Table of Contents

Assignment 4 Part 3

In this part of the assignment, noise was added to the circuit, and the response to this noise was observed

In part a), a current source In was added to the circuit, in parallel with R3. This causes thermal noise to be generated in resistor R3.

In part b), a capacitor Cn = 0.00001 added in parallel with resistor to BW limit the noise. This causes changes to C matrix

Definition of variables based on the components present in the circuit

```
R1 = 1;
G1 = 1/R1;
c = 0.25;
R2 = 2;
G2 = 1/R2;
L = 0.2;
R3 = 10;
G3 = 1/R3;
alpha = 100;
R4 = 0.1;
G4 = 1/R4;
RO = 1000;
GO = 1/RO;
Vin = 1;
Cn_1 = 0.00001;                 % Capacitance value given in part b)
Cn_2 = 10^-8;                   % Cn_2 and Cn_3 are for part d) vi.
 where
Cn_3 = 2.6*2e-5;                  % we change Cn to observe the change
 in BW

% part a) Updating the C matrices

C_Matrix1 = [0 0 0 0 0 0 0;
            -c c 0 0 0 0 0;
             0 0 -L 0 0 0 0;
             0 0 0 -Cn_1 0 0 0;
             0 0 0 0 0 0 0;
             0 0 0 -Cn_1 0 0 0;
             0 0 0 0 0 0 0;];

C_Matrix2 = [0 0 0 0 0 0 0;
            -c c 0 0 0 0 0;
             0 0 -L 0 0 0 0;
             0 0 0 -Cn_2 0 0 0;
```

```
               0 0 0 0 0 0 0;
               0 0 0 -Cn_2 0 0 0;
               0 0 0 0 0 0 0;];

C_Matrix3 = [0 0 0 0 0 0 0;
              -c c 0 0 0 0 0;
               0 0 -L 0 0 0 0;
               0 0 0 -Cn_3 0 0 0;
               0 0 0 0 0 0 0;
               0 0 0 -Cn_3 0 0 0;
               0 0 0 0 0 0 0;];

GO = [1 0 0 0 0 0 0;
    -G2 G1+G2 -1 0 0 0 0;
      0 1 0 -1 0 0 0;
      0 0 -1 G3 0 0 0;
      0 0 0 0 -alpha 1 0;
      0 0 0 G3 -1 0 0;
      0 0 0 0 0 -G4 G4+GO];

F_Matrix = [Vin;
               0;
               0;
               0;
               0;
               0;
               0;];

F0_Matrix = [Vin-Vin;
                  0;
                  0;
                  0;
                  0;
                  0;
                  0;];

step_1 = 1000;
step_2 = 1.9898e4;          % time step 1 was the numer of time steps
 given
                            %   in part 2, step 2 is used
                            % to observe the effects of varying this
 value
                            % on the simulation

vol_1 = zeros(7, step_1);
vol_start = zeros(7, 1);


dt_1 = 10^-3;
dt_2 = 1.9898*10^-4;  %varying the timestep to observe the effects on
 the sim

% Circuit with Noise simulation with default time step
% Time domain simulation
```

