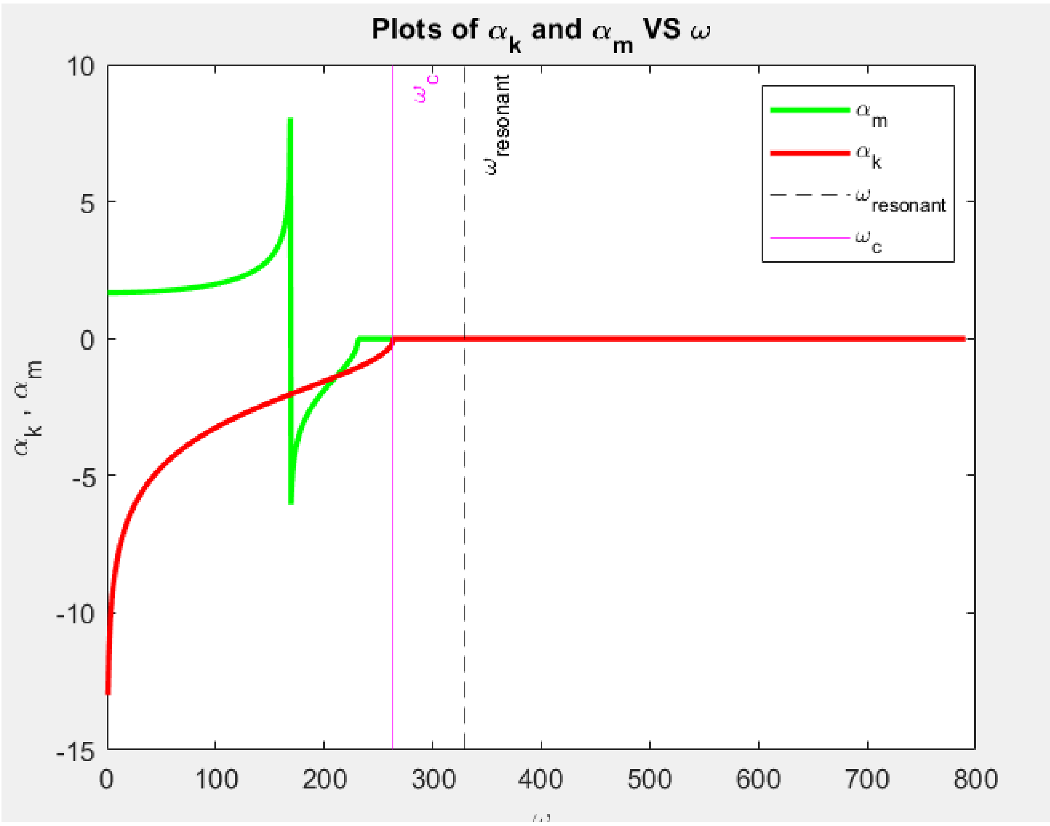
```
clc;
close all;
clear all;
%----------For High pass constant-k Filter--------------%

% shows error in publishing when we have to take inputs.
% Uncomment the below C ,L and f lines for Input from user.
C = input('Enter the Value of Capacitance in micro Farad:- ');
L = input('Enter the Value of Inductance in milli Henry:- ');
%f = input('Enter the value of Frequency in Giga Hertz: -');
% Here is f is not need , we need the frequency for when we want to find
% Image Impedance

%C = 20; L = 5;
C = C * (10^(-6));
L = L * (10^(-3));
%f = f * (10)^9;
%omega = 2*pi*f;
%-------------Cutoff Frequency-------------%
omega_c = 1/(4*L*C)^(1/2);
%disp('Value of omega_c');
%disp(omega_c);
%disp('range of w');
w = linspace(0,3*omega_c,1000);
%--------------Propogation Factor-------------%
% e_gamma = 1 + z1/(2*z2) + (z1/z2    + z1^2/(2*z2)^2)^(1/2) on solving in
```

