Bangladesh University of Engineering & Technology

Course No:	EEE 212
Course Title:	Numerical Technique Laboratory
Project Title:	Passive Filter Circuit Analyzer.
Section:	A1
Group No:	3
Submitted By:	2006005- Ashik Abrar Naeem 2006006- Alvee Rahman
Date of Submission:	01-03-2023
Submitted To:	Dr. Hafiz Imtiaz, Associate Professor, EEE, BUET Mohammad Ali, Lecturer (PT), EEE, BUET

Problem Statement: Solve a simple passive filter circuit.

- Input will be the type of filter (low pass, high pass, band pass, band stop) and components used (resistor, capacitor, inductor, and their values).
- Show the schematic of the given circuit.
- Calculate the parameters of the circuit such as cutoff frequencies, center frequency, bandwidth, quality factor.
- Generate gain vs frequency and phase vs frequency curve.
- Find the values of components for a given type of filter having specific cutoff frequencies, center frequency, bandwidth, quality factor.
- Show the output and spectrum of the filter for any input given the user.
- Build a suitable GUI.

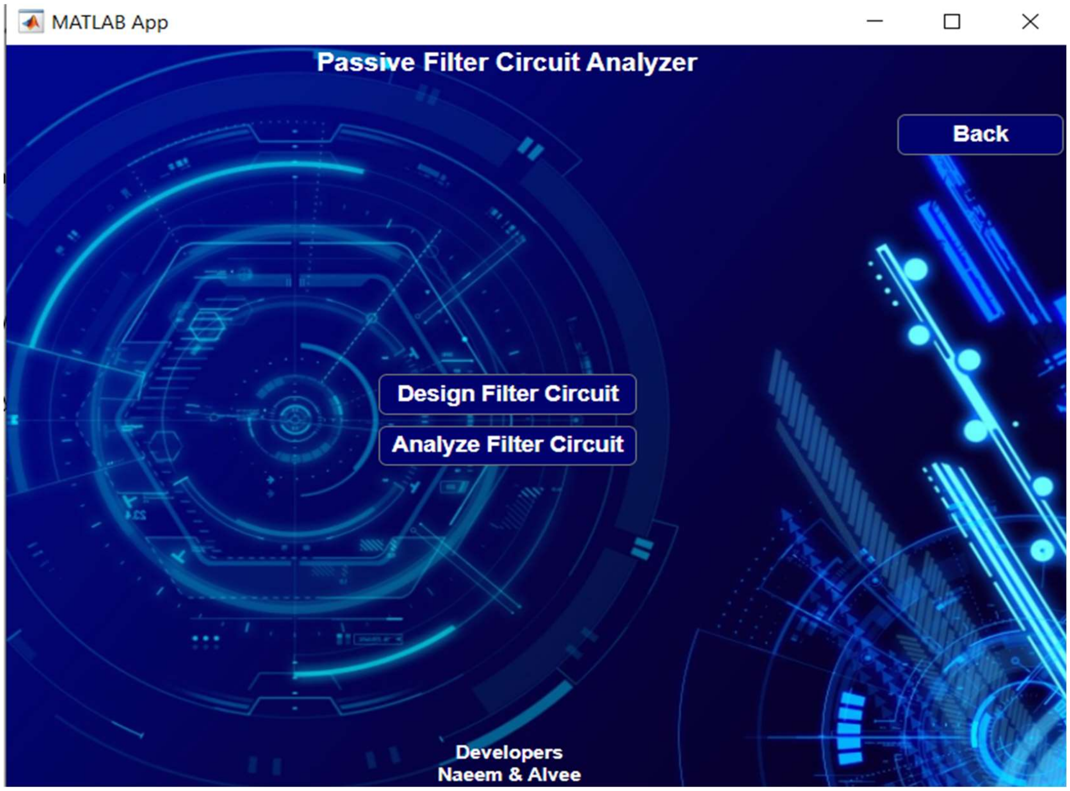
Description:

In this project, we have designed a matlab app that will mainly do two things-

1.Design filter circuit: In this section, the user can design desired filter circuit by giving cutoff frequencies as input. The program will return the values of components (resistance, capacitance, inductance) needed to get the desired output. It will also show from where the output must be taken. Additionally, quality factor, center frequency, and bandwidth will be shown.

2.Analyze filter circuit: In this section, the user can find the parameters (cutoff frequencies, center frequency, bandwidth, quality factor) of a given filter circuit.

In both cases, the gain vs frequency plot and phase vs frequency plot will be shown. Also, the user can see the output signal and spectrum of a given input signal.



After choosing one of the options, the user will have to choose which type of filter they will design/analyze. They will be given following options-

