

گزارش پروژه درس پردازش سیگنال دیجیتال (DSP)

طراحی فیلترهای FIR و IIR

- سروش طیبی آراسته
- دانشجوی کارشناسی مهندسی برق دانشگاه بوعلی سینا
- استاد درس : دکتر علی پور
- نמיسال دوم تحصیلی 1394-1395 (2016)
- <https://github.com/starasteh/>

چکیده:

این پروژه شامل طراحی فیلترهای FIR و IIR خواسته شده و انجام آزمایش های خواسته شده بر روی آن ها می باشد . ابتدا صورت مساله نوشته شده و سپس به حل آن می پردازیم. همچنین در برنامه های نوشته شده از توابع تدریس شده در کلاس از جمله `conv_m` , `stepseq` , `sigmult` و ... استفاده شده است .

بخش اول: طراحی فیلترهای FIR و IIR

مشخصات فیلتر مطلوب :

$$0.97 \leq |H(e^{jw})| \leq 1.1, \quad 0 \leq w \leq 0.5\pi$$

$$|H(e^{jw})| \leq 0.08, \quad 0.65\pi \leq w \leq \pi$$

الف- هدف : طراحی فیلتر IIR

از تقریب Butterworth و روش Bilinear Transformation سوال را حل می کنیم .

قبل از شروع مراحل ، اندازه را نرمال سازی می کنیم ، یعنی تمام اندازه ها را بر بزرگترین مقدار یعنی 1.1 تقسیم می کنیم و در نهایت پس از طراحی فیلتر را در 1.1 ضرب می کنیم :

$$0.88 \leq |H(e^{jw})| \leq 1, \quad 0 \leq w \leq 0.5\pi$$

$$|H(e^{jw})| \leq 0.07, \quad 0.65\pi \leq w \leq \pi$$

در نتیجه داریم :

$$\begin{aligned} R_p &= 1.1103; & \% \text{ Passband ripple in dB} \\ A_s &= 23.098; & \% \text{ Stopband attenuation in dB} \end{aligned}$$

ابتدا به صورت دستی پارامترهای فیلتر پیوسته در زمان نمونه ، شامل N و W_c را تعیین می کنیم .

$$\frac{1}{\sqrt{1 + \left(\frac{2\pi}{W_c}\right)^{2N}}} \geq 0.88$$

$$\frac{1}{\sqrt{1 + \left(\frac{2\pi}{W_c}\right)^{2N}}} \leq 0.07$$

با استفاده از دو رابطه بالا و گرد کردن N رو به بالا ، به دست می آید :

$$N=7, \quad W_c=2.14$$

در نتیجه در 14 $H_c(s)H_c(-s)$ قطب داریم که با زاویه مساوی از یکدیگر (degrees 360/14) و با شعاع 2.14 در s-plane قرار گرفته اند . باید قطب های سمت چپ محور موهومی را انتخاب کنیم . چون پایدار و علی می خواهیم .
قطب های انتخاب شده ، قطب های $H_c(s)$ یعنی تابع تبدیل فیلتر پیوسته در زمان نمونه است :

$$Z_1 = -0.20652 + j0.9523$$

$$Z_2 = -0.20652 - j0.9523$$

$$Z_3 = -0.6325 - j0.7751$$

$$Z_4 = -0.6325 + j0.7751$$

$$Z_5 = -0.9023 - j0.4252$$

$$Z_6 = -0.9023 + j0.4252$$

$$Z_7 = -1 + j0$$

$$H_c(s) = \frac{1}{(s - z_1)(s - z_2) \dots (s - z_7)}$$

حال باید $H(z)$ را به دست آوریم . برای این کار با نداشت زیر مستقیماً از $H_c(s)$ به $H(z)$ می رسم :

$$s = (2/T) \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right)$$

دوره T را برابر یک قرار می دهیم : $T=1$

در نتیجه به دست می آید :

$$H(z) = \frac{(z + 1)^7}{((2 - z_1)z - 2 - z_1)((2 - z_2)z - 2 - z_2) \dots ((2 - z_7)z - 2 - z_7)}$$

