JET Jurnal Elektro Teknik
Volume 1, No.1, Maret 2021, pp. 39-45

Simulasi Perancangan Filter Analog *Chebyshev Tipe I Menggunakan MATLAB*

Herry Satria Utama

Teknik Elektro Universitas Pakuan

Abstrak

Penggunaan rangkaian filter banyak dipakai di dalam sistem Telekomunikasi. Digunakan teknik simulasi dalam perancangan filter. Rancangan filter dengan simulasi ini bertujuan untuk dapat menghasilkan tanggapan filter Chebyshev dan mendapatkan besaran induktor dan kapasitor yang diperlukan untuk merancang penyaring (filter). Simulasi ini memakai Graphical User Interface (GUI). Hasil simulasi tersebut didapatkan tanggapan Chebyshev untuk LPF (Low Pass Filter), HPF (High Pass Filter), BPF (Band Pass Filter) dan BSF (Band Stop Filter) dan sesuai teori respon filter Chebyshev. Aplikasi yang digunakan untuk simulasi tersebut menggunakan Electronic Work Bench (EWB), menunjukkan frekuensi yang bergeser senilai 0,1 kHz

Kata Kunci : Chebyshev, LPF, HPF, BPF, BSF, respon frekuensi.

Abstract

The use of filter circuits is widely used in telecommunications systems. Simulation techniques used in filter design. The filter design with this simulation aims to produce a Chebyshev filter response and obtain the inductor and capacitor quantities needed to design the filter. This simulation uses a Graphical User Interface (GUI). From the simulation results, Chebyshev's responses are obtained for LPF (Low Pass Filter), HPF (High Pass Filter), BPF (Band Pass Filter), and BSF (Band Stop Filter) and according to Chebyshev filter response theory. The application used for the simulation uses an Electronic Work Bench (EWB), showing a shift frequency of 0.1 kHz Keywords: Chebyshev, LPF, HPF, BPF, BSF, frequency response.

Keywords: Chebyshev, LPF, HPF, BPF, BSF, frequency response.

1. PENDAHULUAN

Sistem filter sangat penting pada sistem komunikasi yaitu sisi pengirim maupun sisi penerima. Filter analog yang sering digunakan adalah filter *chebyshev*. Tanggapan *chebyshev* adalah Tanggapan filter yang memiliki *roll-off* yang curam dan *ripple*

Dapat digunakan simulasi untuk perancangan Filter dan dapat menjelaskan karakteristik dan respon dari suatu filter. Salah satu perangkat lunak yang dapat melakukan simulasi perancangan tersebut adalah Matlab. (saputro, 2009). satu kesatuan. Sehingga tidak terjadi

2. LANDASAN TEORI

Pengolahan sinyal filter adalah sebuah rangkaian elektronik yang berfungsi untuk menyaring frekuensi tertentu. Pita frekuensi yang diloloskan adalah pita pelolos dan pita frekuensi yang ditekan adalah pita penghalang. Terdapat berbagai jenis filter dengan fungsinya tertentu,

Filter yang biasa digunakan dalam pengolahan sinyal adalah filter Chebyshev. Filter Chebyshev ini memperlihatkan tanggapan berbentuk *equiripple* di daerah pita lolosnya dan menurun secara monoton di daerah pita penghalang. Bentuk ripple filter Chebyshev dibedakan atas dasar ganjil atau genap orde filternya. Filter berorde

ganjil memiliki penguatan magnituda sama dengan 1 pada $\omega = 0$, sedangkan untuk orde genap memiliki deviasi magnituda maksimum untuk $\omega = 0$. Jumlah total maksima dan minima di daerah passband sama dengan orde filternya (N). Semua zero transmisi dari filter Chebyshev terletak di $\omega = \infty$sehingga menghasilkan tipe all-pole filter.

3. PERANCANGAN FILTER ANALOG DENGAN RESPON CHEBYSHEV TIPE I MENGGUNAKAN MATLAB

Perancangan ini digunakan untuk mengetahui keluaran dari *Low Pass Filter, High Pass Filter, Band Pass Filter dan Band Stop Filter* menggunakan *recursion coefficient* 2 pole dan 4 pole.

