ECSE 597: Circuit Simulations and Modeling

Assignment 3, October 3, 2019
Wenjie Wei, 260685967

1 Chebychev Filter

Matlab code listings will be presented at the end of the document. Figures 1 and 2 below show the magnitude and angular response of the Chebychev filter.

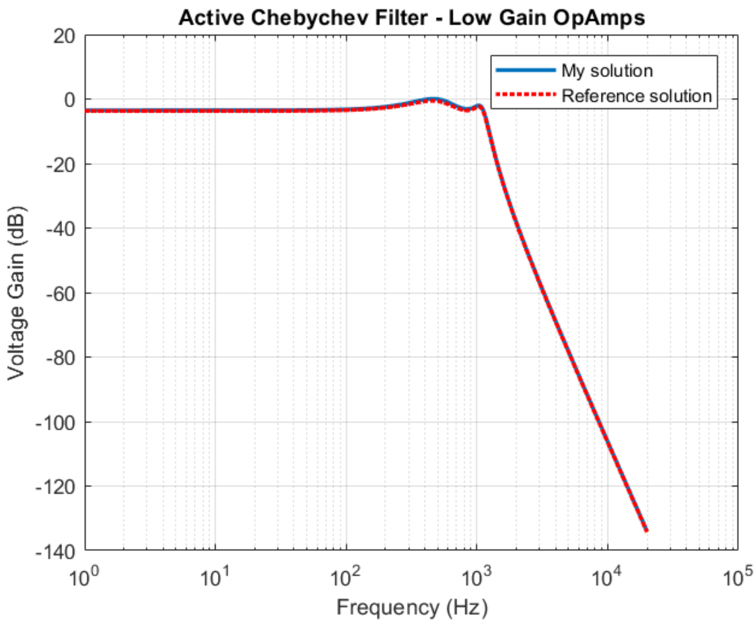

Figure 1: Magnitude Response of the Chebychev Filter

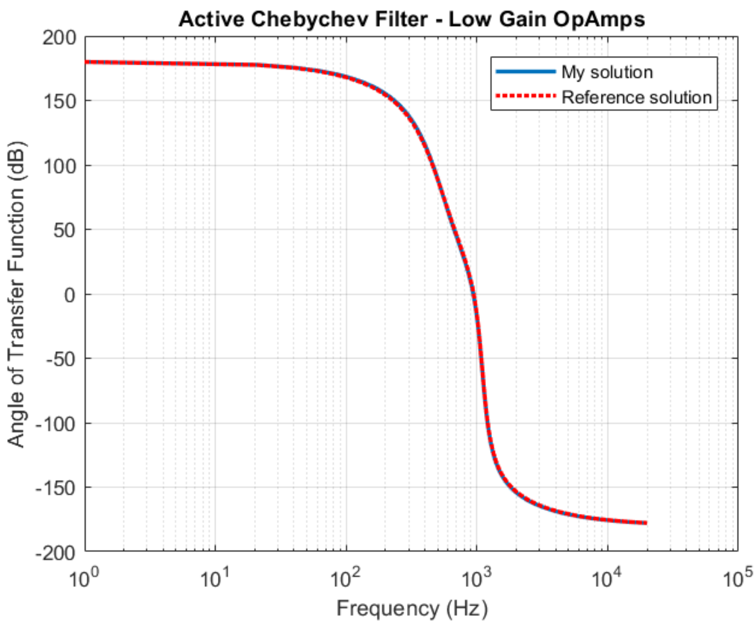

Figure 2: Angular Response of the Chebychev Filter

2 Large Gain Chebychev Filter

Figures 3 and 4 below show the magnitude and angular response of the large gain Chebychev filter.