فرکانس ها هم به صورت زیر از گسسته به پیوسته تبدیل می شوند :

$$\begin{aligned} W_p &= (2/T) * \tan(w_p/2); & \% \text{ Prototype Passband freq} \\ W_s &= (2/T) * \tan(w_s/2); & \% \text{ Prototype Stopband freq} \end{aligned}$$

حال این مراحل را با دستورات آماده متلب انجام می دهیم . با Run کردن برنامه IIR_project موارد خواسته شده به دست می آیند و همچنین نمودار های پاسخ ضربه ، پاسخ دامنه ، پاسخ لگاریتم دامنه و پاسخ تاخیر گروه فیلتر رسم می شود .

```
clear all
clc
close all
```

%% Digital Filter Specifications

```
wp = 0.5*pi;      % digital Passband freq in Hz
ws = 0.65*pi;     % digital Stopband freq in Hz
Rp = 1.1103;      % Passband ripple in dB
As = 23.098;      % Stopband attenuation in dB
```

%% Analog Prototype Specifications: Inverse mapping for frequencies

```

T = 1; % Set T=1
Wp = (2/T)*tan(wp/2); % Prototype Passband freq
Ws = (2/T)*tan(ws/2); % Prototype Stopband freq

%% Analog Butterworth Prototype Filter Calculation
N = ceil((log10((10^(Rp/10)-1)/(10^(As/10)-1)))/(2*log10(Wp/Ws)));
fprintf('\n*** Butterworth Filter Order = %2.0f \n',N)
Wc = Wp/((10^(Rp/10)-1)^(1/(2*N)));

[z,p,k] = buttap(N);
p = p*Wc;
k = k*Wc^N;
B = real(poly(z));
bs = k*B;
as = real(poly(p));

%% Bilinear transformation
[b,a] = bilinear(bs,as,1/T);
b=b.*1.1; % ghablan baraye normal sazi bar 1.1 taghsim karde budim

%%
[H,w] = freqz(b,a);
grp = grpdelay(b,a,w);
subplot(231);plot(w/pi,abs(H));grid on;ylabel('Magnitude')
subplot(232);plot(w/pi,20*log10(abs(H) + eps));grid on;ylabel('Magnitude (dB)')
subplot(233);impz(b,a);grid on;
subplot(234);plot(w/pi,grp);grid on;ylabel('Group Delay')

```

ب- هدف : طراحی فیلتر FIR

ابتدا به صورت تحلیلی پارامترهای M و β پنجره Kaiser مناسب را به دست می آوریم .

در این جا هم مانند قسمت قبل ، قبل از شروع مراحل ، اندازه را نرمال سازی می کنیم ، یعنی تمام اندازه ها را بر بزرگترین مقدار یعنی 1.1 تقسیم می کنیم و در نهایت پس از طراحی فیلتر را در 1.1 ضرب می کنیم :

$$0.88 \leq |H(e^{jw})| \leq 1, \quad 0 \leq w \leq 0.5\pi$$

$$|H(e^{jw})| \leq 0.07, \quad 0.65\pi \leq w \leq \pi$$

5

```

tr_width = ws - wp;
M = ceil((As-7.95)/(2.285*tr_width)+1) + 1
beta = 0.5842*(As-21).^4+.07886*(As-21)
wc = (ws+wp)/2;

alpha = (M - 1)/2;
n = 0:1:M - 1;
m = n - alpha;
fc = wc/pi;
hd = fc*sinc(fc*m);

w_kai = kaiser(M,beta)';
h = hd .* w_kai;
h=h.*1.1;           % ghablan baraye normal sazi bar 1.1 taghsim karde budim

delta_w = 2*pi/1000;
Rp = -(min(db(1:1:wp/delta_w + 1)))           % Actual Passband Ripple Rp
As = -round(max(db(ws/delta_w + 1:1:501)))     % Min Stopband attenuation As

% plots
subplot(2,3,3); stem(n,h); grid on; title('Actual Impulse Response')
axis([0 M-1 -0.1 0.3]);
xlabel('n'); ylabel('h(n)')
[H,w] = freqz(h,1);
a=1; b=h;
grp = grpdelay(b,a,w);
subplot(231);plot(w/pi,abs(H));grid on;ylabel('Magnitude')
subplot(232);plot(w/pi,20*log10(abs(H) + eps));grid on;ylabel('Magnitude (dB)')
subplot(234);plot(w/pi,grp);grid on;ylabel('Group Delay')
subplot(236);impz(b,a);grid on;