3.1. Menggunakan *recursion coefficient 2 pole*

```
% --- Executes on button press in Result.
if get(handles.LPF ,'Value')
fc=400;
fp=200;
f=2000;
rp=0.5;
rs=20;
a0=6.362308E-01;
a1=1.272462E+00;
a2=6.362308E-01;
b1=-1.125379E+00;
b2=-4.195441E-01;

wp=2*(fp/f);
ws=2*(fc/f);
[n,wn]=cheb2ord(wp,ws,rp,rs);

x=a0*wp+a1*(wp-1)+a2*(wp-2);
y=ws-b1*(ws-1)-b2*(ws-2);
%wn=[wp,ws];
[b,a]=cheby1(n,rs,wn, 'low');
[h,w]=freqz(b,a);

plot(w/pi,angle(h),'-b');
xlabel('Normalized
Frequency','color','w','FontName','Berlin
Sans FB Demi','FontSize',10);
ylabel('Amplitudo','color','w','FontName'
,'Berlin Sans FB Demi','FontSize',10);
text(0.5,1,'Phase Response','color','b');
grid on;
box on;
axes(handles.axes1);

plot(w/pi,20*log(abs(h)),'-r');
xlabel('Normalized
Frequency','color','w','FontName','Berlin
Sans FB Demi','FontSize',10);
ylabel('Amplitudo','color','w','FontName'
,'Berlin Sans FB Demi','FontSize',10);
text(0.5,-80,'Magnitude
Response','color','r');
grid on;
box on;
axes(handles.axes2);

elseif get(handles.HPF ,'Value')
fc=400;
fp=200;
f=2000;
rp=0.5;
rs=20;
a0=6.362308E-01;
a1=1.272462E+00;
a2=6.362308E-01;
b1=-1.125379E+00;
b2=-4.195441E-01;

wp=2*(fp/f);
ws=2*(fc/f);
[n,wn]=cheb2ord(wp,ws,rp,rs);

x=a0*wp+a1*(wp-1)+a2*(wp-2);
y=ws-b1*(ws-1)-b2*(ws-2);
%wn=[wp,ws];
[b,a]=cheby1(n,rs,wn, 'high');
[h,w]=freqz(b,a);

plot(w/pi,angle(h),'-b');
grid on;
text(0.45,-0.8,'Phase
Response','color','b');
xlabel('Normalized
Frequency','color','w','FontName','Berlin
Sans FB Demi','FontSize',10);
ylabel('Amplitudo','color','w','FontName'
,'Berlin Sans FB
Demi','FontWeight','bold','FontSize',10);
axes(handles.axes1);

plot(w/pi,20*log(abs(h)),'-r');
```

```
grid on;
text(0.42,-70,'Magnitude
Response','color','r');
xlabel('Normalized
Frequency','color','w','FontName','Berlin
Sans FB Demi','FontSize',10);
ylabel('Amplitudo','color','w','FontName'
,'Berlin Sans FB Demi','FontSize',10);
axes(handles.axes2);

elseif get(handles.BPF ,'Value')
fc=400;
fp=200;
f=2000;
rp=0.5;
rs=20;
a0=6.362308E-01;
a1=1.272462E+00;
a2=6.362308E-01;
b1=-1.125379E+00;
b2=-4.195441E-01;

wp=2*(fp/f);
ws=2*(fc/f);
[n,wn]=cheb2ord(wp,ws,rp,rs);

x=a0*wp+a1*(wp-1)+a2*(wp-2);
y=ws-b1*(ws-1)-b2*(ws-2);
wn=[wp,ws];
[b,a]=cheby1(n,rs,wn, 'bandpass');
[h,w]=freqz(b,a);

plot(w/pi,angle(h),'-b');
xlabel('Normalized
Frequency','color','w','FontName','Berlin
Sans FB
Demi','FontWeight','bold','FontSize',10);
ylabel('Amplitudo','color','w','FontName'
,'Berlin Sans FB
Demi','FontWeight','bold','FontSize',10);
text(0.5,1,'Phase Response','color','b');
grid on;
axes(handles.axes1);

plot(w/pi,20*log(abs(h)),'-r');
xlabel('Normalized
Frequency','color','w','FontName','Berlin
Sans FB
Demi','FontWeight','bold','FontSize',10);
ylabel('Amplitudo','color','w','FontName'
,'Berlin Sans FB
Demi','FontWeight','bold','FontSize',10);
text(0.5,-300,'Magnitude
Response','color','r');
grid on;
axes(handles.axes2);

elseif get(handles.BSF ,'Value')
fc=400;
fp=200;
f=2000;
rp=0.5;
rs=20;
a0=6.362308E-01;
a1=1.272462E+00;
a2=6.362308E-01;
b1=-1.125379E+00;
b2=-4.195441E-01;
wp=2*(fp/f);
ws=2*(fc/f);
[n,wn]=cheb2ord(wp,ws,rp,rs);
x=a0*wp+a1*(wp-1)+a2*(wp-2);
y=ws-b1*(ws-1)-b2*(ws-2);
wn=[wp,ws];
[b,a]=cheby1(n,rs,wn, 'stop');
[h,w]=freqz(b,a);

plot(w/pi,angle(h),'-b');
xlabel('Normalized
Frequency','color','w','FontName','Berlin
Sans FB
Demi','FontWeight','bold','FontSize',10);
ylabel('Amplitudo','color','w','FontName'
,'Berlin Sans FB
Demi','FontWeight','bold','FontSize',10);
text(0.45,-1,'Phase Response','color','b');
grid on;
box on;
axes(handles.axes1);
plot(w/pi,20*log(abs(h)),'-r');
xlabel('Normalized
Frequency','color','w','FontName','Berlin
Sans FB
Demi','FontWeight','bold','FontSize',10);
ylabel('Amplitudo','color','w','FontName'
,'Berlin Sans FB
Demi','FontWeight','bold','FontSize',10);
text(0.45,-80,'Magnitude
Response','color','r');
grid on;
box on;
axes(handles.axes2);
end
```