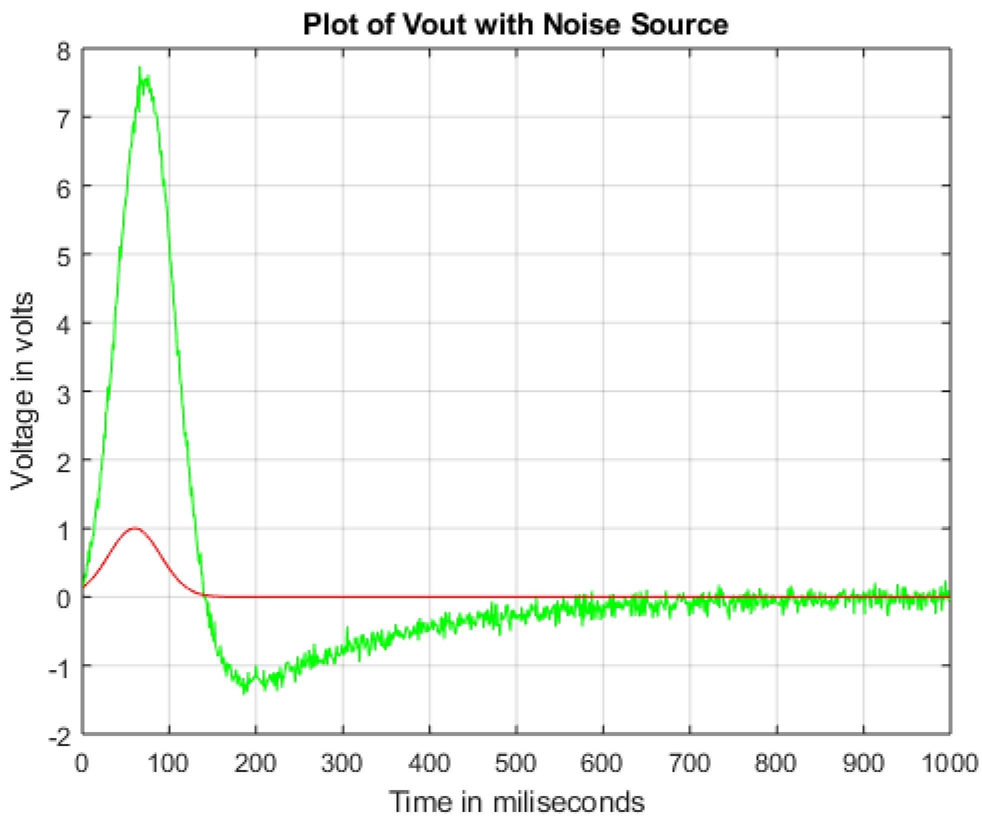
```
%vol_1 = zeros(7, step_1);
Guassian_F = zeros(7,1);

for i = 1:step_1

    Guassian_F(1,1) = exp(-1/2*((i/step_1-0.06)/(0.03))^2);
    Guassian_F(4,1) = 0.001*randn();
    Guassian_F(7,1) = 0.001*randn();

    if i == 1
        vol_1(:,i) = (C_Matrix1./dt_1+GO)\(Guassian_F
+C_Matrix1*vol_start/dt_1);

    else
        vol_1(:,i) = (C_Matrix1./dt_1+GO)\(Guassian_F
+C_Matrix1*vol_old/dt_1);

    end

    vol_old = vol_1(:, i);

end

% Part b)i. modelling Vo signal with noise using the Guassian
 excitation

figure(1)
plot(1:step_1, vol_1(7,:), 'g')
hold on
plot(1:step_1, vol_1(1,:), 'r')
title('Plot of Vout with Noise Source')
xlabel('Time in miliseconds')
ylabel('Voltage in volts')
grid on


freq = (-step_1/2:step_1/2-1);

fft_vol1 = fft(vol_1.');
ffts_vol1 = fftshift(fft_vol1);

%Part c) Fourier Transform plot
figure(2)
plot(freq, abs(ffts_vol1(:, 1)), 'g')
hold on
plot(freq, abs(ffts_vol1(:, 7)), 'r')
title('Fourier-Transform Plot of Vout')
xlabel('frequency in 1/ms')
ylabel('Voltage in volts')
grid on


vol_2 = zeros(7, step_1);
Guassian_F = zeros(7,1);
```