```

حال به به دست آوردن پارامتر طول پنجره Hamming متناسب می پردازیم .

```

Deltaw=ws-wp= 0.15
M = ceil(6.6*pi/Deltaw) + 1 = (6.6/ 0.15)+1= 45
Delta=min(delta1,delta2)=0.07
A=-20log(Delta)=23.098

```

با قرار دادن M به دست آمده در رابطه $w[n]$ پنجره Hamming ، پنجره Hamming متناسب هم به دست می آید .
 حال این مراحل را با دستورات آماده متلب انجام می دهیم . با Run کردن برنامه FIR_project موارد خواسته شده به دست می آیند و همچنین نمودار های پاسخ ضربه ، پاسخ دامنه ، پاسخ لگاریتم دامنه و پاسخ تاخیر گروه فیلتر رسم می شود .

بخش دوم : اعمال ورودی خاص به سیگنال های طراحی شده

الف- یک ورودی سینوسی تک فرکانس به هر کدام از فیلتر ها اعمال می کنیم و عملکرد آن ها در باند های مختلف فرکانسی را بررسی می کنیم .

ابتدا به فیلتر IIR :

ورودی زیر را به فیلتر می دهیم و با اجرا کردن برنامه IIR_sine وضعیت خروجی را در باند های مختلف مشاهده می کنیم . (وضعیت عبور از باند های قطع و عبور)

$$x(n)=\cos(5n)$$

```
clear all
clc
close all
```

%% Digital Filter Specifications

```
wp = 0.5*pi;           % digital Passband freq in Hz
ws = 0.65*pi;          % digital Stopband freq in Hz
Rp = 1.1103;           % Passband ripple in dB
As = 23.098;           % Stopband attenuation in dB
```

%% Analog Prototype Specifications: Inverse mapping for frequencies

```
T = 1;                 % Set T=1
Wp = (2/T)*tan(wp/2);  % Prototype Passband freq
Ws = (2/T)*tan(ws/2);  % Prototype Stopband freq
```

%% Analog Butterworth Prototype Filter Calculation

```
N = ceil((log10((10^(Rp/10)-1)/(10^(As/10)-1)))/(2*log10(Wp/Ws)));
fprintf('\n*** Butterworth Filter Order = %2.0f \n',N)
Wc = Wp/((10^(Rp/10)-1)^(1/(2*N)));
```

```
[z,p,k] = buttap(N);
p = p*Wc;
```

```

k = k*Wc^N;
B = real(poly(z));
bs = k*B;
as = real(poly(p));

%% Bilinear transformation
[b,a] = bilinear(bs,as,1/T);
b=b.*1.1;           % ghablan baraye normal sazi bar 1.1 taghsim karde budim

%%
[H,w] = freqz(b,a);
grp = grpdelay(b,a,w);
subplot(231);plot(w/pi,abs(H));grid on;ylabel('Magnitude')
subplot(232);plot(w/pi,20*log10(abs(H) + eps));grid on;ylabel('Magnitude (dB)')
subplot(233);impz(b,a);grid on;
subplot(234);plot(w/pi,grp);grid on;ylabel('Group Delay')

%%

n=-50:50;
w0=5;           % For example
x=cos(w0*n);
k = 0:511; W = (pi/512)*k;
X = x*(exp(-1j*pi/512)).^(n'*k);
Y=X.*abs(H)';
subplot(235);plot(W/pi,abs(Y));grid on;ylabel(' output')

```

فیلتر FIR :