1.Low Pass RC

2.Low Pass RL

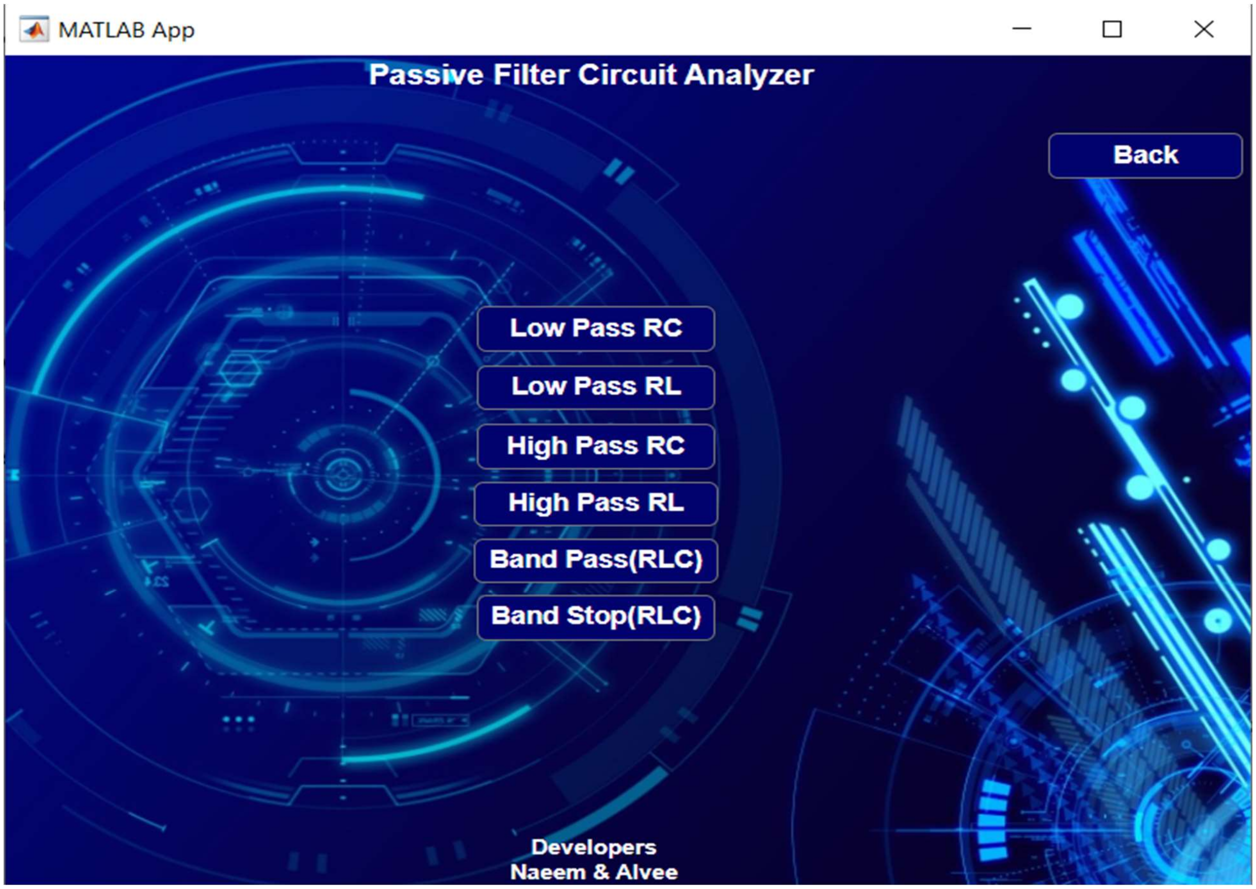
3.High Pass RC

4.High Pass RL

5.Band Pass RLC

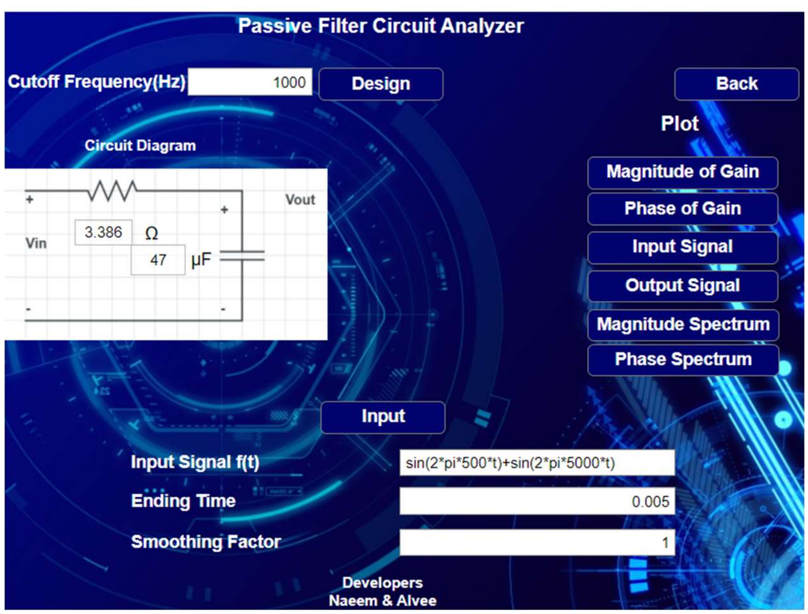
6.Band Stop RLC

After choosing one of the above the user will give input and then output will be shown.



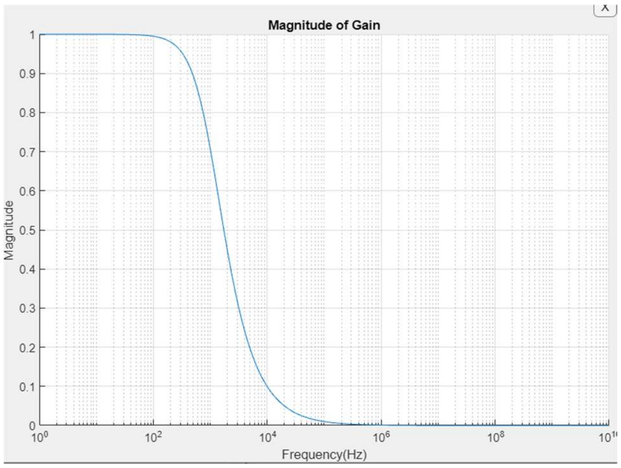
Sample Output:

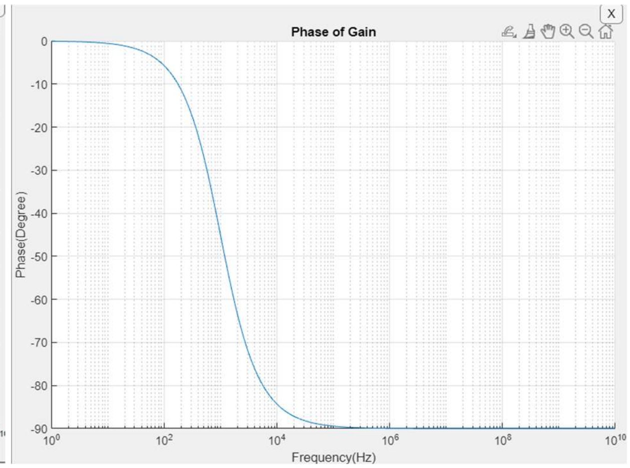
1.Designing Low pass RC filter:

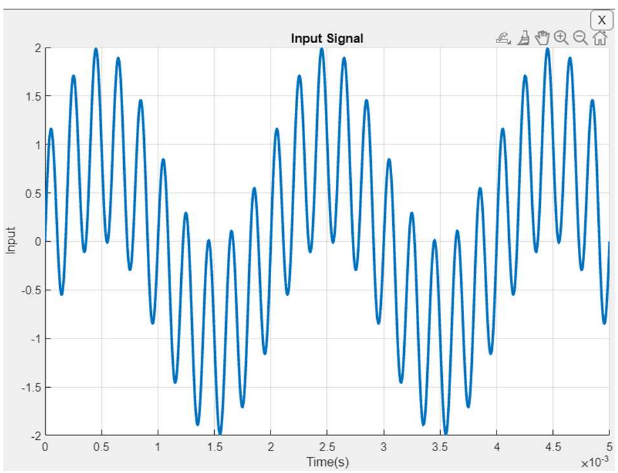


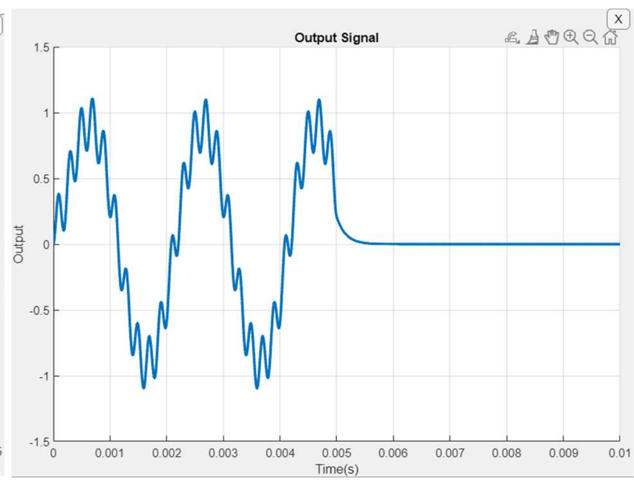
Smoothing Factor:

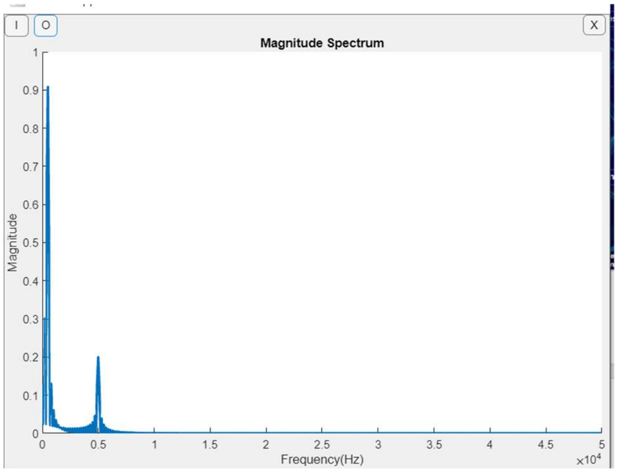
We take sample values between 0 and ending time to plot the curves. Smoothing Factor is the sampling interval divided by 1000. That means if we increase smoothing factor by 1 sampling points will increase by 1000.

