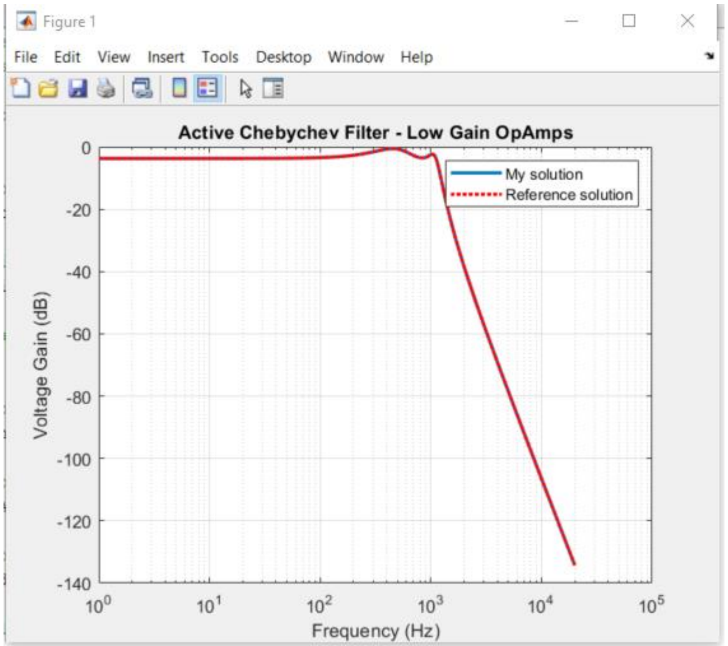
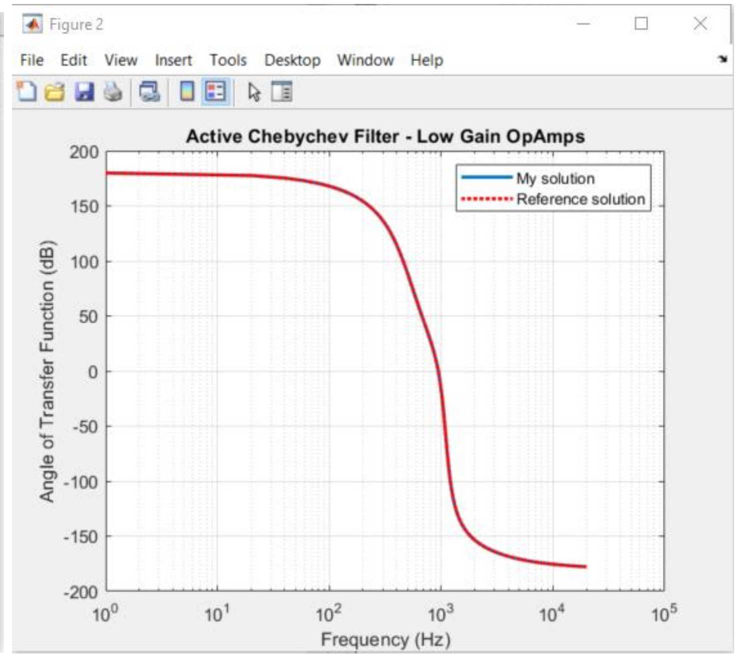
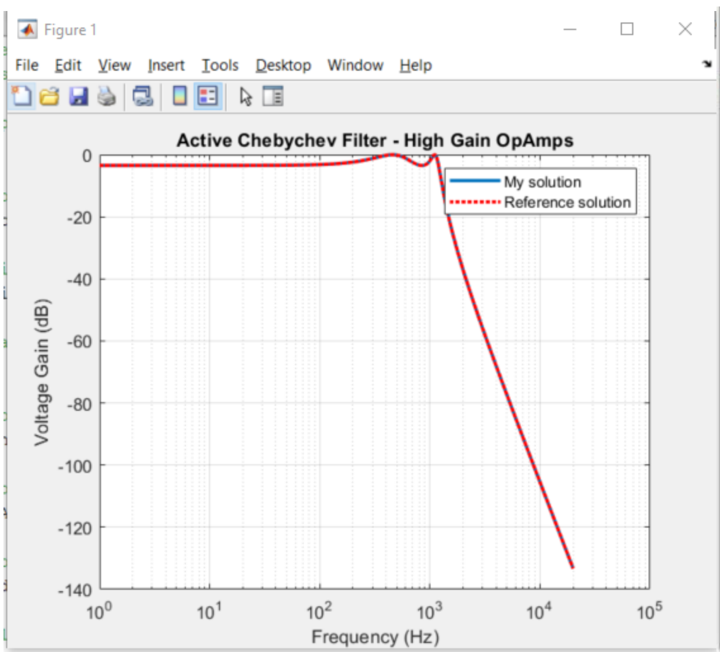
Bryan Jay
260738764