همان ورودی را به فیلتر FIR طراحی شده از روش Kaiser می دهیم و موارد فوق را دنبال می کنیم . از برنامه FIR_sine استفاده می کنیم :

```

clear all
clc
close all
% We do not use the passband ripple value of Rp =1.1103 dB in the

wp = 0.5*pi;
ws = 0.65*pi;
As = 23.098;
tr_width = ws - wp;
M = ceil((As-7.95)/(2.285*tr_width)+1) + 1
beta = 0.5842*(As-21).^4+.07886*(As-21)
wc = (ws+wp)/2;

```



```

alpha = (M - 1)/2;
n = 0:1:M - 1;
m = n - alpha;
fc = wc/pi;
hd = fc*sinc(fc*m);

w_kai = kaiser(M,beta)';
h = hd .* w_kai;
h=h.*1.1;           % ghablan baraye normal sazi bar 1.1 taghsim karde budim

```

```

delta_w = 2*pi/1000;
Rp = -(min(db(1:1:wp/delta_w + 1))) % Actual Passband Ripple Rp
As = -round(max(db(ws/delta_w + 1:1:501))) % Min Stopband attenuation As

```

```

%% plots
subplot(2,3,3); stem(n,h); grid on; title('Actual Impulse Response')
axis([0 M-1 -0.1 0.3]);
xlabel('n'); ylabel('h(n)')
[H,w] = freqz(h,1);
a=1; b=h;
grp = grpdelay(b,a,w);
subplot(231);plot(w/pi,abs(H));grid on;ylabel('Magnitude')
subplot(232);plot(w/pi,20*log10(abs(H) + eps));grid on;ylabel('Magnitude (dB)')
subplot(234);plot(w/pi,grp);grid on;ylabel('Group Delay')
subplot(236);impz(b,a);grid on;

```

```

%%

```

```

n=-50:50;
w0=5; % For example
x=cos(w0*n);
k = 0:511; W = (pi/512)*k;
X = x*(exp(-1j*pi/512)).^(n'*k);
Y=X.*abs(H)';
subplot(235);plot(W/pi,abs(Y));grid on;ylabel(' output')

```

ب- یک سیگنال نویز سفید به عنوان ورودی به هر کدام از فیلتر ها اعمال می کنیم و عملکرد آن ها در باند های مختلف فرکانسی را بررسی می کنیم .

ابتدا فیلتر IIR :

چگالی طیف توان سیگنال برابر است با اندازه تبدیل فوریه آن به توان 2 . در نتیجه ابتدا تبدیل فوریه فیلتر را باید داشته باشیم ، یعنی همان پاسخ دامنه .

از برنامه IIR_noise استفاده می کنیم :

```
clear all
clc
close all

%% Digital Filter Specifications
wp = 0.5*pi;          % digital Passband freq in Hz
ws = 0.65*pi;         % digital Stopband freq in Hz
Rp = 1.103;           % Passband ripple in dB
As = 23.098;          % Stopband attenuation in dB

%% Analog Prototype Specifications: Inverse mapping for frequencies
T = 1;                % Set T=1
Wp = (2/T)*tan(wp/2); % Prototype Passband freq
Ws = (2/T)*tan(ws/2); % Prototype Stopband freq

%% Analog Butterworth Prototype Filter Calculation
N = ceil((log10((10^(Rp/10)-1)/(10^(As/10)-1)))/(2*log10(Wp/Ws)));
fprintf('\n*** Butterworth Filter Order = %2.0f \n',N)
Wc = Wp/((10^(Rp/10)-1)^(1/(2*N)));

[z,p,k] = buttap(N);
p = p*Wc;
k = k*Wc^N;
B = real(poly(z));
bs = k*B;
as = real(poly(p));

%% Bilinear transformation
[b,a] = bilinear(bs,as,1/T);
b=b.*1.1;          % ghablan baraye normal sazi bar 1.1 taghsim karde budim

%%
[H,w] = freqz(b,a);
grp = grpdelay(b,a,w);
subplot(231);plot(w/pi,abs(H));grid on;ylabel('Magnitude')
subplot(232);plot(w/pi,20*log10(abs(H) + eps));grid on;ylabel('Magnitude (dB)')
subplot(233);impz(b,a);grid on;
subplot(234);plot(w/pi,grp);grid on;ylabel('Group Delay')
```

```

%%
n=-255:256;

x = wgn(1,512,1); % noise sefide gausi ba tavane 1
k = 0:511; W = (pi/512)*k;
X = x*(exp(-1j*pi/512)).^(n'*k);

Y=X.*abs(H)';
subplot(235);plot(W/pi,abs(Y));grid on;ylabel(' output')

G=abs(H).*abs(H); % chegali teyfe tavane filter
Gy=abs(Y).*abs(Y);
Gy=Gy'; % chegali teyfe tavane khoruji