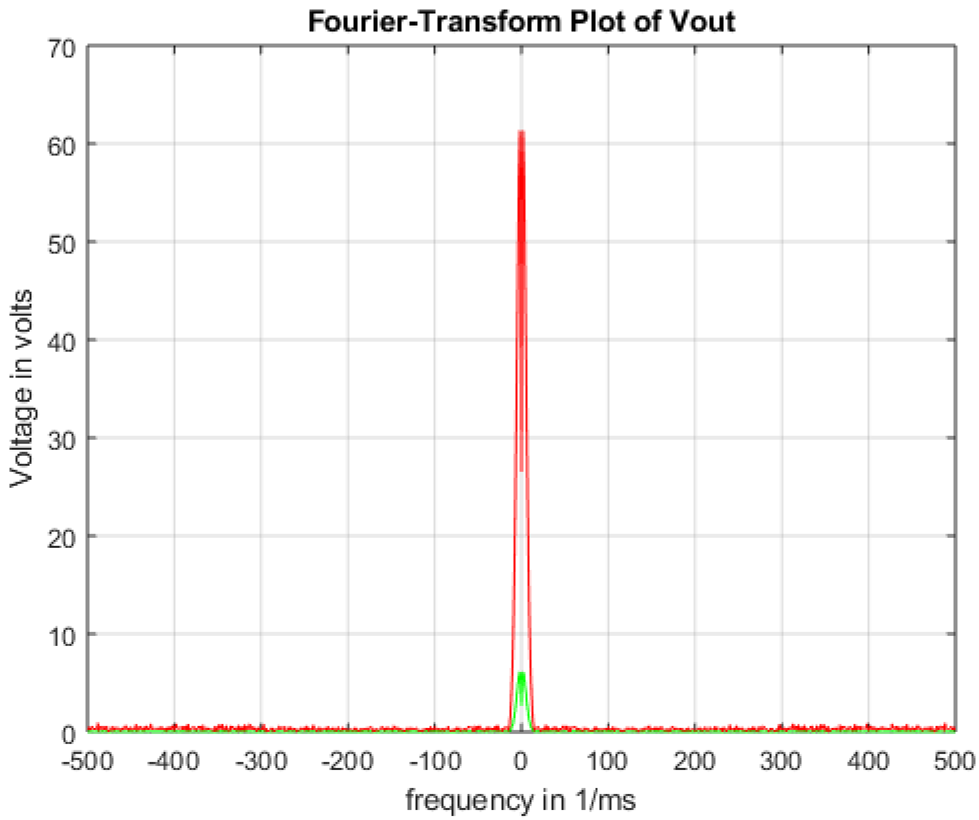
```
for i_2 = 1:step_1

    Guassian_F(1,1) = exp(-1/2*((i_2/step_1-0.06)/(0.03))^2);
    Guassian_F(4,1) = 0.001*randn();
    Guassian_F(7,1) = 0.001*randn();

    if i_2 == 1

        vol_2(:,i_2) = (C_Matrix2./dt_1+GO)\(Guassian_F
+C_Matrix2*vol_start/dt_1);

    else

        vol_2(:,i_2) = (C_Matrix2./dt_1+GO)\(Guassian_F
+C_Matrix2*vol_old/dt_1);

    end

    vol_old = vol_2(:, i_2);

end

%   e) in part e), 3 plots of vout will be made, with each plot made
 using
%   a different value of Cout. A discussion on my findings is placed
 at the
%   end of this document.


% plotting Vout using smaller value of Cout
figure(3)
plot(1:step_1, vol_2(7,:), 'g')
hold on
plot(1:step_1, vol_2(1,:), 'r')
title('Vout plot using smaller value of Cout')
xlabel('Time im milliseconds)')
ylabel('Voltage in volts')
grid on

vol_3 = zeros(7, step_1);
Guassian_F = zeros(7,1);

for i_3 = 1:step_1

    Guassian_F(1,1) = exp(-1/2*((i_3/step_1-0.06)/(0.03))^2);
    Guassian_F(4,1) = 0.001*randn();
    Guassian_F(7,1) = 0.001*randn();

    if i_3 == 1

        vol_3(:,i_3) = (C_Matrix3./dt_1+GO)\(Guassian_F
+C_Matrix3*vol_start/dt_1);
```