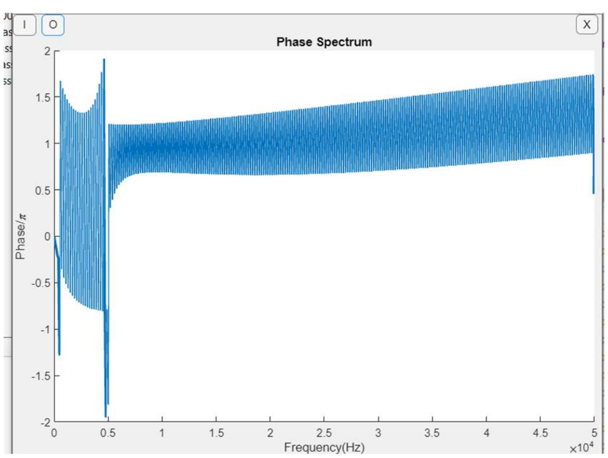






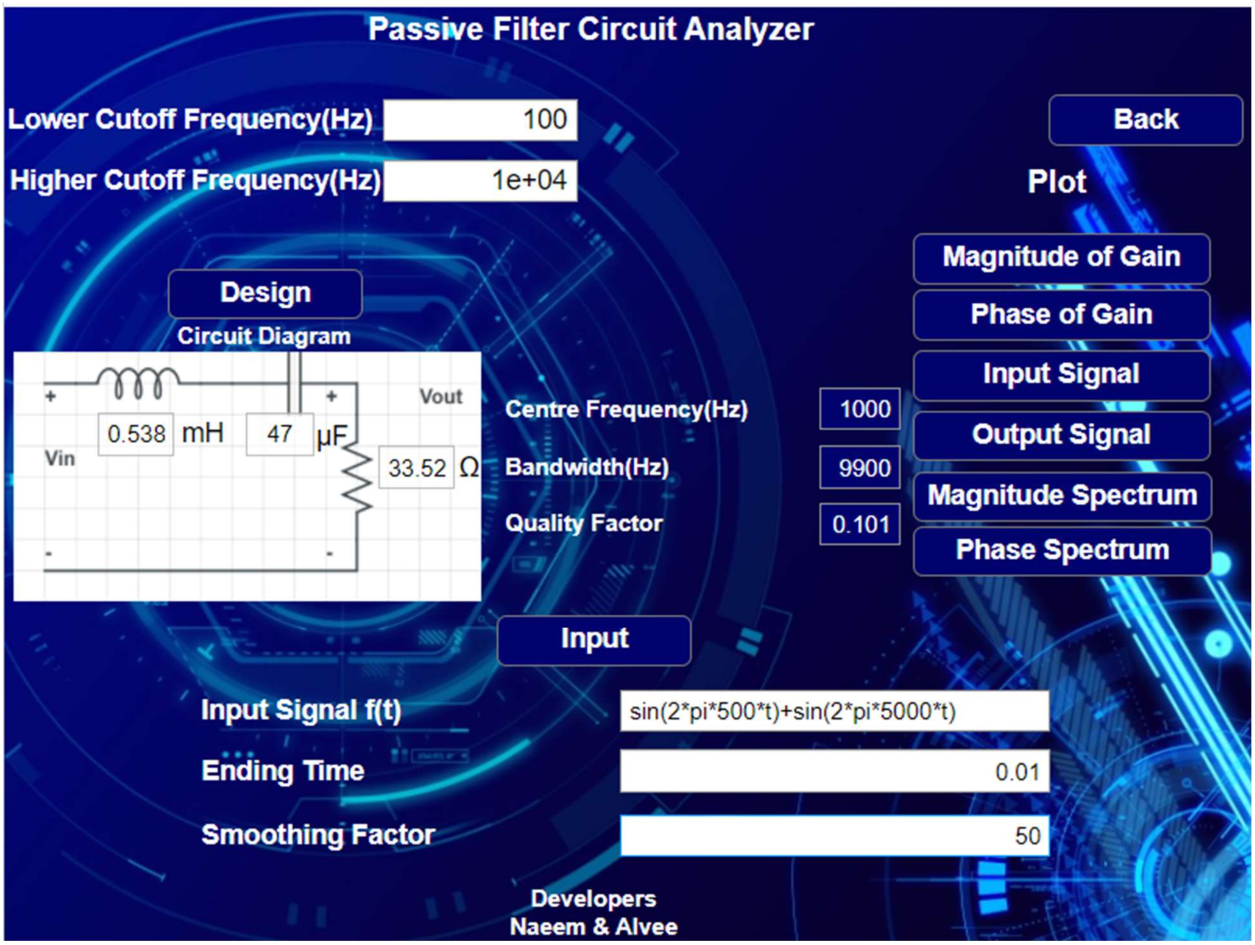


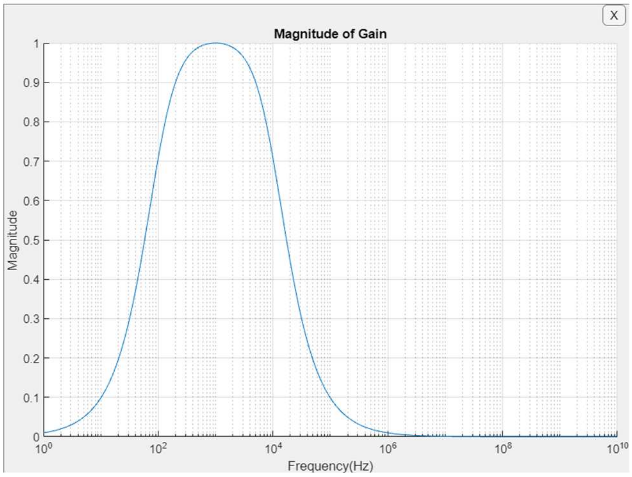


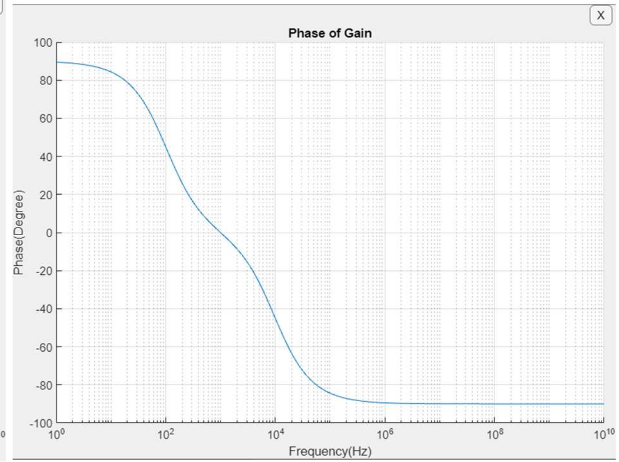


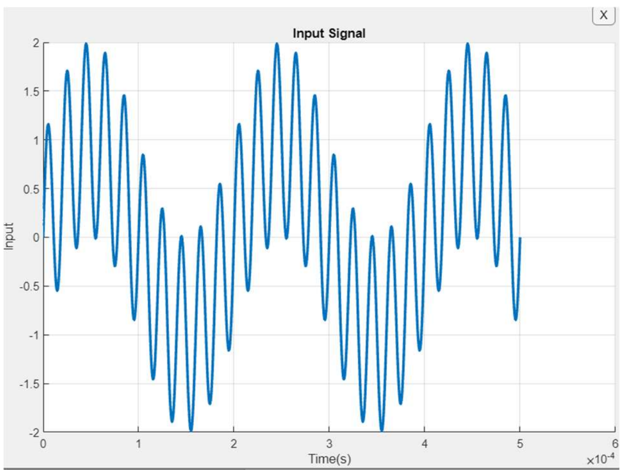
Here the user will give the lower cutoff frequency as input, then the values of R and C will be generated. The value of C was chosen arbitrarily to be 47uF. The user will see the values of components needed to generate this filter circuit. The user will be able to see magnitude of gain vs frequency plot, phase vs frequency plot, input signal given by the user, output signal and spectra of output.