3.2. Menggunakan *recursion coefficient 4 pole*

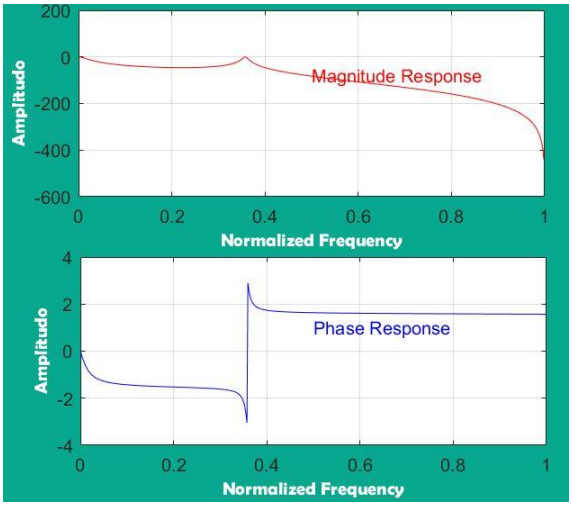
```
%Parameter has been known using
Recursion Coefficient 4 Pole
fc=400;
fp=200;
f=2000;
rp=0.5;
rs=20;
a0 =  2.780754E-03
a1 = -1.112302E-02
a2 =  1.668453E-02
a3 = -1.112302E-02
a4 =  2.780754E-03
b1 = -2.764031E+00
b2 = -3.122854E+00
b3 = -1.664554E+00
b4 = -3.502233E-01

%Formulation
wp=2*(fp/f);
ws=2*(fc/f);
[n,wn]=cheb2ord(wp,ws,rp,rs);
x=a0*wp+a1*(wp-1)+a2*(wp-
2)+a3*(wp-3)+a4*(wp-4);
y=ws-b1*(ws-1)-b2*(ws-2)-b3*(ws-3)-
b4*(ws-4);
wn=[wp,ws];

filter=get(handles.popupmenu2,'value');
if filter==1
[n,wn]=cheb2ord(wp,ws,rp,rs);
[b,a]=cheby1(n,rs,wn, 'low');
[h,w]=freqz(b,a);

%Plotting
plot(w/pi,angle(h),'-b');
xlabel('Normalized
Frequency','color','w','FontName','Berlin
Sans FB
Demi','FontWeight','bold','FontSize',10);

ylabel('Amplitudo','color','w','FontName'
,'Berlin Sans FB
Demi','FontWeight','bold','FontSize',10);
text(0.45,-1,'Phase Response','color','b');
grid on;
box on;
axes(handles.axes1);

plot(w/pi,20*log(abs(h)),'-r');
xlabel('Normalized
Frequency','color','w','FontName','Berlin
Sans FB
Demi','FontWeight','bold','FontSize',10);

ylabel('Amplitudo','color','w','FontName'
,'Berlin Sans FB
Demi','FontWeight','bold','FontSize',10);
text(0.45,-80,'Magnitude
Response','color','r');
grid on;
box on;
axes(handles.axes2);

elseif filter==2
[n,wn]=cheb2ord(wp,ws,rp,rs);
[b,a]=cheby1(n,rs,wn, 'high');
[h,w]=freqz(b,a);

%Plotting
plot(w/pi,angle(h),'-b');
xlabel('Normalized
Frequency','color','w','FontName','Berlin
Sans FB
Demi','FontWeight','bold','FontSize',10);

ylabel('Amplitudo','color','w','FontName'
,'Berlin Sans FB
Demi','FontWeight','bold','FontSize',10);
text(0.45,-1,'Phase Response','color','b');
grid on;
box on;
axes(handles.axes1);

plot(w/pi,20*log(abs(h)),'-r');