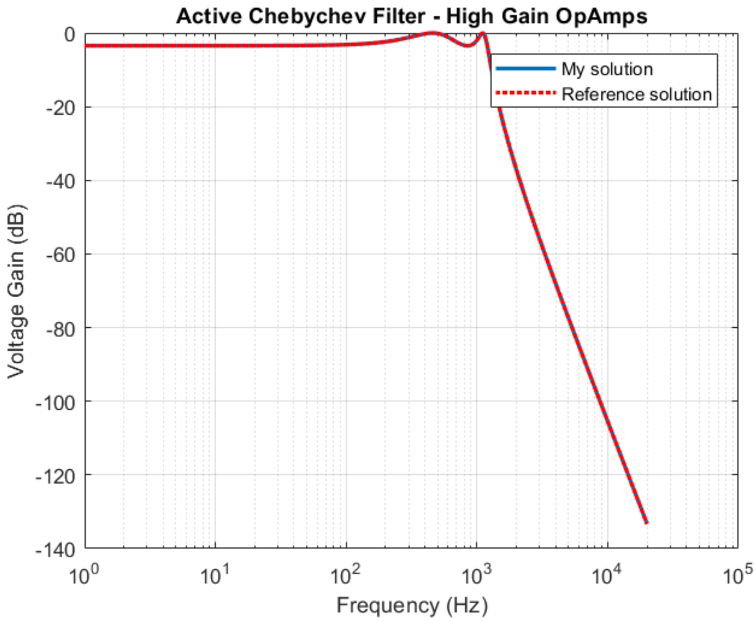

Figure 3: Magnitude Response of the Large Gain Chebychev Filter

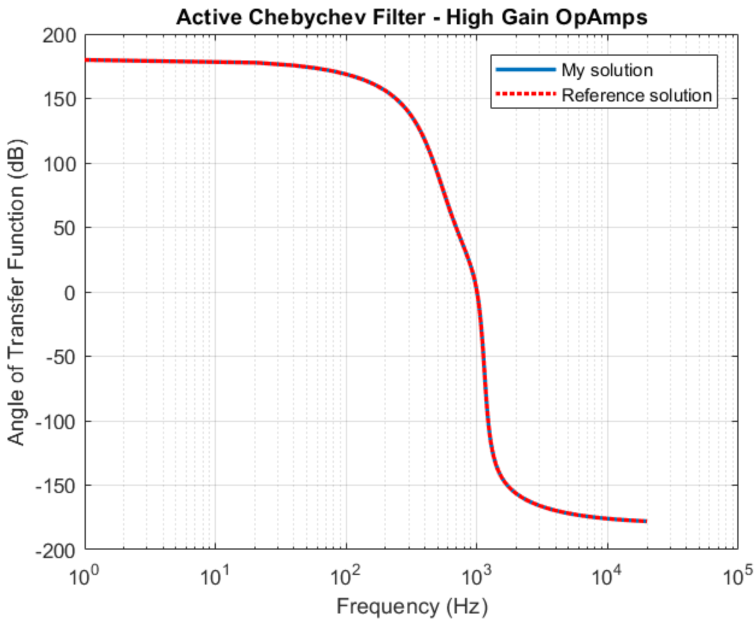

Figure 4: Angular Response of the Large Gain Chebychev Filter

3 Circuit 4 Test Bench

Figures 5 and 6 below show the magnitude and angular response of the Circuit 4 Test Bench.