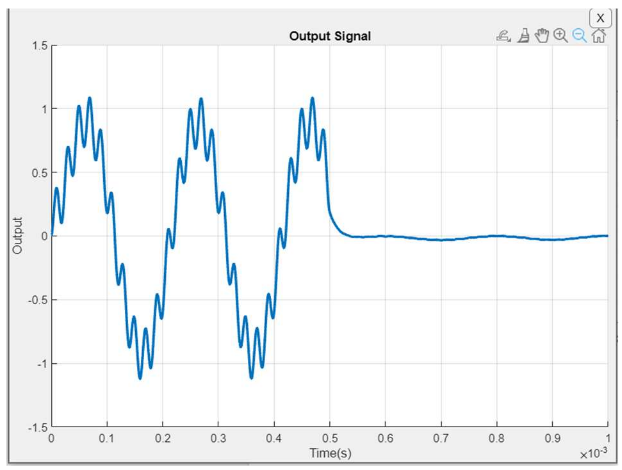
2. **Designing band pass RLC filter:**

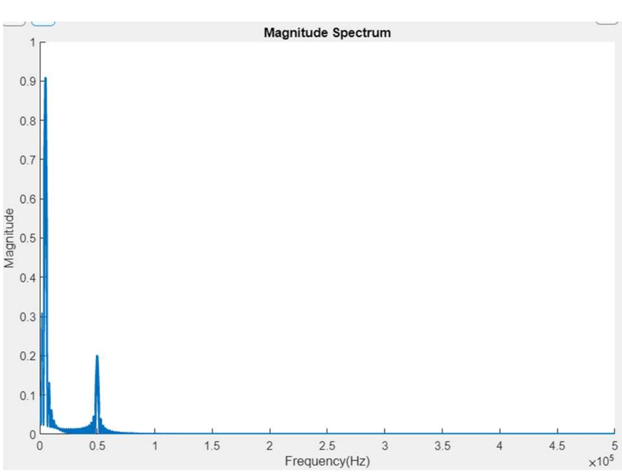


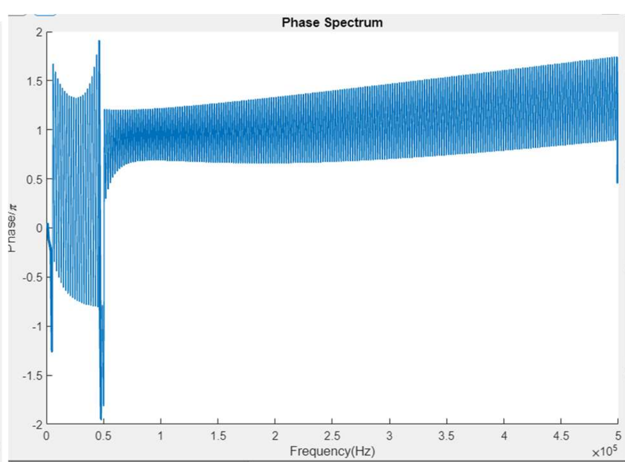






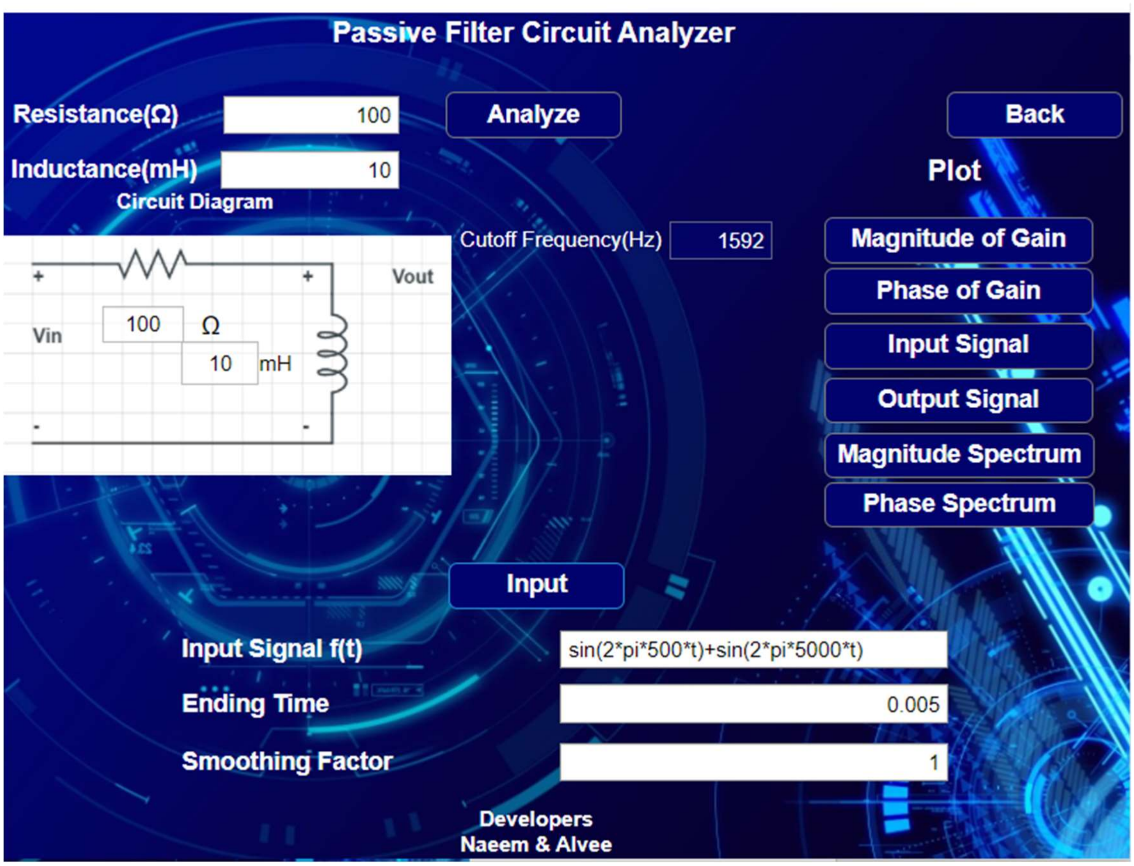


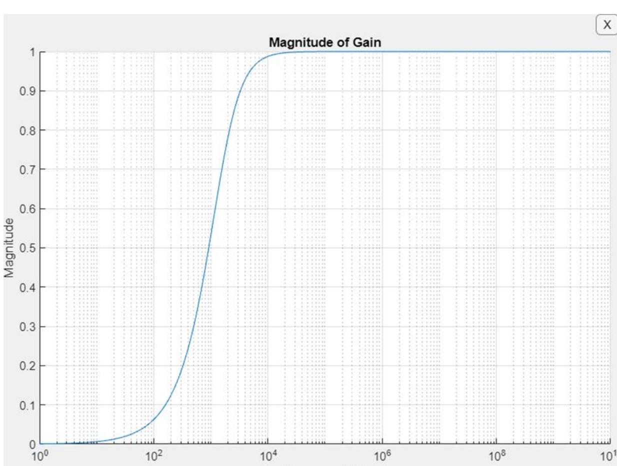


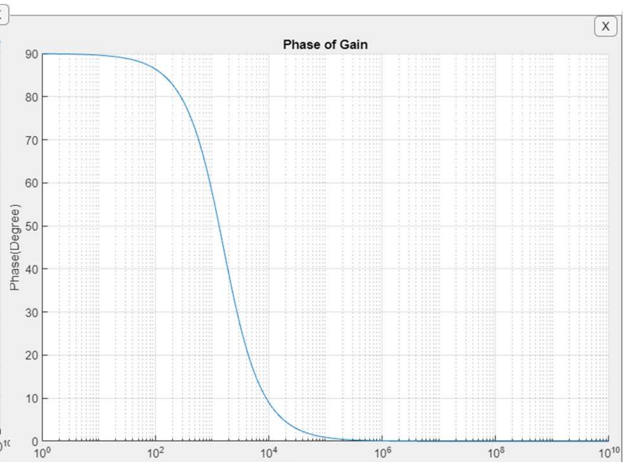


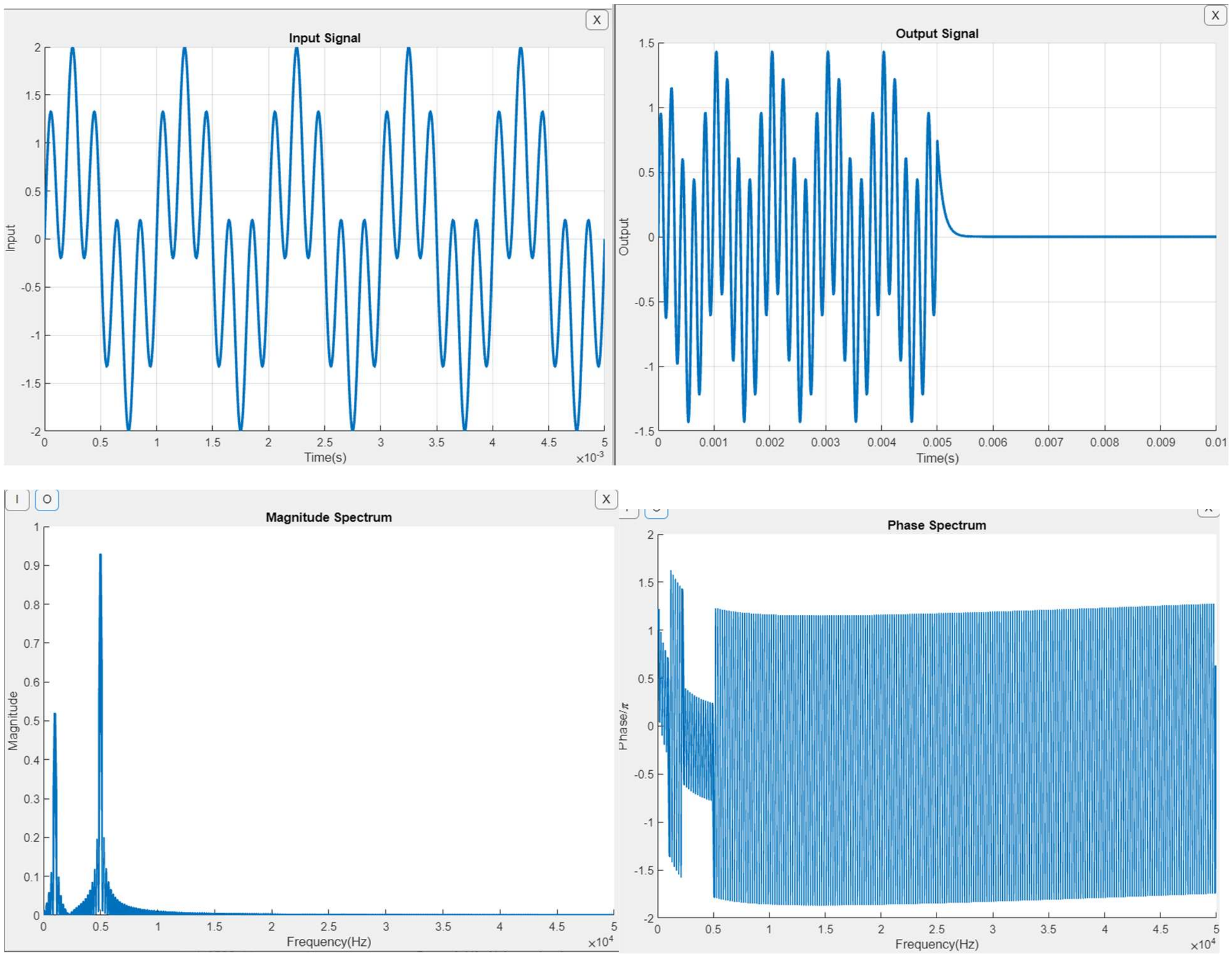
Here the user will give the values of cutoff frequencies as input, then the values of R, L, C, center frequency, bandwidth, and quality factor will be generated. The user will be able to see magnitude of gain vs frequency plot, phase vs frequency plot, input signal given by the user, output signal and spectra of output.