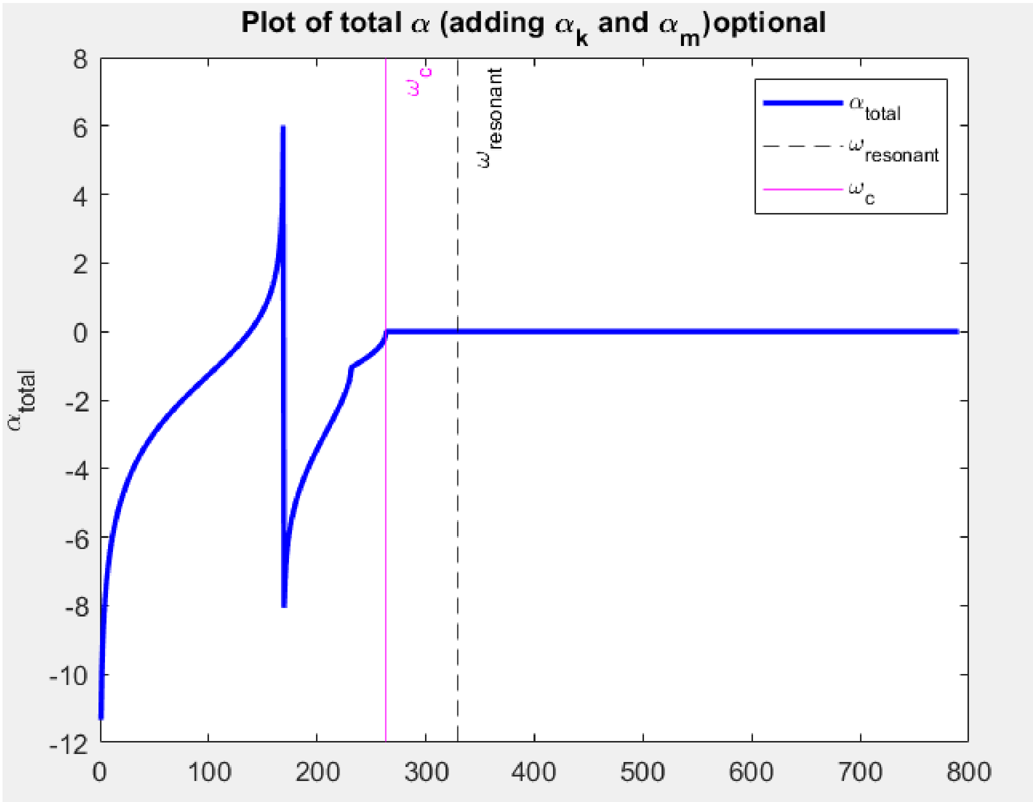
```
% terms of omega and omega_c we get:-
e_gamma_k = zeros(1,length(w));
for k = 1 : length(w)
    e_gamma_k(k) = 1 - 2*((omega_c^2)/w(k)^2)
+((2*omega_c/w(k))*(((omega_c^2/w(k)^2) - 1)^(1/2)));
end
propogation_constant_k = log(e_gamma_k);
alpha_k = real(propogation_constant_k);
%disp('alpha_k has been calculated');



%------------High pass m-derived filter-----------------%
%--------- m = 0.6 for optimum result-------------%

% shows error in publishing when we have to take inputs.
% Uncomment the below m lines for Input from user.
m = input('Enter the value of m:- ');
%m = 0.6;

omega_resonant = omega_c/(1 - (m^2))^(1/2);
Z = zeros(1,length(w));
e_gamma_m = zeros(1,length(w));
for i =1 : length(w)
    Z(i)   = ((2*m*omega_c/w(i))^2)/((((1 - m^2)*(omega_c/w(i)))^2) - 1); % Z =
z1/z2 in terms of omega_c and omega
    %---------------Propogation factor---------------%
    e_gamma_m(i) = 1 + Z(i)/2 + (Z(i)*(1 + Z(i)/4))^(1/2);
```

