Лабораторна робота №2

з дисципліни

Цифрова обробка сигналів та зображень

на тему

«Розробка БІХ-фільтра»

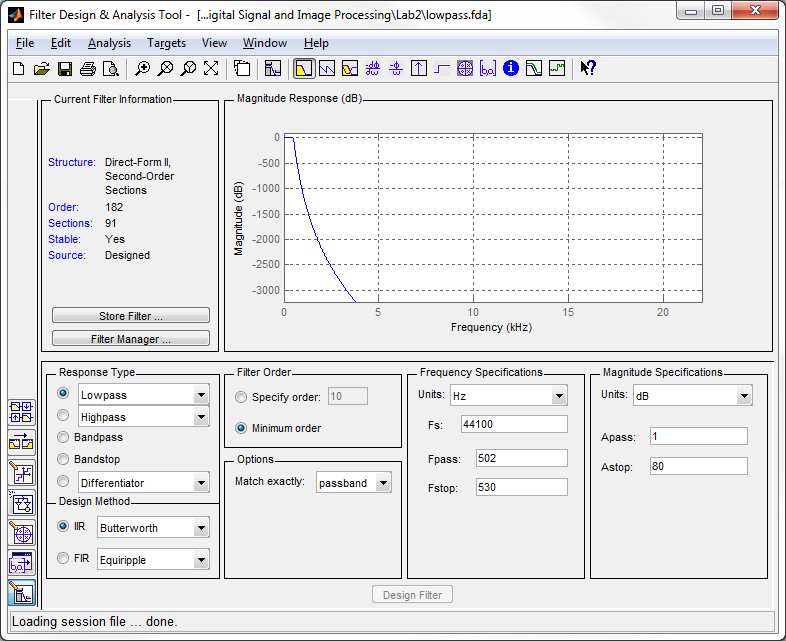
Варіант №17

|  |  |
| --- | --- |
| Виконав студент  групи КВ-64М  Подольський С. В.  залікова книжка № КВ6415 | Перевірив:  Онай М. В.  \_\_\_\_\_\_\_\_\_ |

1. Розробити відповідно до варіанту чотири БІХ-фільтри: фільтр низьких частот (НЧ), смугопропускальний фільтр середньо-низьких частот (СНЧ), смугопропускальний фільтр середньо-високих частот (СВЧ), фільтр високих частот (ВЧ).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | Фільтр НЧ | | | Фільтр СНЧ | | | | Фільтр СВЧ | | | | Фільтр ВЧ | | |
| назва | частота зрізу, Гц | спосіб | назва | частота зрізу 1, Гц | частота зрізу 2, Гц | спосіб | назва | частота зрізу 1, Гц | частота зрізу 2, Гц | спосіб | назва | частота зрізу, кГц | спосіб |
| 17 | Баттерворта | 502 | 1 | Чебишева І | 700 | 7000 | 4 | Чебишева ІІ | 7700 | 14900 | 3 | еліптичний | 16 | 2 |

1. Виконати верифікацію розроблених фільтрів. Якщо деякий фільтр є нестабільним, змінити його порядок та повторно виконати верифікацію.
2. Виконати фільтрацію музичного файлу формату .wav (файл вибрати самостійно, приблизна тривалість звучання – 1 хвилина), застосувавши кожен з розроблених фільтрів, та отримати чотири відфільтровані сигнали.
3. Впевнитись, що додавання чотирьох отриманих відфільтрованих сигналів відновлює вхідний сигнал.
4. Проаналізувати класичні фільтри, на основі яких були побудовані задані БІХ-фільтри, з точки зору доцільності застосування в кожному з розглянутих випадків.
5. Пояснити вибір тих параметрів побудованих БІХ-фільтрів, які не були задані варіантом.
6. Фільтр низьких частот (НЧ):



Смугопропускальний фільтр середньо-низьких частот (СНЧ):

function Hd = sublowpass( Fsamp )

%SUBLOWPASS Returns a discrete-time filter object.