3. **Analyzing High pass RL filter:**



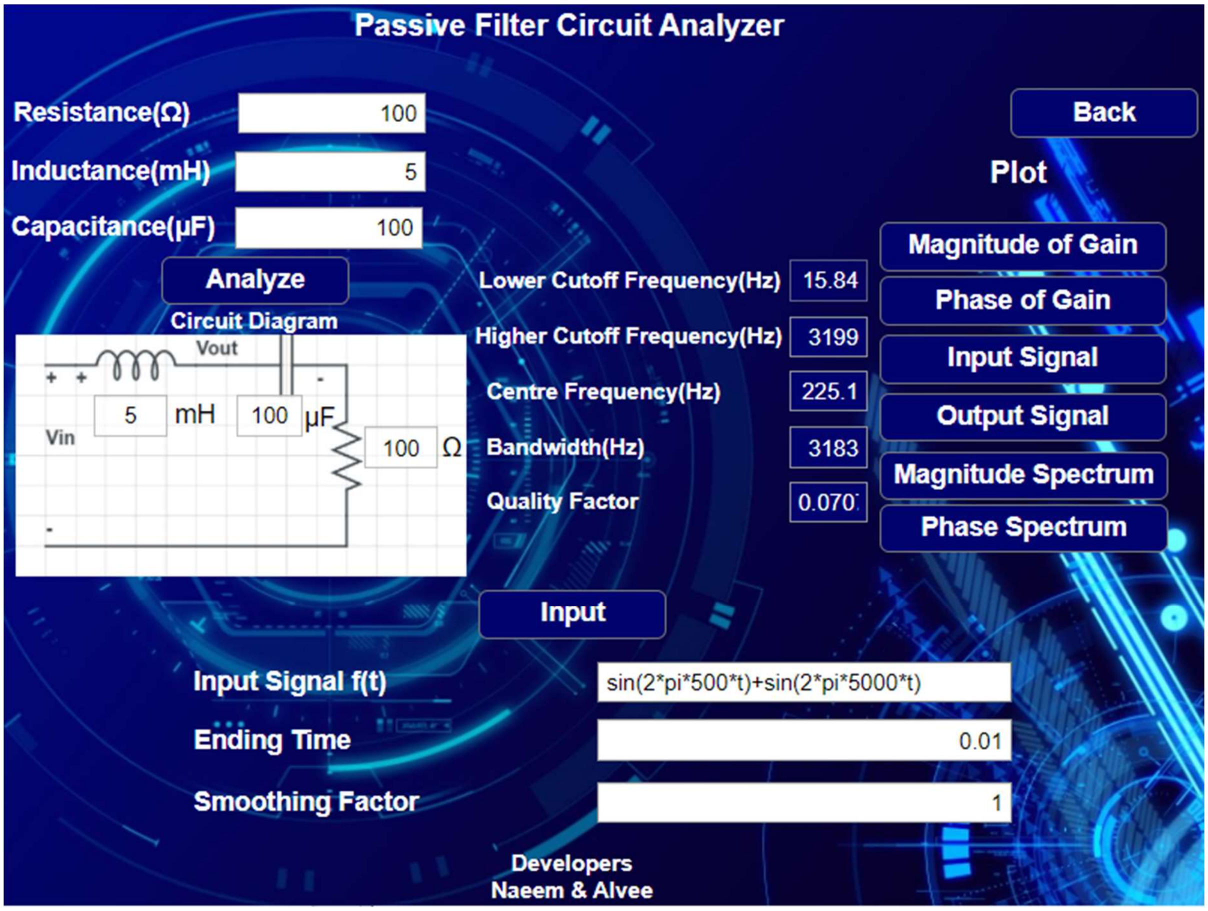


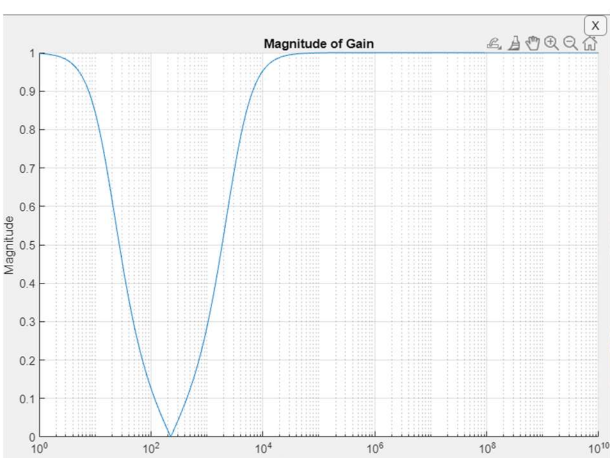


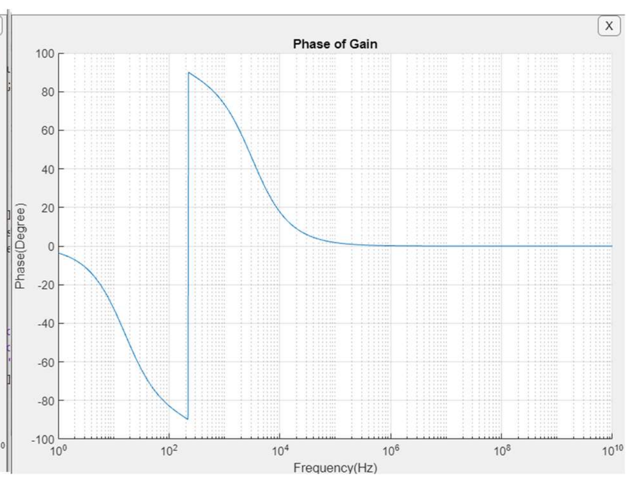


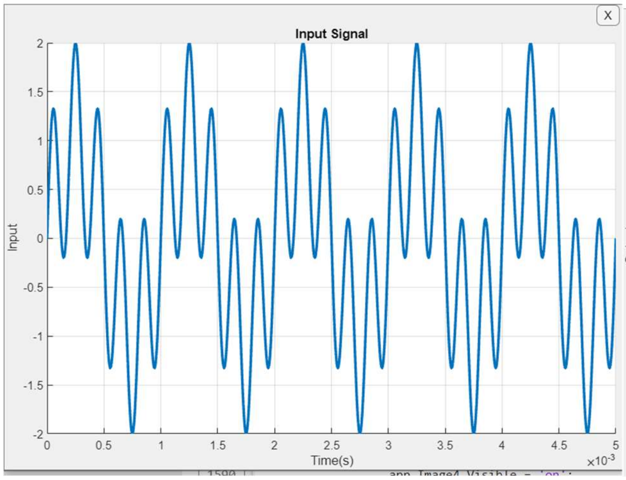
Here the user will give the values of R and L as input, then the value of cutoff frequency will be generated. The user will be able to see magnitude of gain vs frequency plot, phase vs frequency plot, input signal given by the user, output signal and spectra of output.

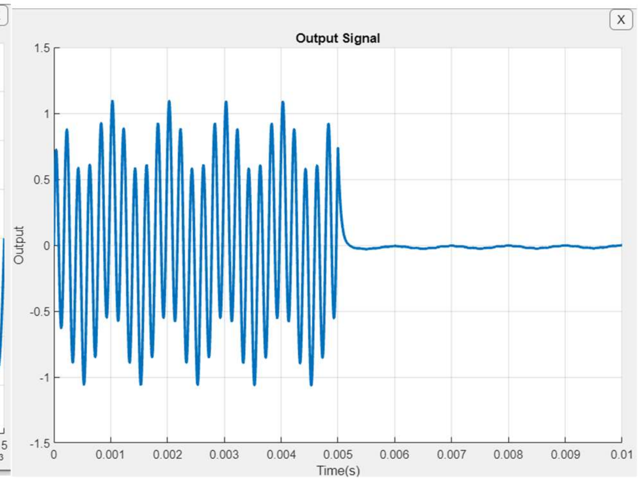
4. **Analyzing Band Stop RLC filter:**

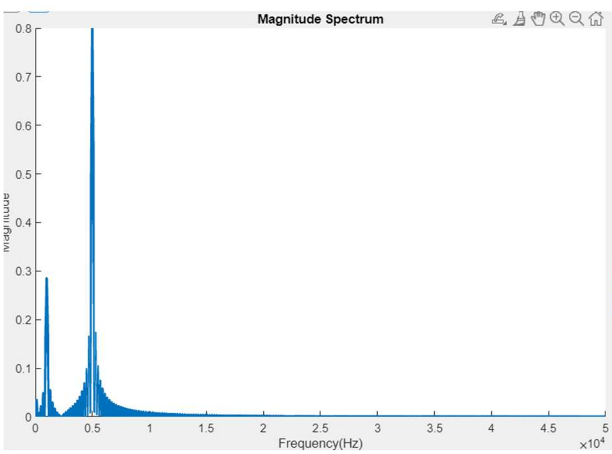


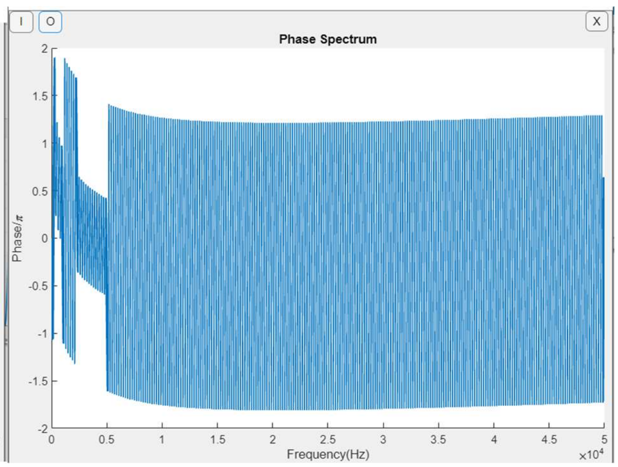












Here the user will give the values of R, L, and C as input, then the values of cutoff frequencies, center frequency, bandwidth, and quality factor will be generated. The user will be able to see magnitude of gain vs frequency plot, phase vs frequency plot, input signal given by the user, output signal and spectra of output.

Process of Solution and related codes:

1.For designing filter circuits the value of capacitor was chosen to be 47uF in RC, and RLC circuits and the value of inductance was chosen to be 25mH in RL circuits. Then the values of other components and parameters were calculated using formulas.

Codes:

Low pass and high pass RC:

```
fc = app.CutoffFrequencyHzEditField_2.Value;
C = 47E-6;
wc = 2*pi*fc;
R = 1/(wc*C);
app.EditField_2.Value = 47;
app.EditField.Value = R;
```

Low pass and high pass RL:

```
fc = app.CutoffFrequencyHzEditField_2.Value;
wc=2*pi*fc;
L=25E-3;
R=wc*L;
app.EditField_2.Value = R;
app.EditField.Value = 25;
```

Band pass and band stop RLC:

```
f1 = app.LowerCutoffFrequencyHzEditField_2.Value;
f2 = app.HigherCutoffFrequencyHzEditField.Value;
C = 47E-6;
w0 = 2*pi*sqrt(f1*f2);
L = 1/(C*w0^2);
B = 2*pi*(f2-f1);
R = B*L;
f0=w0/(2*pi);
Q=w0/(B*2*pi);
app.EditField_4.Value = L*1000;
app.EditField_5.Value = 47;
app.EditField_6.Value = R;

app.CentreFrequencyHzEditField.Value = f0;
app.BandwidthHzEditField.Value = B/(2*pi);
app.QualityFactorEditField.Value = Q;
```

2.For analyzing filter circuits the parameters were calculated from the values of R, L, and C using formulas.