xlabel('Normalized
Frequency','color','w','FontName','Berlin
Sans FB
Demi','FontWeight','bold','FontSize',10);

ylabel('Amplitudo','color','w','FontName'
,'Berlin Sans FB
Demi','FontWeight','bold','FontSize',10);
text(0.45,-80,'Magnitude
Response','color','r');
grid on;
box on;
axes(handles.axes2);

elseif filter==3
[b,a]=cheby1(n,rs,wn, 'bandpass');
[h,w]=freqz(b,a);

%Plotting
plot(w/pi,angle(h),'-b');
xlabel('Normalized
Frequency','color','w','FontName','Berlin
```

```
Sans FB
Demi','FontWeight','bold','FontSize',10);

ylabel('Amplitudo','color','w','FontName'
,'Berlin Sans FB
Demi','FontWeight','bold','FontSize',10);
text(0.45,-1,'Phase Response','color','b');
grid on;
box on;
axes(handles.axes1);

plot(w/pi,20*log(abs(h)),'-r');
xlabel('Normalized
Frequency','color','w','FontName','Berlin
Sans FB
Demi','FontWeight','bold','FontSize',10);

ylabel('Amplitudo','color','w','FontName'
,'Berlin Sans FB
Demi','FontWeight','bold','FontSize',10);
text(0.45,-80,'Magnitude
Response','color','r');
grid on;
box on;
axes(handles.axes2);

elseif filter==4
[b,a]=cheby1(n,rs,wn, 'stop');
[h,w]=freqz(b,a);

%Plotting
plot(w/pi,angle(h),'-b');
xlabel('Normalized
Frequency','color','w','FontName','Berlin
Sans FB
Demi','FontWeight','bold','FontSize',10);

ylabel('Amplitudo','color','w','FontName'
,'Berlin Sans FB
Demi','FontWeight','bold','FontSize',10);
text(0.45,-1,'Phase Response','color','b');
grid on;
box on;
axes(handles.axes1);

plot(w/pi,20*log(abs(h)),'-r');
xlabel('Normalized
Frequency','color','w','FontName','Berlin
Sans FB
Demi','FontWeight','bold','FontSize',10);

ylabel('Amplitudo','color','w','FontName'
,'Berlin Sans FB
Demi','FontWeight','bold','FontSize',10);
text(0.45,-80,'Magnitude
Response','color','r');
grid on;
box on;
axes(handles.axes2);

end
```

