Table of Contents

This file implement the ELEC 4700 PA7

Goal: Do a DC and AC analysis of a linear circuit using MNA techniques

```
% Clear all
clearvars
clearvars -global
close all

% Save some component values
R1 = 1;
C = 0.25;
R2 = 2;
L = 0.2;
R3 = 10;
alpha = 100;
R4 = 0.1;
RO=1000;

% Declare the vectors
vectorV = zeros(9, 1);  % solution vector: [N1, N2, N3, N4, N5, I1, IL, I3,
 I4]
vectorF = zeros(9, 1);  % F vector: F(1) = VIN
```

a) Create the C, G matrices

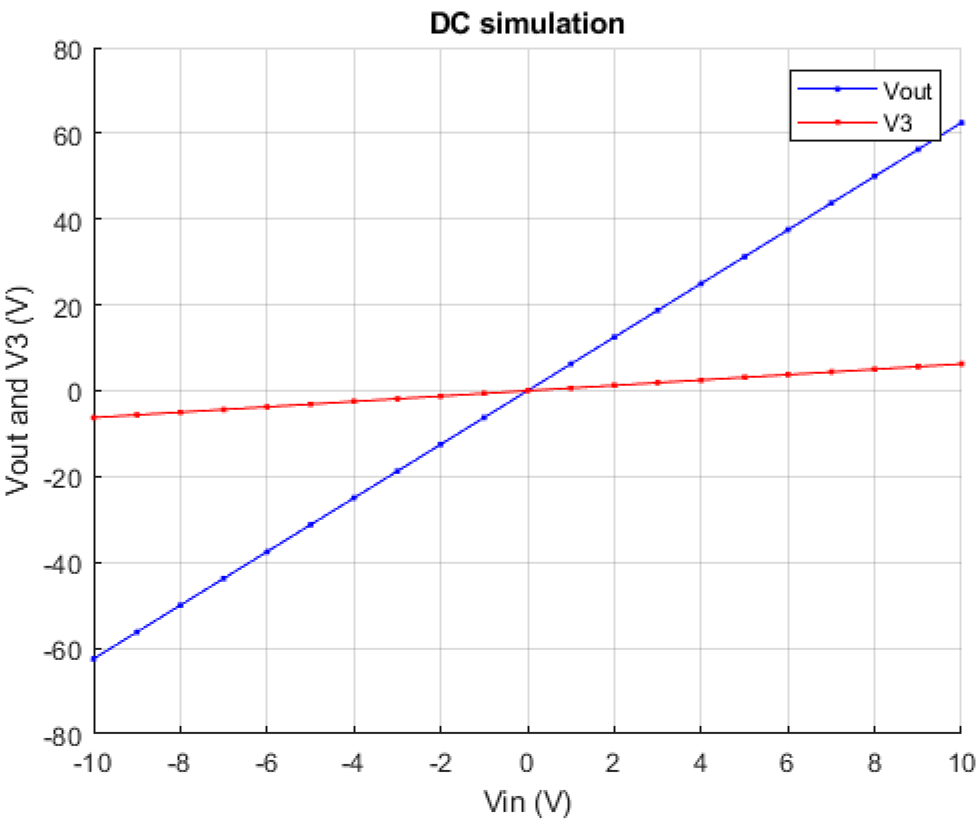
Declare the C matrix

```
matrixC = [0, 0, 0, 0, 0, 0, 0, 0, 0;
           C, -C, 0, 0, 0, 0, 0, 0, 0;
          -C, C, 0, 0, 0, 0, 0, 0, 0;
           0, 0, 0, 0, 0, 0, -L, 0, 0;
           0, 0, 0, 0, 0, 0, 0, 0, 0;
           0, 0, 0, 0, 0, 0, 0, 0, 0;
           0, 0, 0, 0, 0, 0, 0, 0, 0;
           0, 0, 0, 0, 0, 0, 0, 0, 0;
           0, 0, 0, 0, 0, 0, 0, 0, 0];

% Declare the G matrix
matrixG = [1,     0,         0,     0,           0, 0,   0,       0, 0;
          1/R1, -1/R1,       0,     0,           0, 1,   0,       0, 0;
```

```
            -1/R1, 1/R1+1/R2, 0,     0,           0, 0,   1,       0, 0;
              0,     1,        -1,     0,           0, 0,   0,       0, 0;
              0,     0,         0,     0,           0, 0,  -1,       1, 0;
              0,     0,     -1/R3,     0,           0, 0,   0,       1, 0;
              0,     0,         0,  1/R4,       -1/R4, 0,   0,       0, 1;
              0,     0,         0,     1,           0, 0,   0,  -alpha, 0;
              0,     0,         0, -1/R4,  1/R4+1/RO, 0,   0,       0, 0];
```