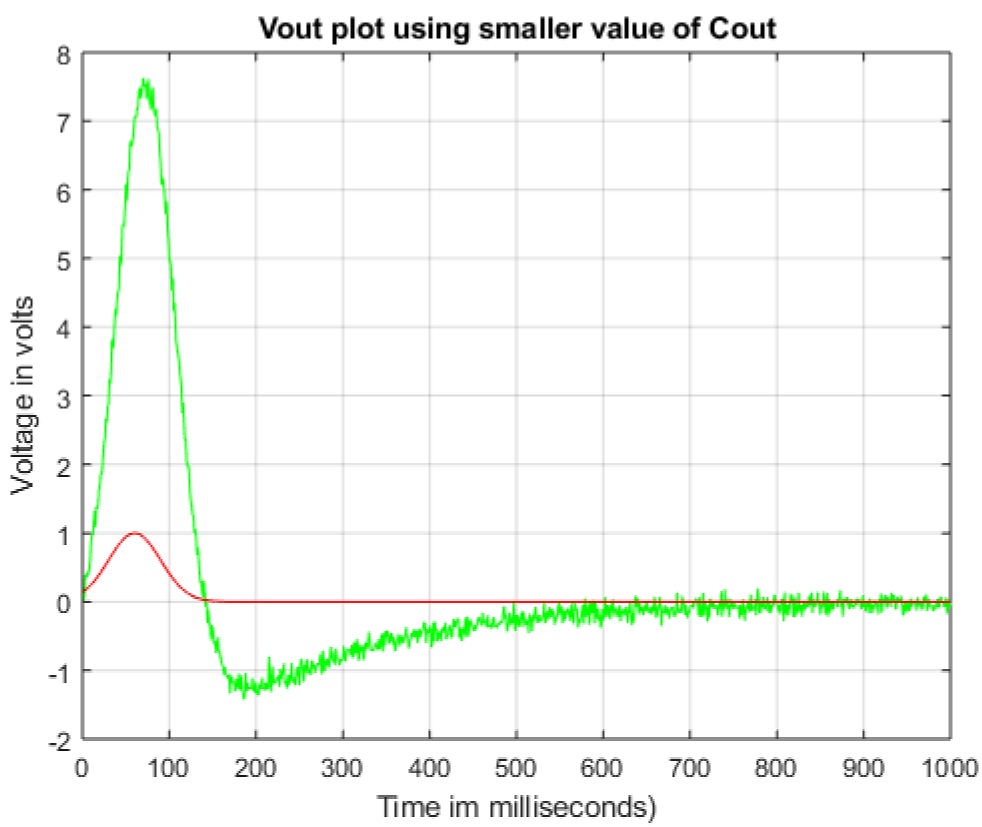
```
    else

        vol_3(:,i_3) = (C_Matrix3./dt_1+GO)\(Guassian_F
+C_Matrix3*vol_old/dt_1);

    end

    vol_old = vol_3(:, i_3);

end


figure(4)
plot(1:step_1, vol_3(7,:), 'g')
hold on
plot(1:step_1, vol_3(1,:), 'r')
title('Vout plot using bigger value of Cout')
xlabel('Time in millseconds')
ylabel('Voltage in volts')
grid on

%Now we will plot Vout using the value of Cout given
for i_4 = 1:step_1

    Guassian_F(1,1) = exp(-1/2*((i_4/step_1-0.06)/(0.03))^2);
    Guassian_F(4,1) = 0.001*randn();
    Guassian_F(7,1) = 0.001*randn();

    if i_4 == 1

        vol_3(:,i_4) = (C_Matrix1./dt_1+GO)\(Guassian_F
+C_Matrix1*vol_start/dt_1);

    else

        vol_3(:,i_4) = (C_Matrix1./dt_1+GO)\(Guassian_F
+C_Matrix1*vol_old/dt_1);

    end

    vol_old = vol_3(:, i_4);

end


figure(5)
plot(1:step_1, vol_3(7,:), 'g')
hold on
plot(1:step_1, vol_3(1,:), 'r')
title('Vout plot using original of Cout of 0.00001')
xlabel('Time in millseconds')
ylabel('Voltage in volts')
grid on
```