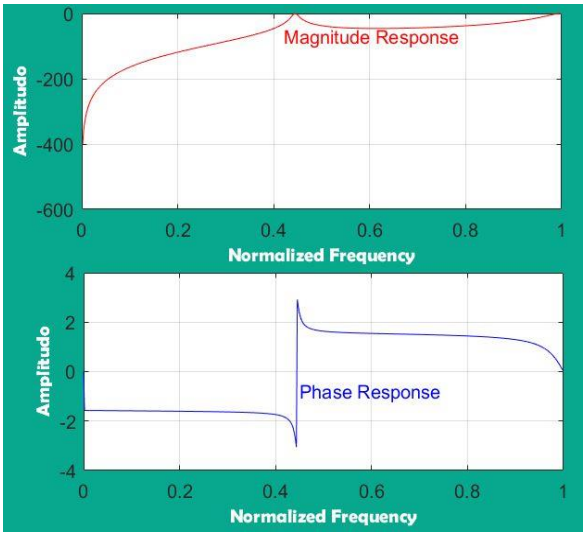
4. HASIL RESPONS CHEBICHEF FILTER MENGGUNAKAN PROGRAM MATLAB

Menggunakan *recursive coefficient 2 pole* dengan parameter sebagai berikut :

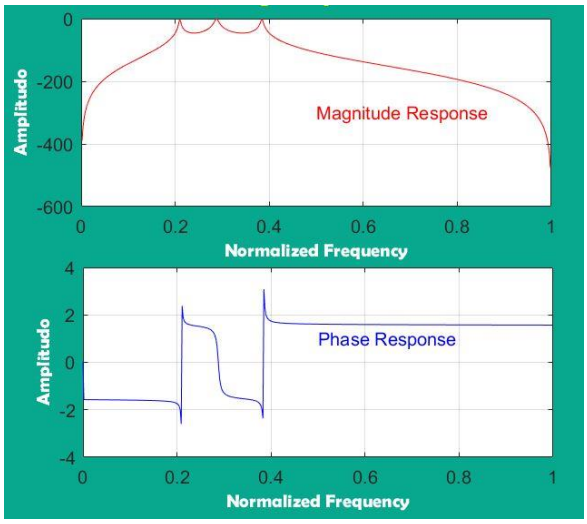
```
fc=400;
fp=200;
f=2000;
rp=0.5;
rs=20;
a0=6.362308E-01;
a1=1.272462E+00;
a2=6.362308E-01;
b1=-1.125379E+00;
b2=-4.195441E-01
```