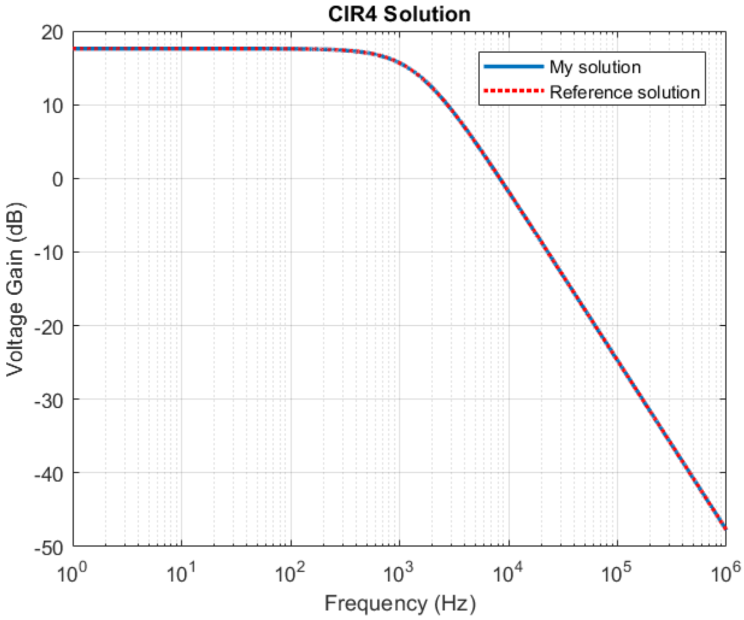

Figure 5: Magnitude Response of the Circuit 4 Test Bench

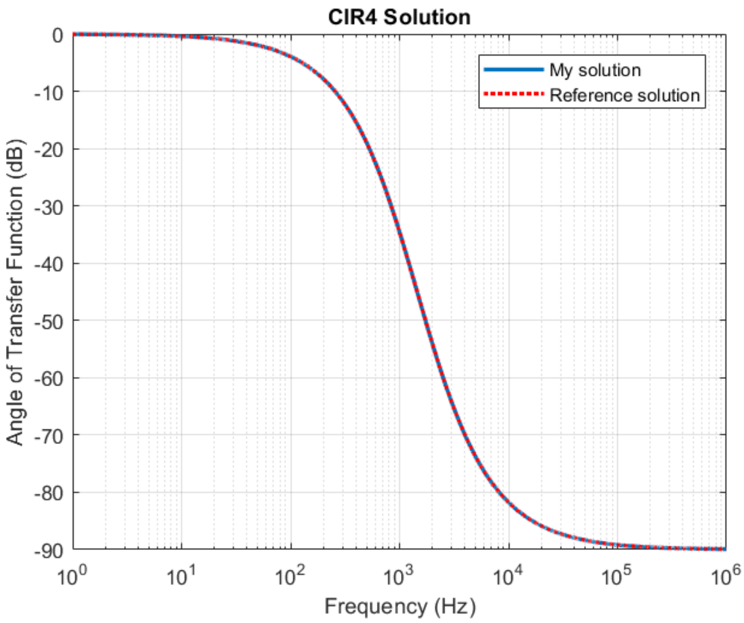

Figure 6: Angular Response of the Circuit 4 Test Bench

4 LC Filter 1 Test Bench

Figures 7 and 8 below show the magnitude and angular response of the LC Filter 1 Test Bench.

