсная характеристика) – этот фильтр имеет “бесконечную” импульсную характеристику, т.е их выходной сигнал зависит от бесконечного числа предыдущих входных сигналов и выходных сигналов. Она имеет рекурсивную структуру, что позволяет иметь большую производительность и точность с меньшим кол-вом коэффициентом. Однако этот фильтр может быть не устойчивым.

*Примеры: Чебышев, Баттерворта, Калмана и Бесселя.*



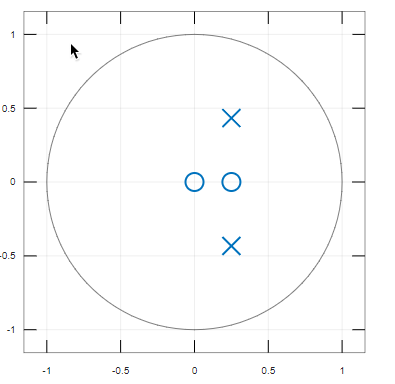
КИХ-фильтр (Конечно импульсная характеристика) – этот фильтр имеет “конечную” импульсную характеристику, т.е их входной сигнал зависит от текущих и предыдущих входных сигналов. Они не имеют рекурсивной структуры, поэтому всегда устойчивы. Требуют больше коэффициентов для достижения той же точности и затратны с точки зрения вычислений.

*Пример: оконные фильтры, линейно-фазовые фильтры и многофазовые фильтры.*



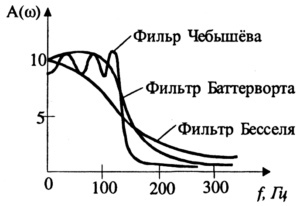
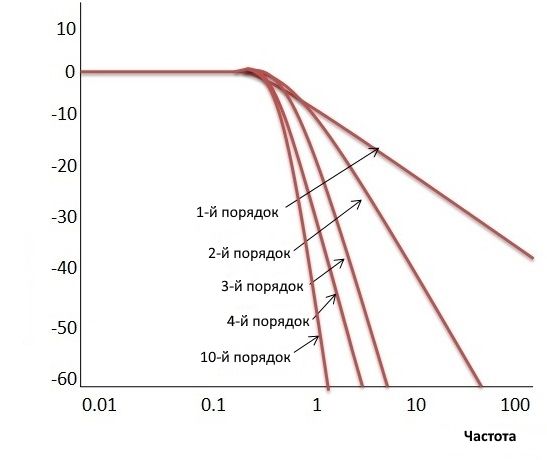
Сравнение:

* **Структура:** 
  + БИХ-фильтры имеют рекурсивную структуру, т.е он использует свою предыдущую выходную информацию для формирования нового входного сигнала. Может привести к накоплению ошибок округления и возникновению ненужных колебаний.
  + КИХ-фильтр нет, т.е он использует только свою текущую входную
* **Стабильность:**
  + БИХ-фильтр не всегда является устойчивым, т. к. полюса могут выйти за пределы, единичного круга на z-плоскости (Из-за рекурсивной структуры, если коэффициенты фильтра не выбраны правильно, система может неконтролируема усиливать частоты, что приводит к перегрузке, формируется обратная связь внутри системы).
  + КИХ-фильтр всегда устойчивы, не имеют полюсов и их выход полностью определяется входным сигналом.



* **Линейность фазы:** 
  + БИХ-фильтр, из-за рекурсивной структуры, фазовая характеристика обычно нелинейна. Это означает, что частотные компоненты входного сигнала могут быть задержены на разное время, что может привести к искажению. Зависит от частоты среза и сложная для контроля.
  + КИХ-фильтр могут быть легко спроектированы т.к чтобы обеспечить линейность фазы, что означает, что все частотные компоненты входного сигнала задерживаются на одинаковое время. Это приводит к меньшим искажениям. Не зависит от частоты среза и легка для контроля.
* **Амплитудная характеристика:**
  + БИХ-фильтр могут достичь более крутого спада амплитуды при меньшем порядке.
  + КИХ-фильтр может достичь произвольной амплитудной характеристики при большом порядке

**Фильтр Баттерворта:** это фильтр с максимально плоской частотной характеристикой этот тип цифрового или аналогового фильтра, который спроектирован таким образом, чтобы иметь амплитудно-частотную характеристику в полосе пропускания.