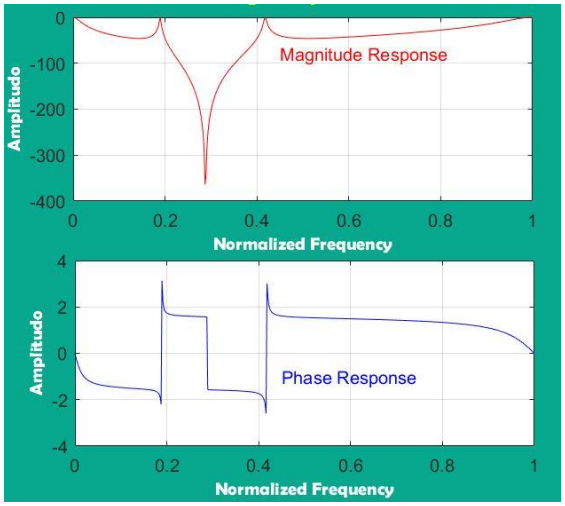
Gambar 1. LPF



Gambar 2. HPF



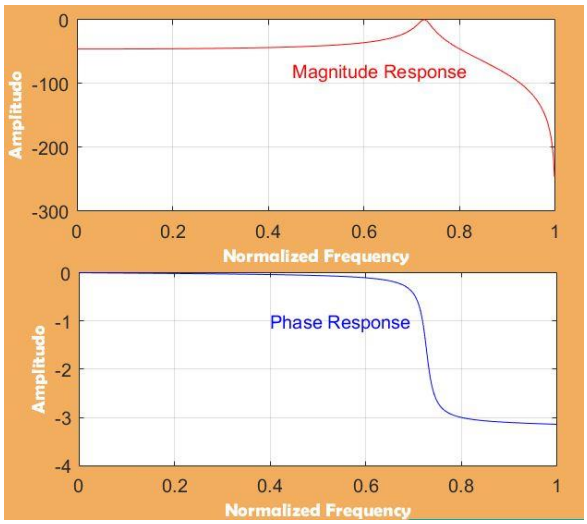
Gambar 3. Band Pass Filter



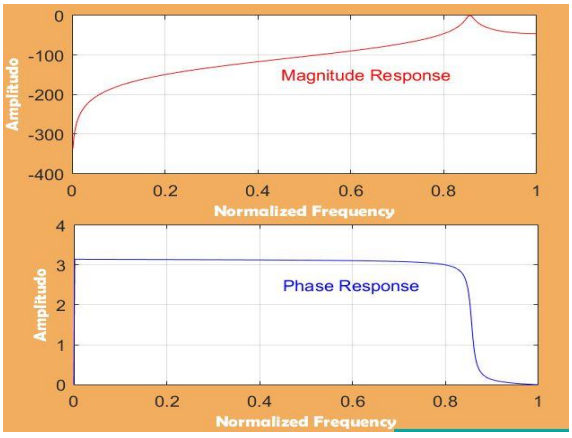
Gambar 4.. BSF

Menggunakan *recursive coefficient 2 pole* dengan parameter sebagai berikut :

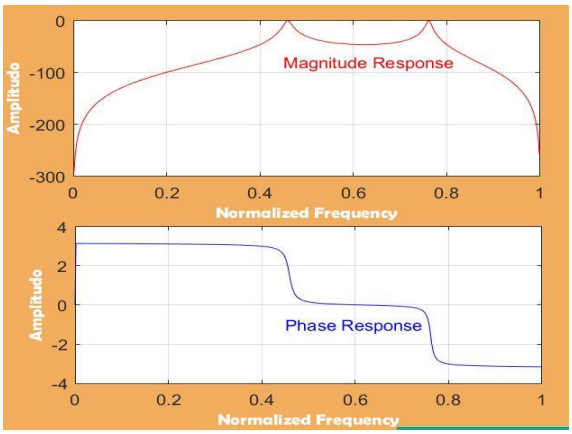
fc=400;
fp=200;
f=2000;
rp=0.5;
rs=20;
a0 = 2.780754E-03
a1 = -1.112302E-02
a2 = 1.668453E-02
a3 = -1.112302E-02
a4 = 2.780754E-03
b1 = -2.764031E+00
b2 = -3.122854E+00
b3 = -1.664554E+00
b4 = -3.502233E-01



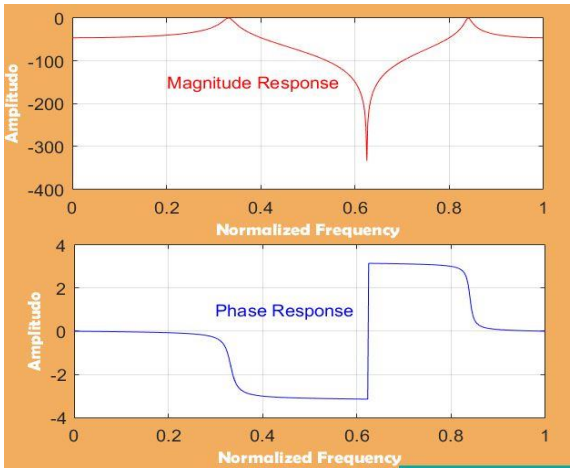
Gambar 5. LPF



Gambar 6. HPF



Gambar 7. BPF



Gambar 8. BSF

5. KESIMPULAN

Respon *Chebyshev* pada *LPF, HPF, BPF dan BSF* didapatkan hasil yang sesuai dengan tanggpan *filter Chebyshev.*

DAFTAR PUSTAKA

1. Bowick, Chris. (1982). *RF Circuit Design*. Indianapolis : Howard W. Sam & Co.,Inc.
2. Sugiharto, Aris. (2006). *Pemrograman Dengan Matlab* . Yogyakarta : Penerbit Andi.
3. Saputro, Wahyu Hadi. (2009). *Analog dan Digital Filter Menggunakan Matlab.*