Codes:

Low pass and high pass RC:

```
C = app.CapacitanceFEditField.Value;
R = app.ResistanceEditField.Value;
app.EditField_2.Value = C;
app.EditField.Value = R;
C = C*1E-6;
wc = 1/(R*C);
fc = wc/(2*pi);
```

Low pass and high pass RL:

```
L = app.InductancemHEditField.Value;
R = app.ResistanceEditField.Value;
app.EditField_2.Value = R;
app.EditField.Value = L;
app.mHLabel.Visible = 'on';
app.Label_2.Visible = 'on';
L = L*1E-3;
wc=R/L;
fc = wc/(2*pi);
```

Band pass and band stop RLC:

```
R = app.ResistanceEditField.Value;
L = app.InductancemHEditField.Value;
C = app.CapacitanceFEditField_2.Value;
app.EditField_4.Value = L;
app.EditField_5.Value = C;
app.EditField_6.Value = R;
app.mHLabel_3.Visible = 'on';
app.FLabel_3.Visible = 'on';
app.Label_4.Visible = 'on';
L = L*1E-3;
C = C*1E-6;
w0=1/sqrt(L*C);
f0=w0/(2*pi);
w1=-R/(2*L)+sqrt((R/(2*L))^2+1/(L*C));
w2=R/(2*L)+sqrt((R/(2*L))^2+1/(L*C));
f1=w1/(2*pi);
f2=w2/(2*pi);
B=f2-f1;
Q=w0/(B*2*pi);
```

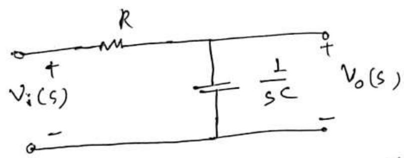
3.To generate magnitude of gain vs frequency and phase of gain vs frequency, the gain function was first calculated. Then for a wide range of frequency the magnitude of gain and phase of gain was evaluated and finally the plots were generated.

Codes:

```
f=logspace(0,10,1000);

w = 2*pi.*f;

gain=1./sqrt(1+(R*C*w).^2); %LOW PASS RC
phase=-atand(w*R*C);

gain=1./sqrt(1+(L*w/R).^2); %LOW PASS RL
phase=-atand(w*L/R);

gain=1./sqrt(1+(1./(R*C*w)).^2); %HIGH PASS RC
phase=atand(1./(w*R*C));
gain=1./sqrt(1+(R./(w*L)).^2); %HIGH PASS RL
phase=atand(R./(w*L));

gain=1./sqrt(1+((w*L-1./(w*C))/R).^2); %BAND PASS RLC
phase=-atand((w*L-1./(w*C))/R);

gain=1./sqrt(1+(R./(w*L-1./(w*C))).^2); %BAND STOP RLC
phase=atand(R./(w*L-1./(w*C)));
```

4. To generate the output signal of a given input signal, the user will have to enter ending time of the signal, starting time was chosen to be 0. To generate the output signal, the equivalent s-domain circuit was generated. From the s-domain equivalent circuit transfer function H(s)=Vo(s)/Vi(s) was calculated for each case. Now Vo(s) will be Vi(s) multiplied by H(s). If we change output signal to time domain then the output will be convolution of input signal and transfer function in time domain. To determine the transfer function or impluse response of the circuit inverse Laplace transform was used. The transfer function for each circuit was calculated in the following manner-

1. Low pass RC,



Transfer function, $H(s) = \dfrac{V_o(s)}{V_i(s)}$

$$= \frac{\frac{1}{sC}}{R + \frac{1}{sC}}$$

$$= \frac{1}{1 + sRC}$$

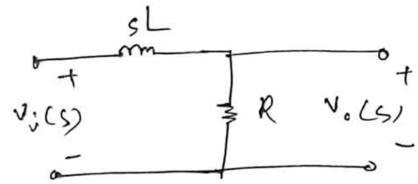
$$= \frac{1}{RC} \cdot \frac{1}{s + \frac{1}{RC}}$$

$\therefore \; V_o(s) = V_i(s) \times H(s)$

~~Using~~ $\therefore \; V_o(t) = V_i(t) * H(t)$

Using inverse Laplace, $H(t) = \dfrac{1}{RC} e^{-\frac{t}{RC}} u(t)$

2. Low pass RL



Transfer function, $H(s) = \dfrac{V_o(s)}{V_i(s)}$

$$= \frac{R}{R + sL}$$

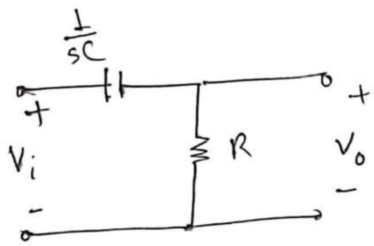
$$= \frac{R}{L} \times \frac{1}{s + \frac{R}{L}}$$

Now, $V_o(s) = H(s) \times V_i(s)$

$\therefore \; V_o(t) = H(t) * V_i(t)$

Using inverse Laplace, $H(t) = \dfrac{R}{L} e^{-\frac{R}{L}t} u(t)$

3. High Pass RC



Transfer function, $H(s) = \dfrac{V_o(s)}{V_i(s)}$

$$= \frac{R}{R + \frac{1}{sC}}$$

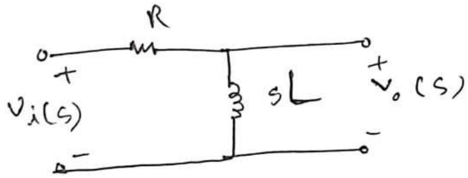
$$= 1 - \frac{1}{RC} \times \frac{1}{s + \frac{1}{RC}}$$

Now, $V_o(s) = V_i(s) \times H(s)$

$\therefore V_o(t) = V_i(t) * H(t)$

Using inverse Laplace, $H(t) = \delta(t) - \frac{1}{RC} e^{-\frac{1}{RC}t} u(t)$

4. High pass RL



Transfer function, $H(s) = \dfrac{V_o(s)}{V_i(s)}$

$$= \frac{sL}{sL + R}$$

$$= 1 - \frac{R}{L} \times \frac{1}{s + \frac{R}{L}}$$

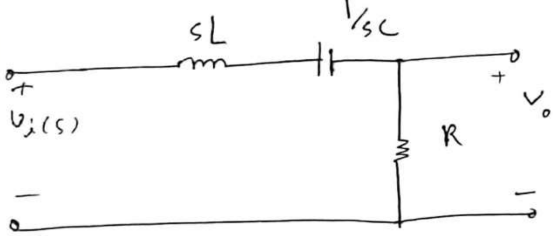
Now, $V_o(s) = V_i(s) \times H(s)$

$\therefore V_o(t) = V_i(t) * H(t)$

Using inverse Laplace transform,

$$H(t) = \delta(t) - \frac{R}{L} e^{-\frac{R}{L}t} u(t)$$

5. Band pass (RLC) :



Transfer function, $H(s) = \dfrac{R}{R + sL + \frac{1}{sC}}$

$$= \frac{RsC}{s^2LC + RsC + 1}$$

$$= \frac{R}{L} \cdot \frac{s}{s^2 + \frac{R}{L}s + \frac{1}{LC}}$$

$$= 2a \cdot \frac{s}{(s - P_1)(s - P_2)}$$

[Let $\frac{R}{2L} = a$ and $\frac{1}{LC} = b$, then $P_1 = -a+\sqrt{a^2-b^2}$ and $P_2 = -a-\sqrt{a^2-b^2}$]

Now, if $a>b$,

$$H(s) = \frac{2aP_1}{(P_1-P_2)(s-P_1)} + \frac{2aP_2}{(P_2-P_1)(s-P_2)}$$

$$= \frac{2a}{(P_1-P_2)} \times \left\{\frac{P_1}{s-P_1} - \frac{P_2}{s-P_2}\right\}$$

$$\therefore H(t) = \frac{2a}{P_1-P_2}\left\{P_1 e^{P_1 t} - P_2 e^{P_2 t}\right\} u(t)$$

if $\cancel{H(s)}$ $\cancel{a=b}$ $a<b$,

$$H(s) = 2a\frac{s+a}{(s+a)^2+q^2} - \frac{2a^2}{q} \times \frac{q}{(s+a)^2+q^2}$$

$$H(t) = 2ae^{-at}\cos qt - \frac{2a^2}{q}e^{-at}\sin qt \; u(t)$$

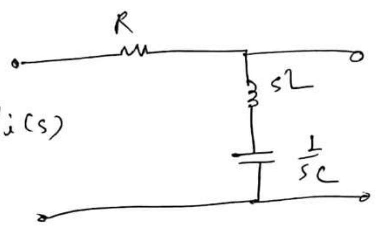
if $a = b$

$$H(t) = 2a\,(1+at)\,e^{at}\,u(t)$$

Now, $V_o(s) = V_i(s) \times H(s)$

$\therefore V_o(t) = V_i(t) * H(t)$

6. Band ~~Pas~~ stop (RLC)



For band stop filter $H(t)$ will $\delta(t) - H'(t)$; where $H'(t)$ is the transfer function obtained for band pass RLC filters.