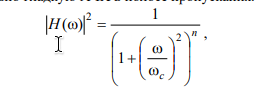
Ключевые особенности:

* Обеспечивает максимально плоскую частотную характеристику в полосе пропускания и затухания.
* Фильтр Баттерворта не имеет “пиков” или “провалов” в его частотной характеристике, что делает его очень полезным во многих приложениях.
* Фильтр имеет монотонное убывание амплитуды с увеличением частоты вне полосы пропускания.

Недостатка фильтра:

* Непропорциональная частоте фазовая характеристика уже в начале области пропускания, что влечет искажение сигнала по времени. Комплексные полюса передаточной функции фильтра, что вслечет колебания с чрезмерной амплитудой и более продолжительное установление колебаний по переходной характеристике.

Фильтр Баттерворта может быть разработан для работы с низкочастотными, высокочастотными, полосовыми фильтрами.



n – порядок фильтра, – частота среза.

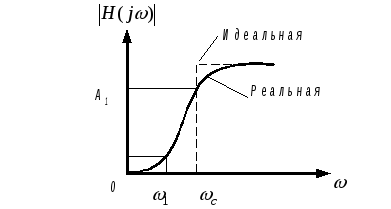
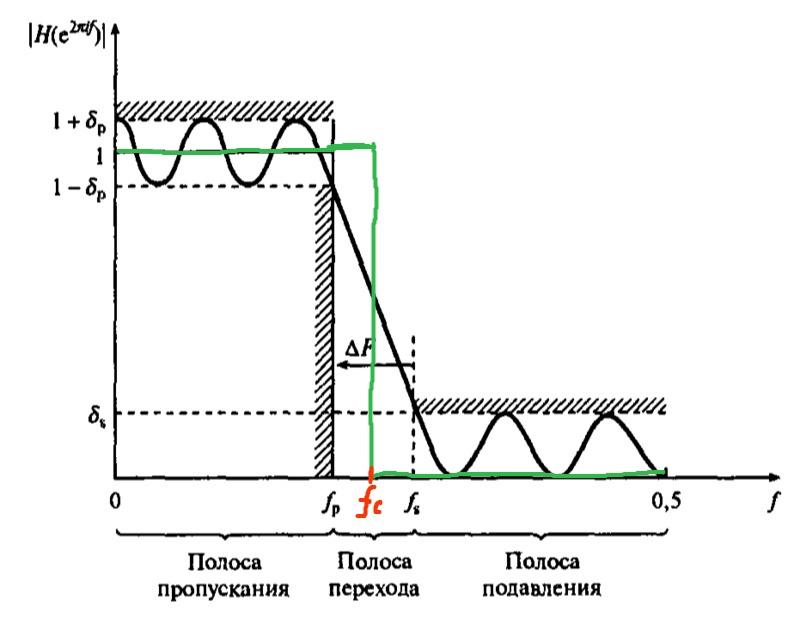
Функции фильтра:

[n, W n ]=buttord(W p , W s , R p , R s )

[b,a] = butter(n, W n , options)

**ФВЧ – фильтр верхних частот. Какие частоты.**

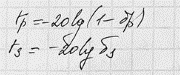
ФВЧ – тип фильтра, который пропускает сигналы с частотой выше определенного порогового значения, известного как частота среза, и ослабляет (полностью отсекает) сигналы с частотой ниже этого порога.

Графики:  
 

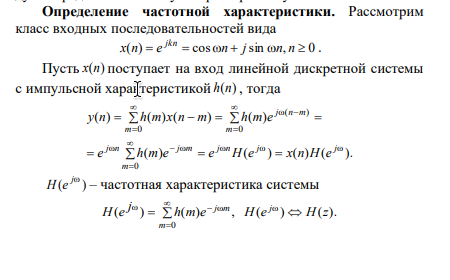
Идеальный фильтр построить невозможно. Для решения задачи аппроксимации должны быть заданы частоты среза полосы пропускания и полосы непропускания и допустимые пульсации АЧХ в полосе пропускания и полосы непропускания.