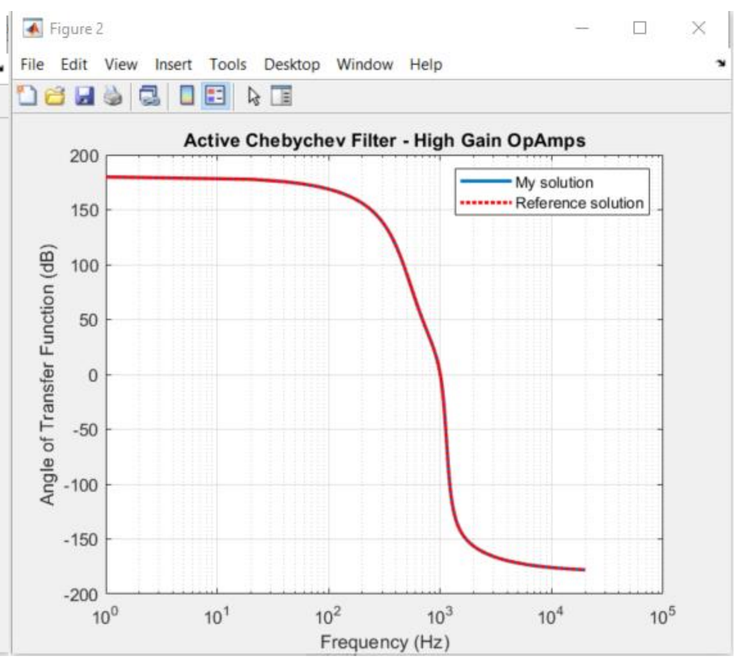
Testbench_chebychev_filter.m





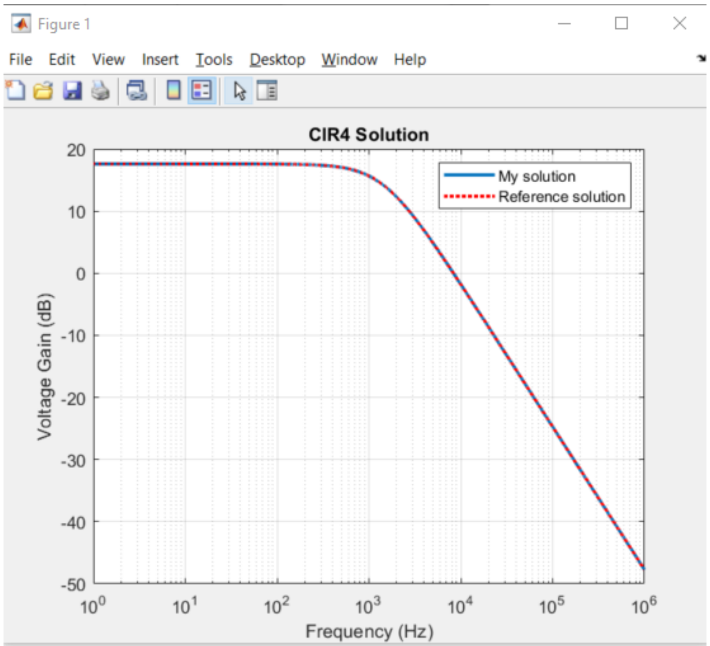
Testbench_chebychev_filter_largeGain.m

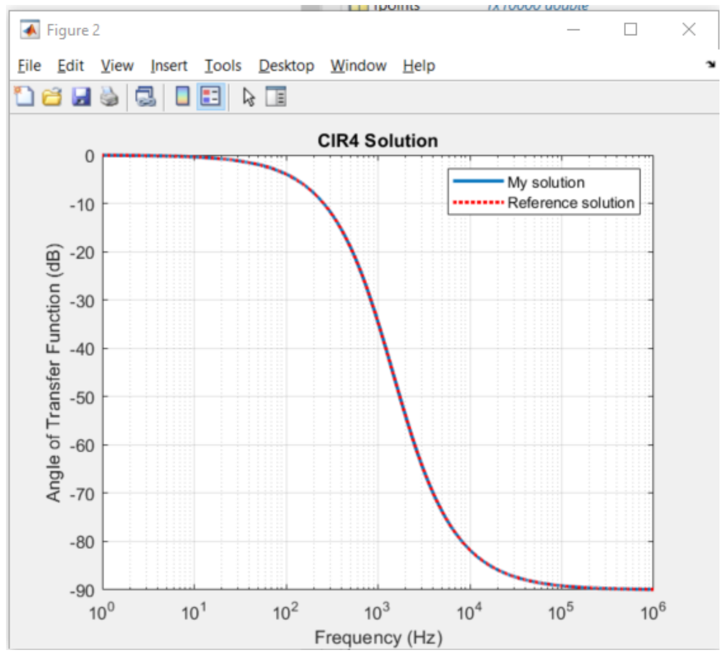




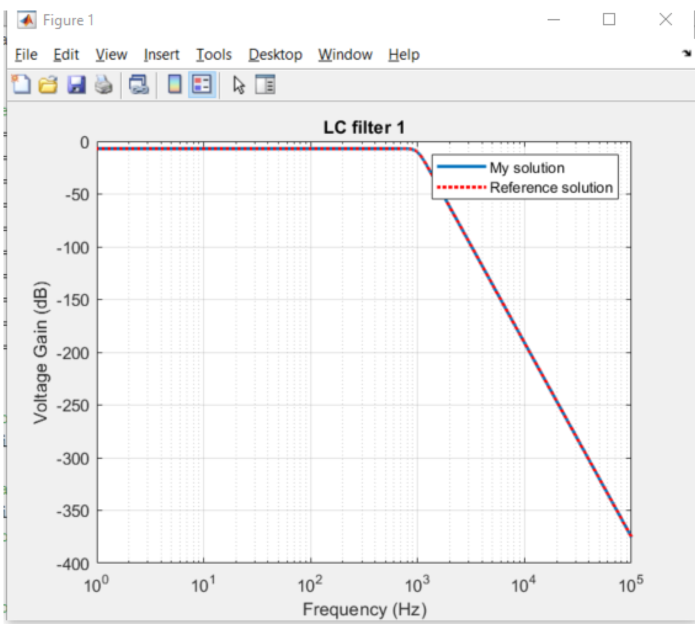
TestBench_CIR4.m

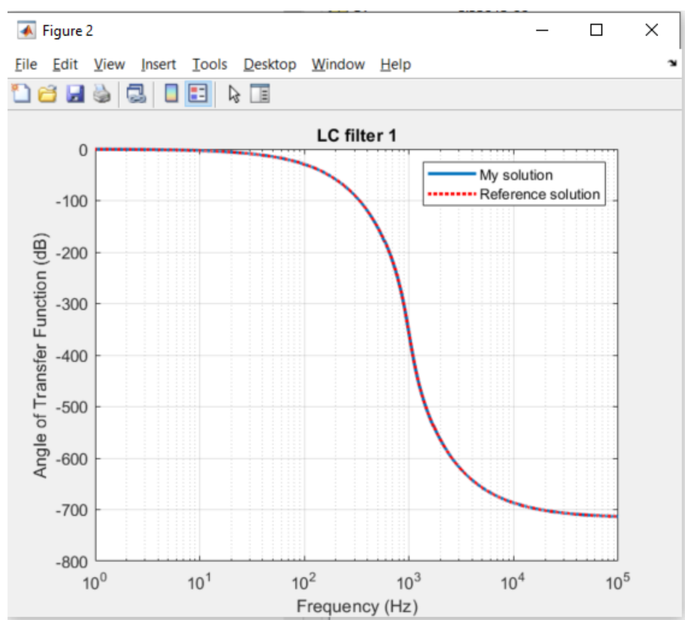
Bryan Jay
260738764





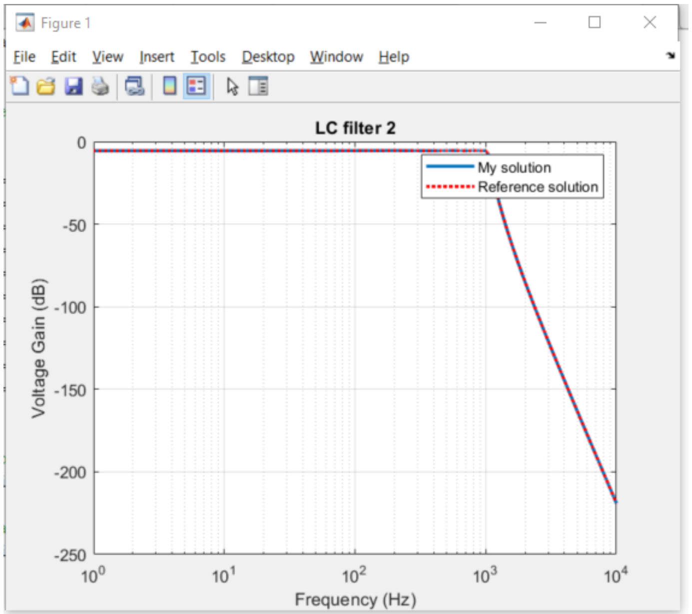
TestBench_LCfilter1.m

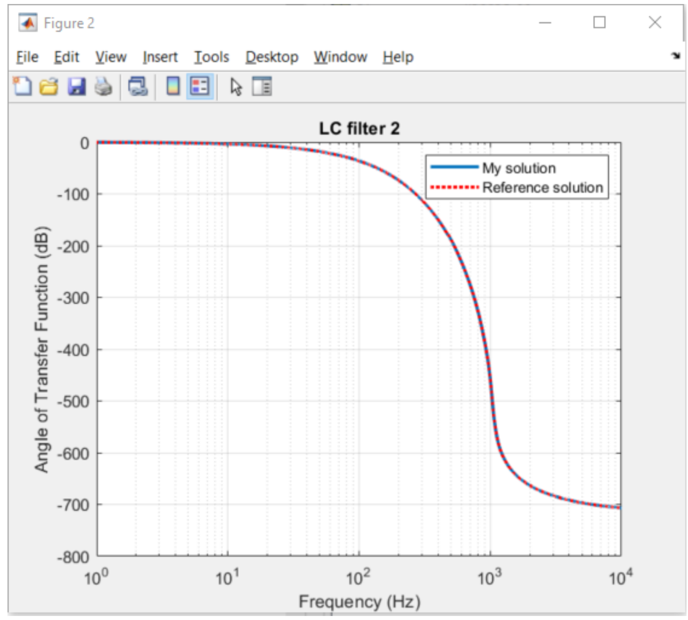




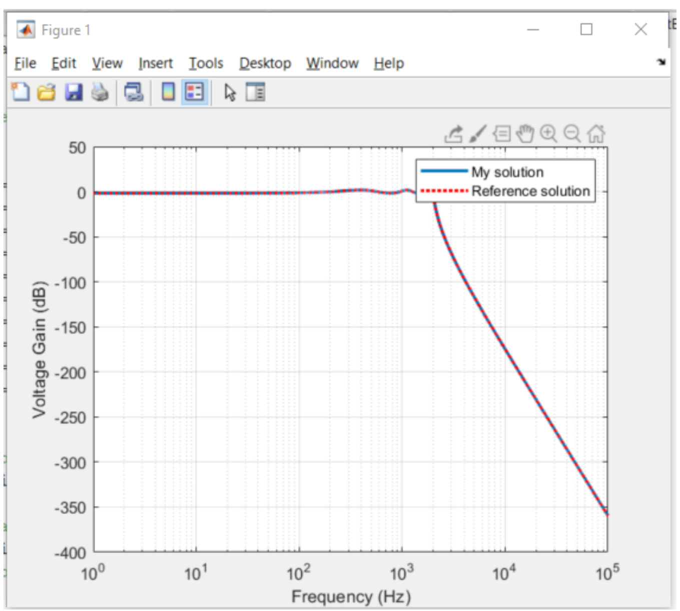
TestBench_LCfilter2.m

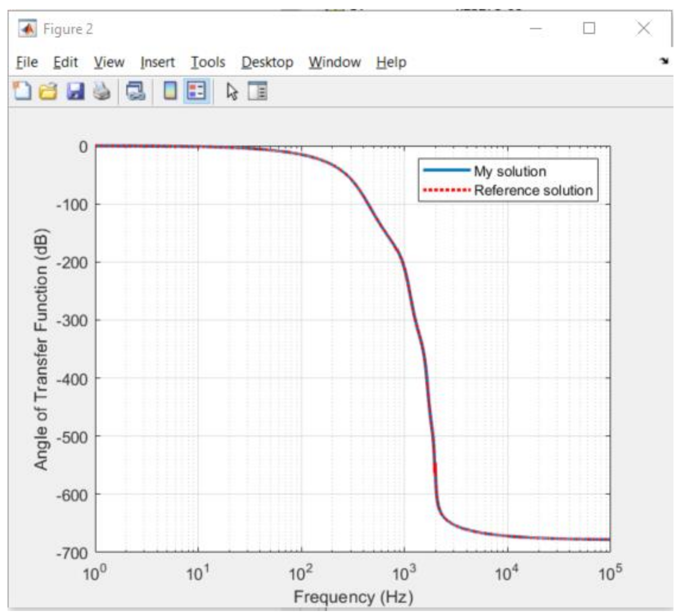
Bryan Jay
260738764





TestBench_LCfilter3.m





Bryan Jay
260738764

MATLAB Code

ind.m

```
function ind(n1,n2,val)
         % ind(n1,n2,val)
         % Add stamp for inductor to the global circuit representation
         % Inductor connected between n1 and n2