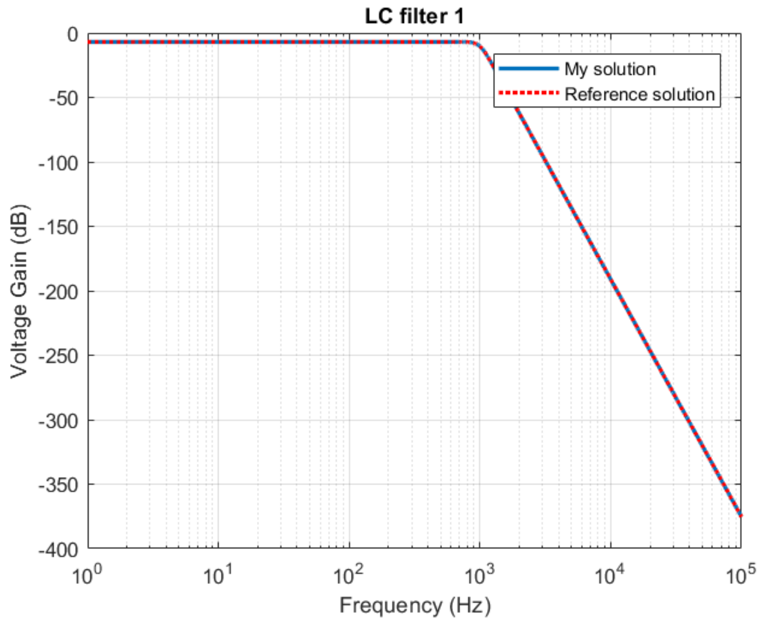

Figure 7: Magnitude Response of the LC Filter 1 Test Bench

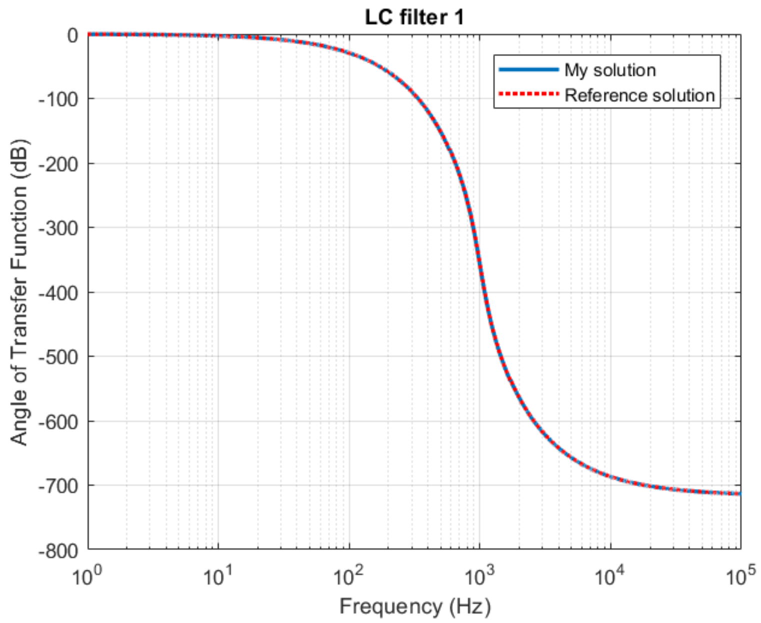

Figure 8: Angular Response of the LC Filter 1 Test Bench

5 LC Filter 2 Test Bench

Figures 9 and 10 below show the magnitude and angular response of the LC Filter 2 Test Bench.

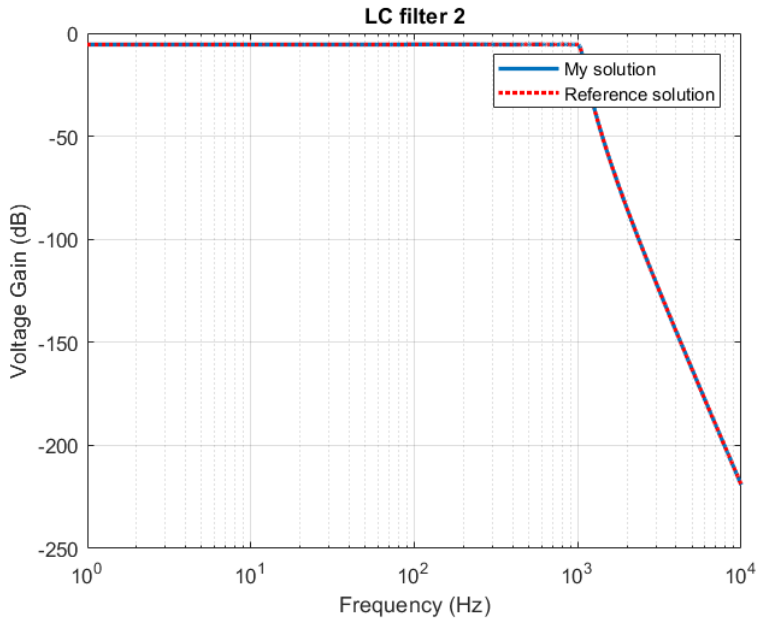

Figure 9: Magnitude Response of the LC Filter 2 Test Bench

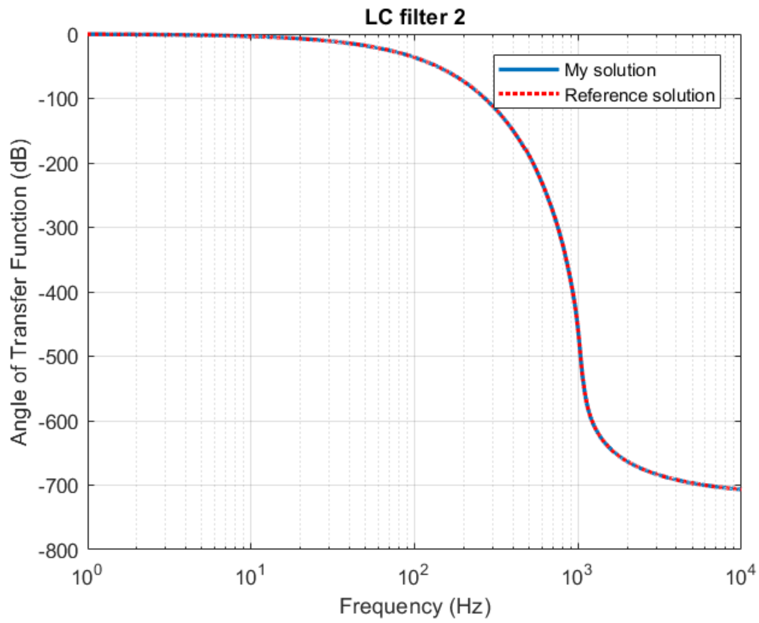

Figure 10: Angular Response of the LC Filter 2 Test Bench

6 LC Filter 3 Test Bench

Figures ?? and 12 below show the magnitude and angular response of the LC Filter 3 Test Bench.