b) DC sweep input voltage from -10V to 10V and plot V0 and V3 (N3)

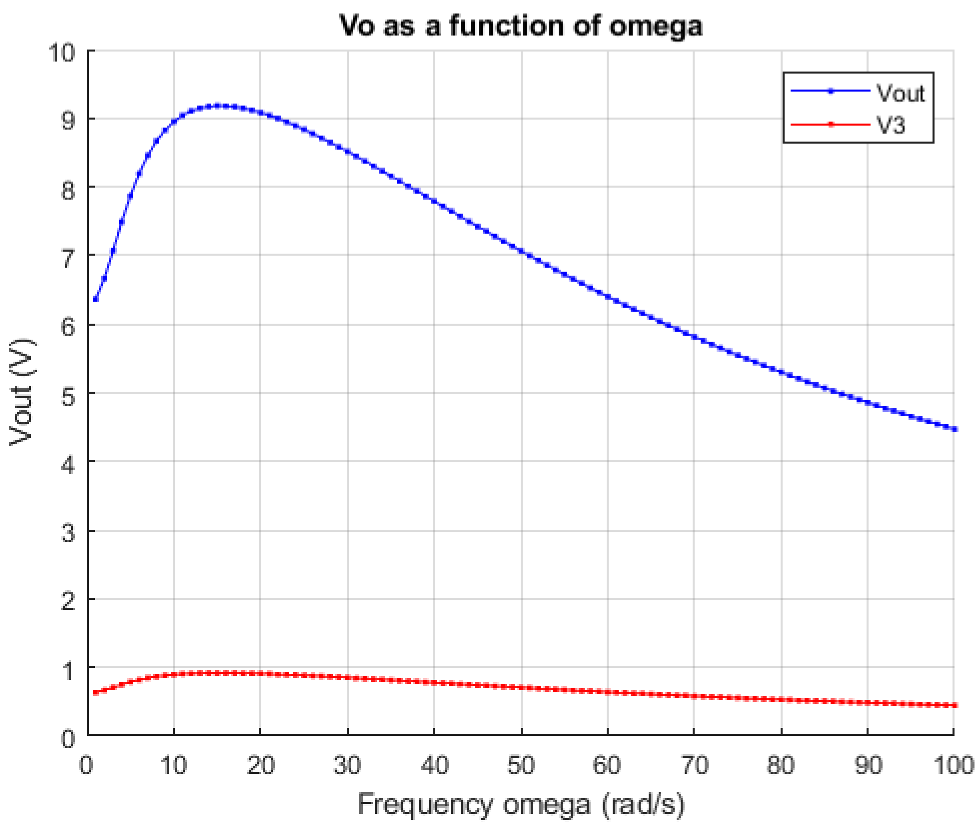
```
simStep = 21; % Simulation steps
V1 = linspace(-10, 10, simStep);  % vector for input voltages
Vo = zeros(simStep, 1);  % vector for holding the output voltage
V3 = zeros(simStep, 1);  % vector for holding the voltage at V3
% Loop for the DC simulation
for iSim = 1:simStep
    % Setup the F vector
    vectorF(1) = V1(iSim);  % Stepup the input voltage
    % Find the solution
    vectorV = matrixG\vectorF;
    % Save answers
    Vo(iSim) = vectorV(5);  % Save Vout
    V3(iSim) = vectorV(3);  % Save V3
end
% Plot the DC simulation
figure(1)
hold on
plot(V1, Vo, "-b.")
plot(V1, V3, "-r.")
title("DC simulation")
xlabel("Vin (V)")
ylabel("Vout and V3 (V)")
legend("Vout", "V3")
grid on;
```