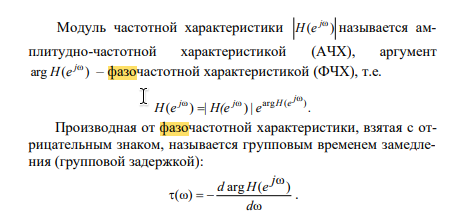
Rp – пульсация в полосе пропускания

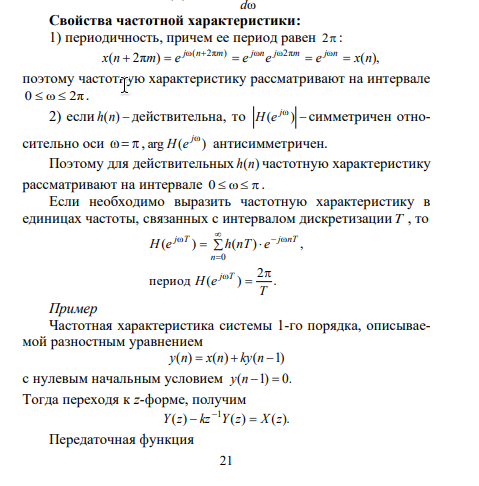
Rs – пульсация в полосе непропускания

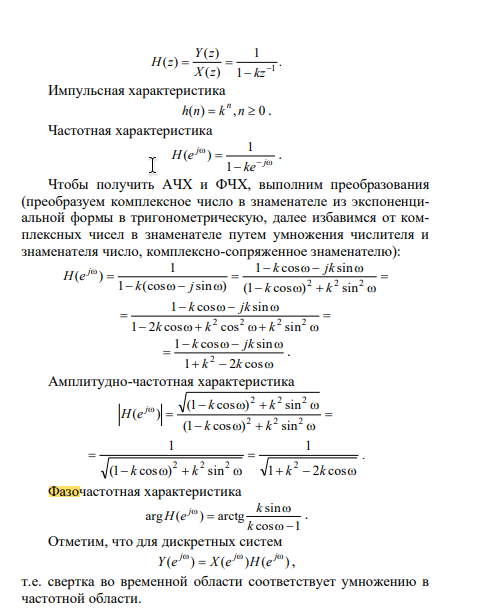


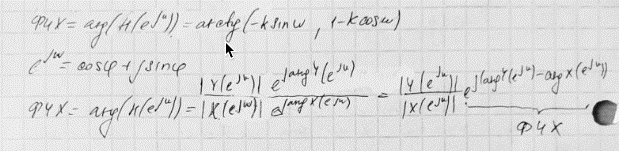
1. ФЧХ (Фазово-частотная характеристика):
   1. Определение: ФЧХ представляет собой зависимость фазы сигнала от его частоты.
   2. Физический смысл: ФЧХ позволяет оценить, как фаза сигнала изменяется в зависимости от его частоты. Это важно при работе с фильтрами, электрическими цепями и системами, где фазовые соотношение сигнала может влиять на результат.
2. АЧХ (Амлитудно-часточной характеристики):
   1. Определение: АЧХ отображает зависимость амплитуды сигнала от его частоты.
   2. Физический смысл: АЧХ позволяет оценить, как амплитуды сигнала изменяется в зависимости от его частоты. Это особенно полезно при анализе систем передачи сигналов, таких как аудио и видеоаппаратура, радиосвязь и другие системы, где важна передача сигнала с определенным диапазоном частот.



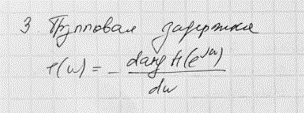








1. Групповая задержка

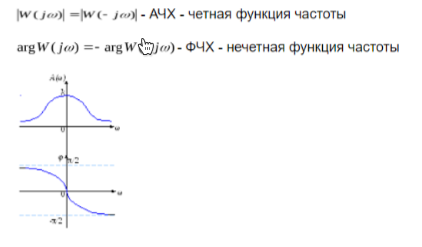


Показывает, насколько сигнал задерживает на определенной частоте при прохождении через фильтр

1. Импульсная характеристика:
   1. Определение: Импульсная характеристика описывает отклик системы на единичный импульс входного сигнала
   2. Физический смысл: Импульсная характеристика позволяет оценить, как система реагирует на быстрое воздействие или скачкообразное изменение сигнала. Например, в электронике импульсная характеристика может помочь определить, как система фильтрует или задерживает сигналы различной длительности.