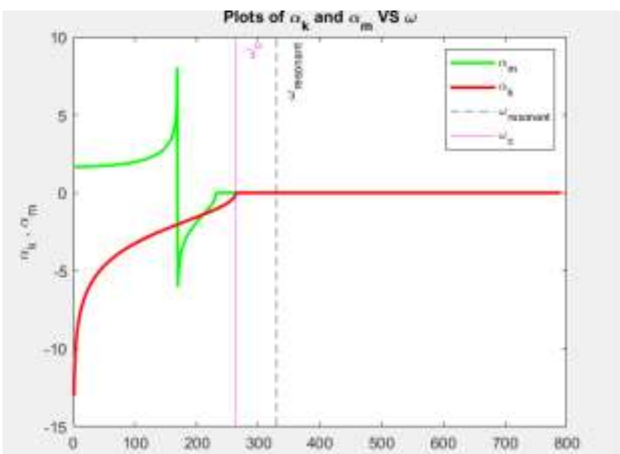
```
end
%--------------Propogation Constant---------%
propogation_constant_m = log(e_gamma_m);
alpha_m = real(propogation_constant_m);
figure(1);
p = plot(w,alpha_m,'-g',w,alpha_k,'r');
p(1).LineWidth = 2;
p(2).LineWidth = 2;
xline(omega_resonant,'--k','\omega_{resonant}')
xline(omega_c,'magenta','\omega_c');
legend('\alpha_m','\alpha_k','\omega_{resonant}','\omega_c');
title('Plots of \alpha_k and \alpha_m VS \omega');
xlabel('\omega');
ylabel('\alpha_k , \alpha_m');

% The below was not asked to calculate. It's only to know the total value
% of alpha since in real time applications we use an m-derived filter
% together with constant-k filter

figure(2);
total_alpha = alpha_k + alpha_m;
q = plot(w,total_alpha,'b');
q(1).LineWidth = 2;
xline(omega_resonant,'--k','\omega_{resonant}')
xline(omega_c,'magenta','\omega_c');
legend('\alpha_{total}','\omega_{resonant}','\omega_c');
```

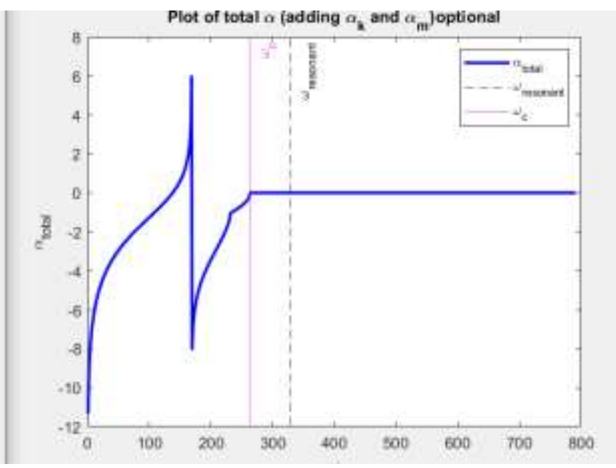
```
title('Plot of total \alpha (adding \alpha_k and \alpha_m){optional}');
xlabel('\omega');
ylabel('\alpha_{total}');
```



Both the above and below plots are for the corresponding values of L = 40mH , C = 90µF and m = 0.6


Plot of total α (adding α_k and α_m)optional
α_{total}
$\omega_{resonant}$
ω_c
0
100
200
300
400
500
600
700
800
8
6
4
2
0
-2
-4
-6
-8
-10
-12


Plots of α_k and α_m VS ω
α_m
α_k
$\omega_{resonant}$
ω_c
α_k , α_m


Plot of total α (adding α_k and α_m)optional
α_{total}
$\omega_{resonant}$
ω_c