if nargin < 1 % If there is no input arguments

Fsamp = 44100; % Sampling Frequency

end

Fst1 = 600; % Stopband Frequency 1

Fp1 = 700; % Passband Frequency 1

Fp2 = 7000; % Passband Frequency 2

Fst2 = 7100; % Stopband Frequency 2

Ast1 = 80; % Stopband Attenuation 1 (dB)

Ast2 = 80; % Stopband Attenuation 2 (dB)

Ap = 1; % Passband Ripple (dB)

d = fdesign.bandpass(Fst1, Fp1, Fp2, Fst2, Ast1, Ap, Ast2, Fsamp);

Hd = design(d, 'cheby1');

Смугопропускальний фільтр середньо-високих частот (СВЧ):

function Hd = subhighpass( Fsamp )

%SUBHIGHPASS Returns a discrete-time filter object.

if nargin < 1 % If there is no input arguments

Fsamp = 44100; % Sampling Frequency

end

Fnyq = Fsamp / 2; % Nyquist frequency

Fst1 = 7600; % Stopband Frequency 1

Fp1 = 7700; % Passband Frequency 1

Fp2 = 14900; % Passband Frequency 2

Fst2 = 15000; % Stopband Frequency 2

Wp = [ Fp1 Fp2 ] / Fnyq; % Normalized passband frequencies

Ws = [ Fst1 Fst2 ] / Fnyq; % Normalized stopband frequencies

Rp = 1; % Passband Ripple (dB)

Rs = 80; % Stopband Attenuation (dB)

[n, Ws] = cheb2ord(Wp, Ws, Rp, Rs); % n - order; Ws - cutoff frequencies

% Show that there is no 'cheby2' design method in any specification with N

% designmethods(fdesign.bandpass('Fst1,Fp1,Fp2,Fst2,Ast1,Ap,Ast2'))

% designmethods(fdesign.bandpass('N,F3dB1,F3dB2'))

% designmethods(fdesign.bandpass('N,Fc1,Fc2'))

% designmethods(fdesign.bandpass('N,Fc1,Fc2,Ast1,Ap,Ast2'))

% designmethods(fdesign.bandpass('N,Fp1,Fp2,Ap'))

% designmethods(fdesign.bandpass('N,Fp1,Fp2,Ast1,Ap,Ast2'))

% designmethods(fdesign.bandpass('N,Fst1,Fp1,Fp2,Fst2'))

% designmethods(fdesign.bandpass('N,Fst1,Fp1,Fp2,Fst2,Ap'))

n = 2 \* round(n / 2); % Nearest greater even integer

d = fdesign.bandpass('N,F3dB1,F3dB2', n, Ws(1), Ws(2));

%designmethods(d, 'full')

Hd = design(d); %Hd = design(d, 'cheby2'); % 'cheby2' is invalid method

Фільтр високих частот (ВЧ):

function Hd = highpass( Fsamp )

%HIGHPASS Returns a discrete-time filter object.

% Detailed explanation goes here

if nargin < 1 % If there is no input arguments

Fsamp = 44100; % Sampling Frequency

end

Fnyq = Fsamp / 2; % Nyquist frequency

Fst = 15000; % Stopband Frequency

Fp = 16000; % Passband Frequency

Wp = Fp / Fnyq; % Normalized passband frequency

Ws = Fst / Fnyq; % Normalized stopband frequency

Rp = 1; % Passband Ripple (dB)

Rs = 80; % Stopband Attenuation (dB)

[n, Wp] = ellipord(Wp, Ws, Rp, Rs); % n - order; Ws - cutoff frequency

[z, p, k] = ellip(n, Rp, Rs, Wp, 'high');

[sos,g] = zp2sos(z, p, k); % Convert to SOS form

Hd = dfilt.df2tsos(sos,g); % Create a dfilt object

end

1. Верифікація розроблених фільтрів:

****

На карті нулів та полюсів видно, що передатні функції усіх побудованих квантизованих фільтрів не мають полюсів, що лежать за межами одиничного кола. Отже, фільтри є стабільними.