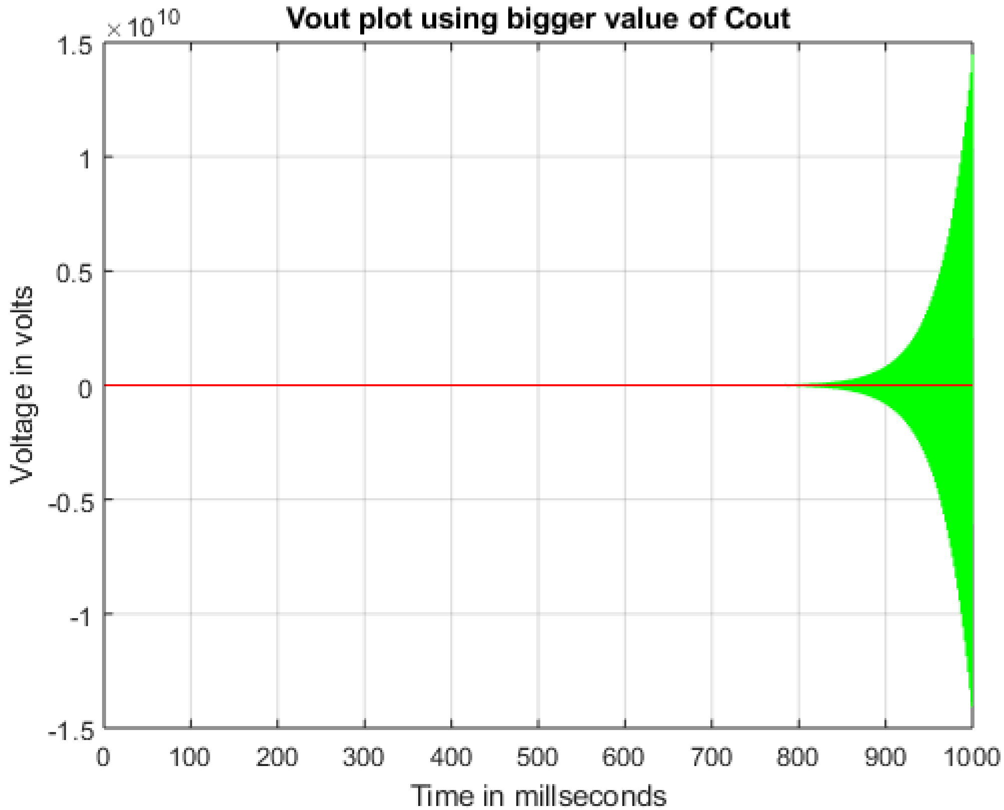
```
vol_7 = zeros(7, step_1);
Guassian_F = zeros(7,1);

%In part f), we are observing the effects of plotting Vout with
 differet
%time steps. A discussion on my findings will take place at the end of
 this
%document.

%Note that the original timestep was used in part b), and will not be
%replotted to avoid redundancy.

%using the timestep given

for i_5 = 1:step_1

    Guassian_F(1,1) = exp(-1/2*((i_5/step_1-0.06)/(0.03))^2);
    Guassian_F(4,1) = 0.001*randn();
    Guassian_F(7,1) = 0.001*randn();

    if i_5 == 1
        vol_7(:,i_5) = (C_Matrix1./dt_1+GO)\(Guassian_F
+C_Matrix1*vol_start/dt_1);

    else
        vol_7(:,i_5) = (C_Matrix1./dt_1+GO)\(Guassian_F
+C_Matrix1*vol_old/dt_1);

    end

    vol_old = vol_7(:, i_5);

end

figure(6)
plot(1:step_1, vol_7(7,:), 'g')
hold on
plot(1:step_1, vol_7(1,:), 'r')
title('Vout plot using original timestep of 10^-3')
xlabel('Time in picoseconds')
ylabel('Voltage in volts')
grid on

%using a smaller timestep

for i_6 = 1:step_2

    Guassian_F(1,1) = exp(-1/2*((i_6/step_2-0.06)/(0.03))^2);
    Guassian_F(4,1) = 0.001*randn();
    Guassian_F(7,1) = 0.001*randn();

    if i_6 == 1
        vol_4(:,i_6) = (C_Matrix1./dt_2+GO)\(Guassian_F
+C_Matrix1*vol_start/dt_2);
```