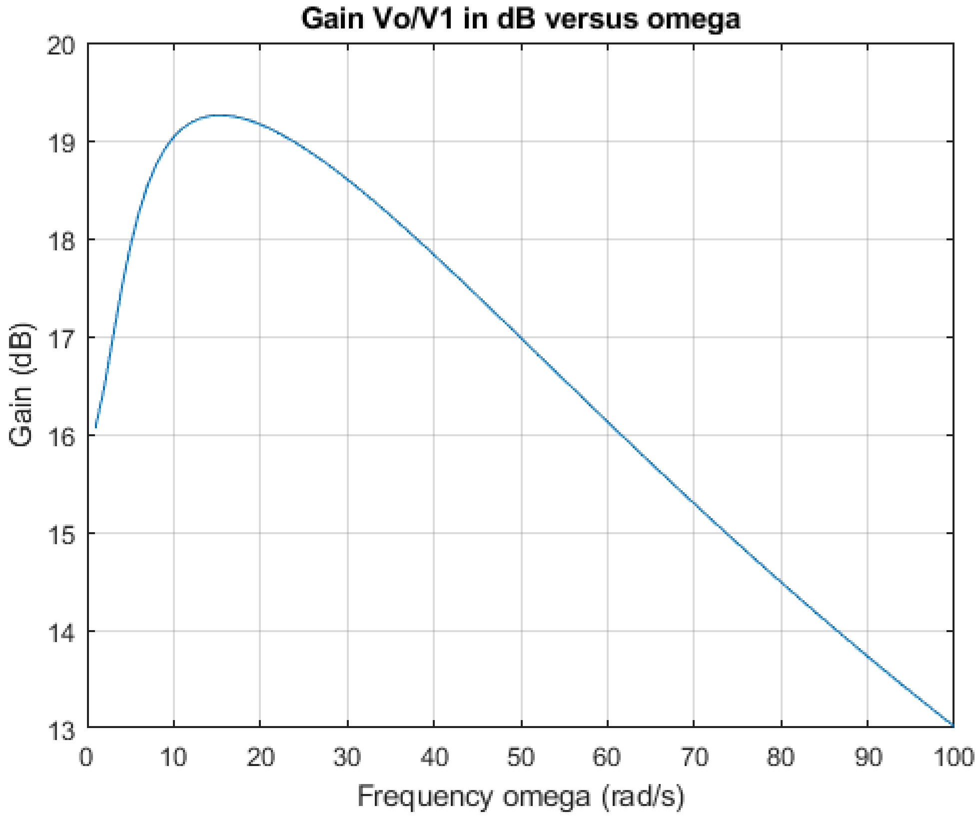


c) AC case plot Vo as a function of omega and plot the gain Vo/V1 in dB

```
simSteps = 100;  % Simulation steps
Vin = 1;  % Value for input voltage
vectorF(1) = Vin;  % Setup the input voltage
omega = linspace(1, 100, simSteps);  % vector for frequencies
Vo = zeros(simSteps, 1);  % vector store the output voltages
V3 = zeros(simStep, 1);  % vector for holding the voltage at V3

% Loop for simulation
for iSim = 1:simSteps
    w = omega(iSim);  % Retrieve the simulation frequency
    % Construct the G+jwC matrix
    matrixGC = matrixG + 1j*w*matrixC;
    % Find the solution
    vectorV = matrixGC\vectorF;
    % Save answers
    Vo(iSim) = abs(vectorV(5));  % Save Vout
    V3(iSim) = abs(vectorV(3));  % Save V3
end
% Plot Vo as a function of omega
figure(2)
hold on
```

```
plot(omega, Vo, "-b.");
plot(omega, V3, "-r.");
title("Vo as a function of omega")
xlabel("Frequency omega (rad/s)")
ylabel("Vout (V)")
legend("Vout", "V3")
grid on

% Plot the gain Vo/V1 in dB
figure(3)
gain = 20*log10(Vo ./ Vin);  % Calculate the gain in dB
plot(omega, gain);
title("Gain Vo/V1 in dB versus omega")
xlabel("Frequency omega (rad/s)")
ylabel("Gain (dB)")
grid on
```