```

فیلتر FIR :

برای FIR هم مانند قسمت قبل عمل می کنیم . (روی فیلتر تولید شده با روش Kaiser)

```

clear all
clc
close all
% We do not use the passband ripple value of Rp =1.1103 dB in the

wp = 0.5*pi;
ws = 0.65*pi;
As = 23.098;
tr_width = ws - wp;
M = ceil((As-7.95)/(2.285*tr_width)+1) + 1
beta = 0.5842*(As-21).^4+.07886*(As-21)
wc = (ws+wp)/2;

alpha = (M - 1)/2;
n = 0:1:M - 1;
m = n - alpha;
fc = wc/pi;
hd = fc*sinc(fc*m);

w_kai = kaiser(M,beta)';
h = hd .* w_kai;
h=h.*1.1; % ghablan baraye normal sazi bar 1.1 taghsim karde budim

delta_w = 2*pi/1000;

```

```
Rp = -(min(db(1:1:wp/delta_w + 1))) % Actual Passband Ripple Rp
As = -round(max(db(ws/delta_w + 1:1:501))) % Min Stopband attenuation As
```

```
%% plots
```

```
subplot(2,3,3); stem(n,h); grid on; title('Actual Impulse Response')
axis([0 M-1 -0.1 0.3]);
xlabel('n'); ylabel('h(n)')
[H,w] = freqz(h,1);
a=1; b=h;
grp = grpdelay(b,a,w);
subplot(231);plot(w/pi,abs(H));grid on;ylabel('Magnitude')
subplot(232);plot(w/pi,20*log10(abs(H) + eps));grid on;ylabel('Magnitude (dB)')
subplot(234);plot(w/pi,grp);grid on;ylabel('Group Delay')
subplot(236);impz(b,a);grid on;
```

```
%%
```

```
n=-255:256;
```

```
x = wgn(1,512,1); % noise sefide gausi ba tavane 1
k = 0:511; W = (pi/512)*k;
X = x*(exp(-1j*pi/512)).^(n'*k);
```

```
Y=X.*abs(H)';
subplot(235);plot(W/pi,abs(Y));grid on;ylabel(' output')
```

```
G=abs(H).*abs(H); % chegali teyfe tavane filter
Gy=abs(Y).*abs(Y);
Gy=Gy'; % chegali teyfe tavane khoruji
```

جمع بندی :

در این پروژه دو نوع فیلتر خاص را به دو صورت دستی و با متلب طراحی کردیم و آزمایش های مختلفی روی آن ها انجام دادیم . به وضوح مشاهده شد که متلب ابزاری بسیار کارآمد برای طراحی انواع فیلتر و همچنین باتوجه به بخش اول پروژه ، بسیار مناسب برای انواع کار های پردازش سیگنال می باشد . حجم محاسبات بالا در محاسباتی که به صورت دستی انجام دادیم یکی دیگر از مزیت های متلب را در این امر مشخص کرد تا به وسیله آن دقت و سرعت خود را در طراحی افزایش دهیم .

همچنین مشاهده شد که با متلب می توانیم به راحتی نمودار های پاسخ ضربه ، پاسخ دامنه ، پاسخ لگاریتم دامنه و پاسخ تاخیر گروه فیلتر را مشاهده کنیم و با تغییر پارامتر ها ، آن ها را به مقدار مطلوب برسانیم .