1. Виконання фільтрації музичного файлу формату .wav:

[x, Fsamp] = wavread('Boy is Fiction - The Bits In The Numbers');

f2 = sublowpass(Fsamp);

f3 = subhighpass(Fsamp);

f4 = highpass(Fsamp);

y1 = filter(f1, x);

y2 = filter(f2, x);

y3 = filter(f3, x);

y4 = filter(f4, x);

1. Додавання чотирьох отриманих відфільтрованих сигналів:

y = y1 + y2 + y3 + y4;

px = audioplayer(x, Fsamp);

py1 = audioplayer(y1, Fsamp);

py2 = audioplayer(y2, Fsamp);

py3 = audioplayer(y3, Fsamp);

py4 = audioplayer(y4, Fsamp);

py = audioplayer(y, Fsamp);

1. Аналіз класичних фільтрів, на основі яких були побудовані задані БІХ-фільтри, з точки зору доцільності застосування в кожному з розглянутих випадків:



Особливістю фільтра Чебишева першого роду є більш крутий спад АЧХ та значні пульсації АЧХ на частотах полоси пропускання та заглушення, ніж у фільтрів інших видів.

Фільтр Чебишева другого роду характеризується менш крутим спадом амплітудної характеристики, що призводить до збільшення числа компонентів. У нього відсутні пульсації в полосі пропускання, однак відсутні в полосі заглушення.

АЧХ фільтра Баттерворта максимально гладка на частотах пропускання та знижується практично до нуля на частотах полоси заглушення. При відображенні частотного відгуку фільтра Баттерворта на логарифмічній АФЧХ амплітуда знижується до мінус нескінченності н частотах полоси заглушення. АЧХ фільтра Баттерворта – монотонно спадна функція частоти.

Характерною особливістю еліптичного фільтра є пульсації АЧХ як в полосі пропускання, так і в полосі заглушення. Величина пульсацій в кожній із полос незалежна одна від одної. Іншою характерною особливістю такого фільтра є дуже крутий спад амплітудної характеристики, тому за допомогою цього фільтра можна досягати більш ефективного розділення частот, ніж у інших лінійних фільтрів.

Доцільність застосування даних фільтрів у кожному з випадків визначається насамперед вимогами до способу фільтрації діапазонів частот, а саме:

1. на скільки ефективно повинно досягатися розділення частот за амплітудою; чи повинні частоти затухати за амплітудою різко, або плавно;
2. на скільки потрібно зрізати деякі частоти, або лише приглушати;
3. який допустимий рівень пульсацій в полосі пропускання та в полосі заглушення
4. який допустимий порядок фільтра та його обчислювальна складність.
5. Вибір тих параметрів побудованих БІХ-фільтрів, які не були задані варіантом, пояснюється емпіричним дослідженням та пошуком оптимального зрізу частот, мінімального значення пульсацій в полосі пропускання та приглушення за умови раціонально допустимого з точки зору обчислювальної складності порядку фільтрів, а також умови їх стабільності.

**Додаток 1:** скрипт автоматизованого створення фільтрів та їх запуску для обробки завантаженого сигналу

clear variables

clc

[x, Fsamp] = wavread('Boy is Fiction - The Bits In The Numbers');

%fdatool

f1 = lowpass(Fsamp);

f2 = sublowpass(Fsamp);

f3 = subhighpass(Fsamp);

f4 = highpass(Fsamp);

fvtool(f1, f2, f3, f4);

y1 = filter(f1, x);

y2 = filter(f2, x);

y3 = filter(f3, x);

y4 = filter(f4, x);

y = y1 + y2 + y3 + y4;

px = audioplayer(x, Fsamp);

py1 = audioplayer(y1, Fsamp);

py2 = audioplayer(y2, Fsamp);

py3 = audioplayer(y3, Fsamp);

py4 = audioplayer(y4, Fsamp);

py = audioplayer(y, Fsamp);