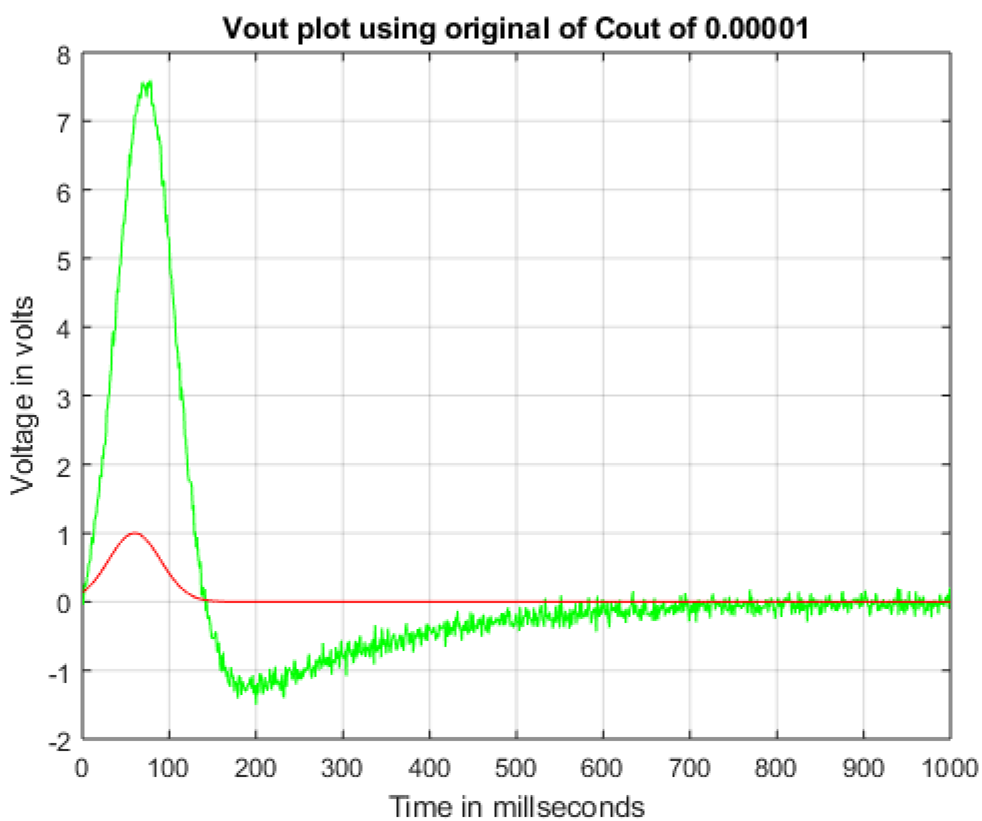
```
    else
        vol_4(:,i_6) = (C_Matrix1./dt_2+GO)\(Guassian_F
+C_Matrix1*vol_old/dt_2);

    end

    vol_old = vol_4(:, i_6);

end


figure(7)
plot(1:step_2, vol_4(7,:), 'g')
hold on
plot(1:step_2, vol_4(1,:), 'r')
title('Vout plot using a smaller timestep')
xlabel('Time in picoseconds')
ylabel('Voltage in volts')
grid on
```