$\therefore$ And $V_o(t) = V_i(t) * H(t)$

Finally, convolution integral was evaluated and output was generated. We used 3 user defined function.

Codes:

```
function input = Input(app,t)
    xt = str2func(['@(t)',app.inputexpression]);
    input = xt(t);
end

function  output = Convolution(app,step_response,input)
    if app.convolution2 == 1
        output = conv(step_response*(app.te/(1000*app.k)),input);
        input(2*length(input)-1) = 0;
        app.convolution2 = 0;
        output = input - output;
    else
        output = conv(step_response*(app.te/(1000*app.k)),input);
    end
end
```


```
function  output = Convolution(app,step_response,input)
    if app.convolution2 == 1
        output = conv(step_response*(app.te/(1000*app.k)),input);
        input(2*length(input)-1) = 0;
        app.convolution2 = 0;
        output = input - output;
    else
        output = conv(step_response*(app.te/(1000*app.k)),input);
    end
end
```


```
function  step_response = Step_Response(app,t)
    if app.page == 111 %(Low Pass RC)
        C = 47E-6;
        R = app.EditField.Value;
        ht = @(t) (1/(R*C))*exp(-t/(R*C));
        step_response = ht(t);
    elseif app.page == 121
        C = app.CapacitanceFEditField.Value;
        R = app.ResistanceEditField.Value;
        C = C*1E-6;
        ht = @(t) (1/(R*C))*exp(-t/(R*C));
        step_response = ht(t);
    elseif app.page == 112 %(Low Pass RL)
        L = 25E-3;
        R = app.EditField_2.Value;
        ht = @(t) (R/L)*exp(-t*R/L);
        step_response = ht(t);
```

```
elseif app.page == 122
    L = app.InductancemHEditField.Value;
    R = app.ResistanceEditField.Value;
    L = L*1E-3;
    ht = @(t) (R/L)*exp(-t*R/L);
    step_response = ht(t);
elseif app.page == 113 %(High Pass RC)
    C = 47E-6;
    R = app.EditField_3.Value;
    ht = @(t) (1/(R*C))*exp(-t/(R*C));
    app.convolution2 = 1;
    step_response = ht(t);
elseif app.page == 123
    C = app.CapacitanceFEditField.Value;
    R = app.ResistanceEditField.Value;
    C = C*1E-6;
    ht = @(t) (1/(R*C))*exp(-t/(R*C));
    app.convolution2 = 1;
    step_response = ht(t);
elseif app.page == 114 %(High Pass RL)
    L = 25E-3;
    R = app.EditField.Value;
    ht = @(t) (R/L)*exp(-t*R/L);
    app.convolution2 = 1;
    step_response = ht(t);
elseif app.page == 124
    L = app.InductancemHEditField.Value;
    R = app.ResistanceEditField.Value;
    L = L*1E-3;
    ht = @(t) (R/L)*exp(-t*R/L);
    app.convolution2 = 1;
    step_response = ht(t);
elseif app.page == 115 %(BandPass RLC)
    L = app.EditField_4.Value;
    L = L*1E-3;
    C = 47*1E-6;
    R = app.EditField_6.Value;
    a = R/(2*L);
    b = 1/(sqrt(L*C));
    if a > b
        p1 = -a+sqrt(a^2-b^2);
        p2 = -a-sqrt(a^2-b^2);
        ht = @(t) (2*a/(p1-p2))*(p1*exp(p1*t)-p2*exp(p2*t));
    elseif a < b
        q = sqrt(b^2-a^2);
        ht = @(t) (2*a*cos(q*t)-((2*a^2)/q)*sin(q*t)).*exp(-a*t);
    else
        ht = @(t) 2*a(1+a*t).*exp(a*t);
    end
    step_response = ht(t);
elseif app.page == 125
    R = app.ResistanceEditField.Value;
    L = app.InductancemHEditField.Value;
    C = app.CapacitanceFEditField_2.Value;
    L = L*1E-3;
```

```
            C = C*1E-6;
            a = R/(2*L);
            b = 1/(sqrt(L*C));
            if a > b
                p1 = -a+sqrt(a^2-b^2);
                p2 = -a-sqrt(a^2-b^2);
                ht = @(t) (2*a/(p1-p2))*(p1*exp(p1*t)-p2*exp(p2*t));
            elseif a < b
                q = sqrt(b^2-a^2);
                ht = @(t) (2*a*cos(q*t)-((2*a^2)/q)*sin(q*t)).*exp(-a*t);
            else
                ht = @(t) 2*a(1+a*t).*exp(a*t);
            end
            step_response = ht(t);
        elseif app.page == 116 (BandStop RLC)
            L = app.EditField_4.Value;
            L = L*1E-3;
            C = 47*1E-6;
            R = app.EditField_6.Value;
            a = R/(2*L);
            b = 1/(sqrt(L*C));
            if a > b
                p1 = -a+sqrt(a^2-b^2);
                p2 = -a-sqrt(a^2-b^2);
                ht = @(t) (2*a/(p1-p2))*(p1*exp(p1*t)-p2*exp(p2*t));
            elseif a < b
                q = sqrt(b^2-a^2);
                ht = @(t) (2*a*cos(q*t)-((2*a^2)/q)*sin(q*t)).*exp(-a*t);
            else
                ht = @(t) 2*a(1+a*t).*exp(a*t);
            end
            app.convolution2 = 1;
            step_response = ht(t);
        elseif app.page == 126
            R = app.ResistanceEditField.Value;
            L = app.InductancemHEditField.Value;
            C = app.CapacitanceFEditField_2.Value;
            L = L*1E-3;
            C = C*1E-6;
            a = R/(2*L);
            b = 1/(sqrt(L*C));
            if a > b
                p1 = -a+sqrt(a^2-b^2);
                p2 = -a-sqrt(a^2-b^2);
                ht = @(t) (2*a/(p1-p2))*(p1*exp(p1*t)-p2*exp(p2*t));
            elseif a < b
                q = sqrt(b^2-a^2);
                ht = @(t) (2*a*cos(q*t)-((2*a^2)/q)*sin(q*t)).*exp(-a*t);
            else
                ht = @(t) 2*a(1+a*t).*exp(a*t);
            end
            app.convolution2 = 1;
            step_response = ht(t);
        end
    end
```

```
end
```

5. To obtain frequency spetrum of the output we used Discrete Fourier Transform. The codes are given below

Magnitude Spectrum:

```
T = app.te/(1000*app.k);
Fs = 1/T;
L = app.te/T + 1;
t = (0:L-1)*T;
S = app.Input(t);
Y = fft(S);
P2 = abs(Y/L);
P1 = P2(1:L/2+1);
P1(2:end-1) = 2*P1(2:end-1);
f = Fs*(0:(L/2))/L;
plot(app.UIAxes,f,P1,'LineWidth',2);
```

Frequency Spectrum

```
T = app.te/(1000*app.k);
Fs = 1/T;
L = app.te/T + 1;
t = (0:L-1)*T;
S = app.Input(t);
Y = fft(S);
P2 = angle(Y/L);
P1 = P2(1:L/2+1);
P1(2:end-1) = 2*P1(2:end-1);
f = Fs*(0:(L/2))/L;
plot(app.UIAxes,f,P1/pi,'LineWidth',2);
```

Conclusion:

Filter circuits can extract important frequencies from signals that also contain undesirable or irrelevant frequencies. Hence, designing filter circuit is very important in signal processing. To get desired filter circuit we need to choose the values of component correctly. This app will do the calculation and also show the output signal for a given input signal. So, this app can be very helpful for signal processing.