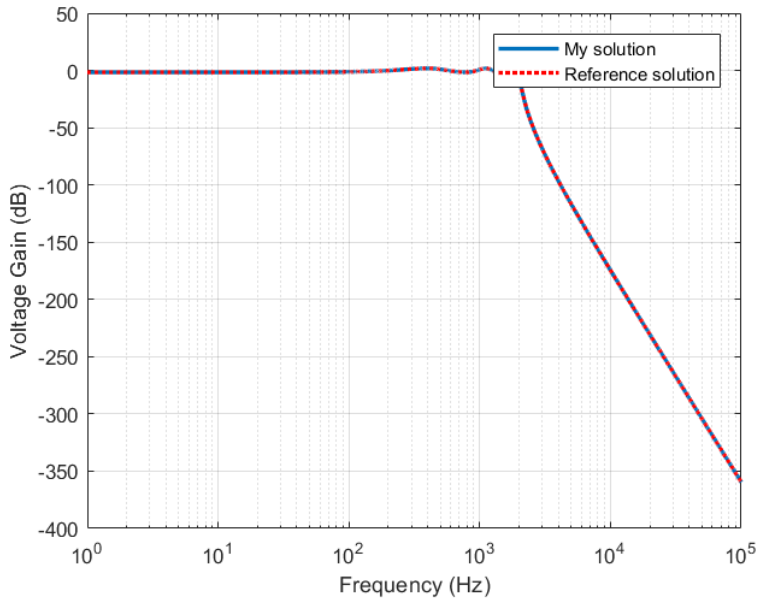

Figure 11: Magnitude Response of the LC Filter 3 Test Bench

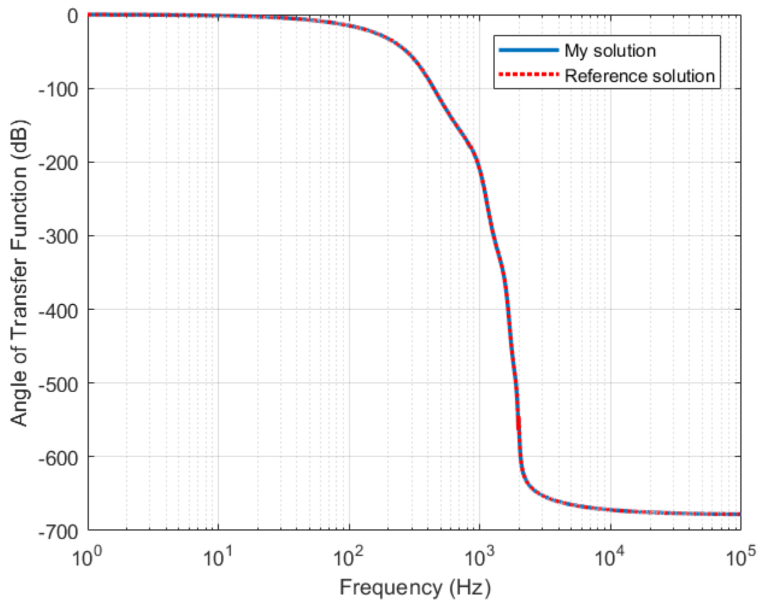

Figure 12: Angular Response of the LC Filter 3 Test Bench

A Code Listings

Listing 1: MATLAB Function to Compute the Frequency Response (`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
%
    global G C b

    r = zeros(1, size(fpoints, 2));

    for i = 1:size(fpoints, 2)
        X = (G + 2 * pi * fpoints(i) * 1j * C) \ b;
        r(1, i) = X(out);
    end
end
```

Listing 2: MATLAB Function to Generate Inductance Matrices (`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

     % Add additional entries to G, C, and b matrices.
     G(end + 1, :) = 0;
     G(:, end + 1) = 0;
     C(end + 1, :) = 0;
     C(:, end + 1) = 0;
     b(end + 1) = 0;

     % Modify the G matrix.
     if n1 ~= 0
         G(end, n1) = 1;
         G(n1, end) = 1;
     end

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

     % Modify the C matrix.
     C(end, end) = -val;
end

```

Listing 3: MATLAB Function to Generate VCCS Matrices (`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
end
```

Listing 4: MATLAB Function to Generate VCVS Matrices (vccs.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

    % Extend the matrices.
    G(end + 1, :) = 0;
    G(:, end + 1) = 0;
    b(end + 1) = 0;
    C(end + 1, :) = 0;
    C(:, end + 1) = 0;

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

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

    if ni1 ~= 0
        G(end, ni1) = -val;
    end
    if ni2 ~= 0
        G(end, ni2) = val;
```

```
    end
end
```