         % The indjuctance is val in Henry
         % global G
         % global C
         % global b
         % Date:

    % defind global variables
    global G
    global b
    global C

    len = length(G);
    sz = len + 1;

    b(sz) = 0;
    G(sz,sz) = 0;

    C(sz,sz) = -val;

    if n2 ~= 0
       G(n2,sz) = -1;
       G(sz,n2) = -1;
    end

    if n1 ~= 0
       G(n1,sz) = 1;
       G(sz,n1) = 1;
    end
```

Bryan Jay
260738764

vccs.m

```
function vccs(nd1,nd2,ni1,ni2,val)
% vccs(nd1,nd2,ni1,ni2,val)
% Add stamp for voltage controlled current source
% to the global circuit representation
% ni1 and ni2 are the controlling voltage nodes
% the controlled current source is between nd1 and nd2
% The controlled current (from nd1 to nd2) is val*(Vni1-Vni2)

global G

if nd1 ~= 0
    if ni1 ~= 0
       G(nd1,ni1) = G(nd1,ni1) + val;
    end

    if ni2 ~= 0
       G(nd1,ni2) = G(nd1,ni2) - val;
    end
end

if nd2 ~= 0
    if ni1 ~= 0
       G(nd2,ni1) = G(nd2,ni1) - val;
    end

    if ni2 ~= 0
       G(nd2,ni2) = G(nd2,ni2) + val;
    end
end
```

Bryan Jay
260738764

vcvs.m

```
function vcvs(nd1,nd2,ni1,ni2,val)
% vcvs(nd1,nd2,ni1,ni2,val)
% Add stamp for a voltage controlled voltage source
% to the global circuit representation
% val is the gain of the vcvs
% ni1 and ni2 are the controlling voltage nodes
% nd1 and nd2 are the controlled voltage nodes
% The relation of the nodal voltages at nd1, nd2, ni1, ni2 is:
% Vnd1 - Vnd2 = val*(Vni1 - Vni2)

global G
global b
global C

d = length(G);        %current size of the MNA
sz = d+1;             %new row

% increase the size of the matrix
G(sz,sz) = 0;
C(sz,sz) = 0;
b(sz) = 0;


if nd1~=0
    G(sz,nd1) = G(sz,nd1) + 1;
    G(nd1,sz) = G(nd1,sz) + 1;
end

if nd2~=0
    G(sz,nd2) = G(sz,nd2) - 1;
    G(nd2,sz) = G(nd2,sz) - 1;
end

if ni1~=0
    G(sz,ni1) = G(sz,ni1) - val;
end

if ni2~=0
    G(sz,ni2) = G(sz,ni2) + val;
end
```

Bryan Jay
260738764

fsolve.m

```
function r = fsolve(fpoints ,out)
%  fsolve(fpoints ,out)
%  Obtain frequency domain response
% global variables G C b
% Inputs: fpoints is a vector containing the fequency points at which
%         to compute the response in Hz
%         out is the output node
% Outputs: r is a vector containing the value of
%            of the response at the points fpoint


% define global variables
global G C b

shape = length(fpoints);
r = zeros(shape,1);

for i = 1:shape
    x = inv(G + C*1i*2*pi*fpoints(i))*b;
    r(i) = x(out);
end
```