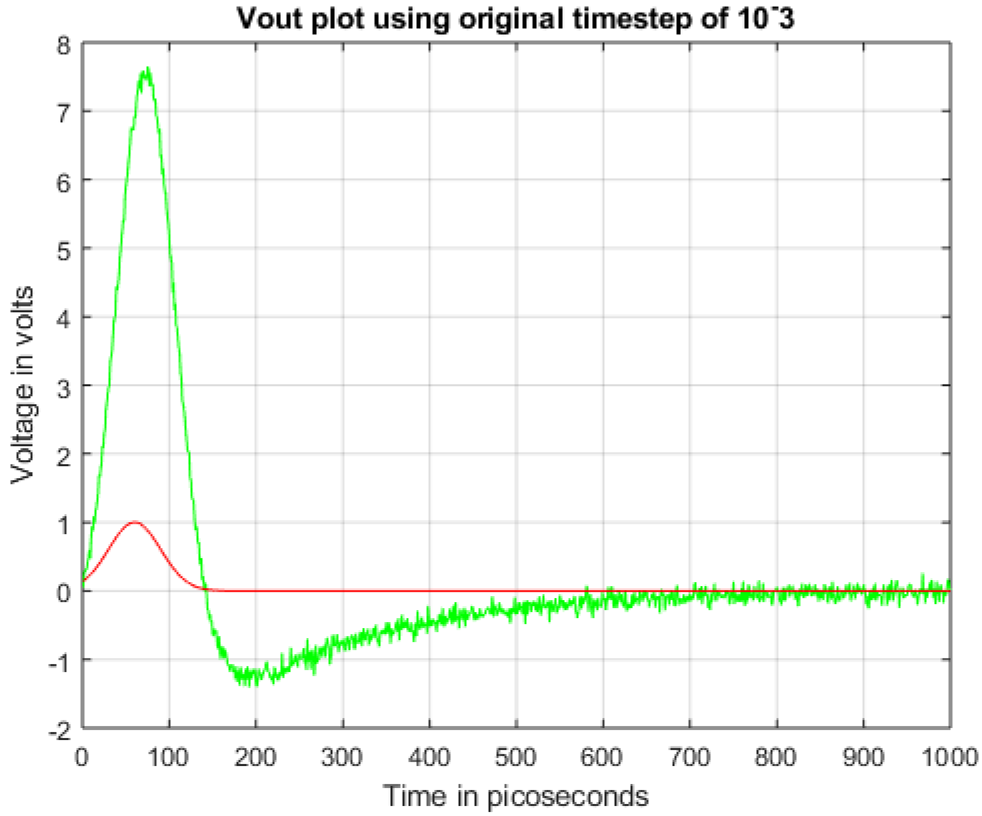


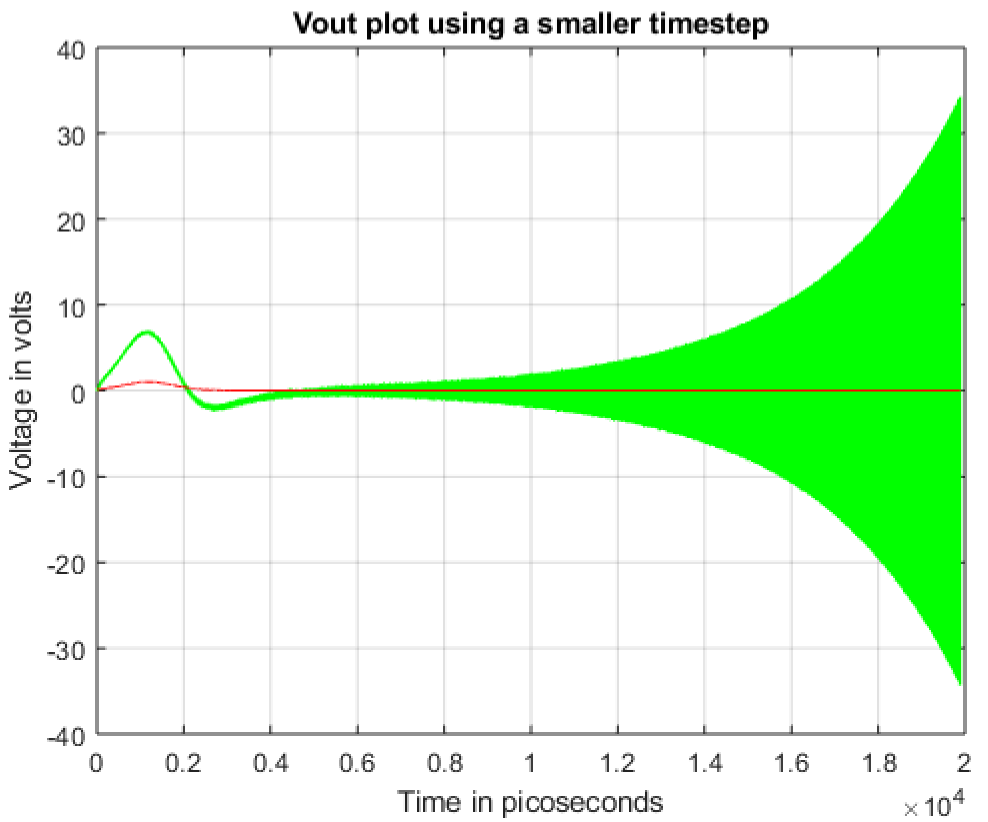

Fourier-Transform Plot of Vout
Voltage in volts
70
60
50
40
30
20
10
0
-500
-400
-300
-200
-100
0
100
200
300
400
500
frequency in 1/ms


Vout plot using smaller value of Cout
Voltage in volts
8
7
6
5
4
3
2
1
0
-1
-2
0
100
200
300
400
500
600
700
800
900
1000
Time im milliseconds)


Vout plot using bigger value of Cout
×10^10
1.5
1
0.5
0
-0.5
-1
-1.5
Voltage in volts
0 100 200 300 400 500 600 700 800 900 1000
Time in millseconds


Vout plot using original of Cout of 0.00001
8
7
6
5
4
3
2
1
0
-1
-2
Voltage in volts
0 100 200 300 400 500 600 700 800 900 1000
Time in millseconds


Vout plot using original timestep of 10ˉ3
Voltage in volts
Time in picoseconds


Vout plot using a smaller timestep
Voltage in volts
Time in picoseconds
×10^4

Discussion

In part e), 3 plots of Vout were made using 3 values of Cout. It is noticed that using a smaller value of Cout does not change the plot. However, a larger value of Cout causes the simulation to break down. This is because the circuit becomes trapped in a feedback loop.

In part f), 2 plots of Vout were made using different timesteps. Using a smaller timestep than the original timestep of 10^-3 causes the simulation to break down because the circuit becomes trapped in a feedback loop.

PART 4

If the e voltage source on the output stage described by the transconductance equation V = alpha*I3 was instead modeled by V = alpha*I3 +beta*I2^2 + gamma*I3^3, the changes required in my simulator are more than just simply changing the matrices we used in the simulation for part 3. This new equation would need to be fitted, And new matrices would have to be created for the Jacobian method for the simulation of the new equation in the circuit. Note that the inclusion of this equation would in turn increase the size of the matrix and the iterations required to traverse through it. The values of alpha and beta that we define will have to be considerably large, as if i3 is smaller than 1 this will be too small to have a noticeable effect on the simulation.

Published with MATLAB® R2018b