1. Временная характеристика:
   1. Определение: Временная характеристика отражает изменение сигнала или системы во времени
   2. Физический смысл: Временная характеристика позволяет анализировать динамику сигналов или системы во времени. Например, в физике временная характеристика может помочь понять, как тело движется или как изменится физическая величина в зависимости от времени.



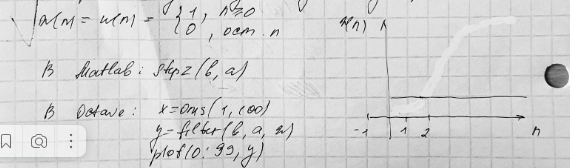
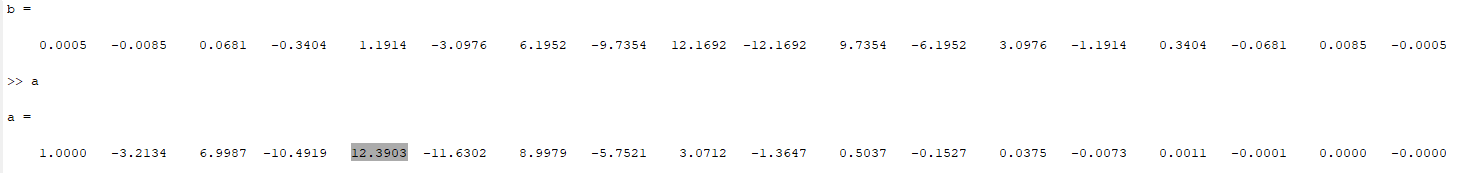
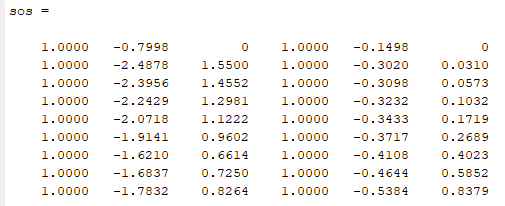
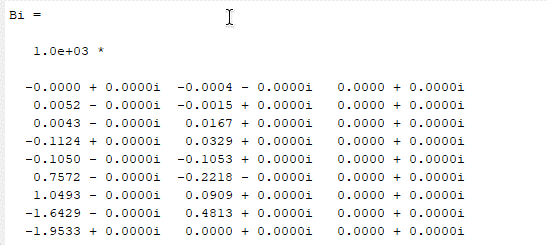


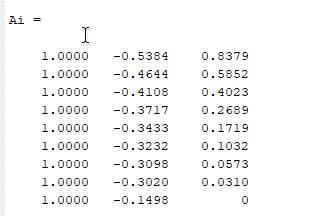
Таблица с динамическими коэффициентов в различных формах:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Прямая форма | Последовательная форма | Параллельная форма |
|  | 0.0681 | 0.6614 | 0.043 |
|  | 12.1692 | 2.4878 | 1.9533 |
|  | 178.696 | 3.761 | 45.425 |
|  | 0.0375 | 0.0310 | 0.0310 |
|  | 12.3903 | 1.0 | 1.0 |
|  | 330.408 | 32.258 | 32.258 |









**Прямая и последовательные** формы – два основных метода реализации цифровых фильтров. Они обеспечивают разные структуры, которые могут быть более или менее эффективными в зависимости от конкретных требований приложения.

Прямая форма:

* Простая реализация: прямая форма обычно проста в реализации, поскольку она напрямую следует из разностного уравнения фильтра.
* Эффективность: прямая форма может потребовать больше операций и памяти, чем каскадная форма, особенно для фильтров с большим числом полюсов и нулей.
* Устойчивость: прямая форма может быть менее устойчива с точки зрения количественных характеристик, таких как округление и переполнение.

Последовательная форма:

* Эффективность: каскадная форма обычно более эффективна с точки зрения использования ресурсов, поскольку она разбивает фильтр на серию второго порядка (или меньше) секций, каждая их которых общее кол-во операций и использование памяти.
* Устойчивость: каскадная форма может быть более устойчива с точки зрения количественных отдельно, что снижает вероятность переполнения и других проблем.
* Сложность: основным недостатком каскадной формы является ее сложность в реализации, поскольку требуется фильтр на отдельные секции и корректно реализовать каждую из них.