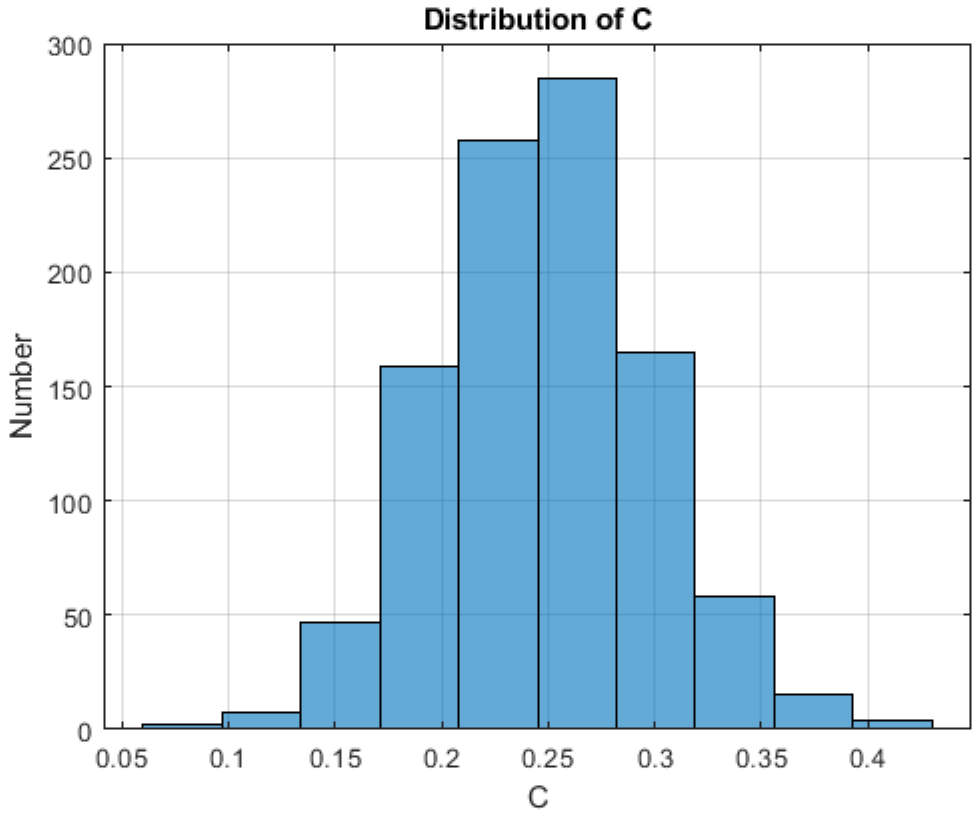


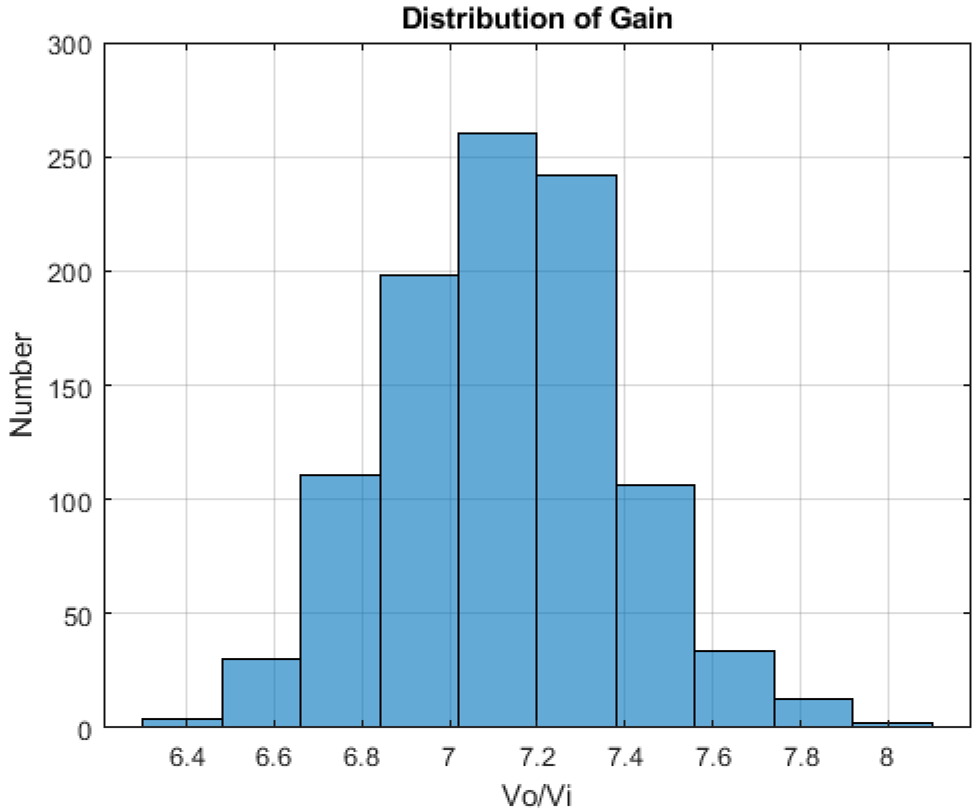


d) Plot gain as a random perturbations on C at omega=pi

```
simSteps = 1000;  % Simulation steps
omega = pi;
std = 0.05;  % Standard deviation of the normal distribution
randomC = std .* randn(simSteps, 1)+C;  % vector store the random C
Vo = zeros(simSteps, 1);  % vector store the output voltages
Vin = 10;  % Value for input voltage
vectorF(1) = Vin;  % Setup the input voltage

% Plot the normal distribution of C
nbins = 10;  % Number of bins for the histogram
figure(5)
histogram(randomC, nbins);
title("Distribution of C")
xlabel("C")
ylabel("Number")
grid on

% Loop through the random C
for iSim=1:simSteps
    C = randomC(iSim);  % Retrieve the C value
    % Reconstruct the C matrix
```

```
    matrixC = [0, 0, 0, 0, 0, 0, 0, 0, 0;
    C, -C, 0, 0, 0, 0, 0, 0, 0;
    -C, C, 0, 0, 0, 0, 0, 0, 0;
    0, 0, 0, 0, 0, 0, -L, 0, 0;
    0, 0, 0, 0, 0, 0, 0, 0, 0;
    0, 0, 0, 0, 0, 0, 0, 0, 0;
    0, 0, 0, 0, 0, 0, 0, 0, 0;
    0, 0, 0, 0, 0, 0, 0, 0, 0;
    0, 0, 0, 0, 0, 0, 0, 0, 0];

    % Construct the G+jwC matrix
    matrixGC = matrixG + 1j*omega*matrixC;
    % Find the solution
    vectorV = matrixGC\vectorF;
    % Save answers
    Vo(iSim) = abs(vectorV(5));  % Save Vout
end

% Plot the distribution of gain
figure(6)
gain = Vo ./ Vin;  % Calculate the gain
histogram(gain, nbins);
title("Distribution of Gain")
xlabel("Vo/Vi")
ylabel("Number")
grid on
```





Published with MATLAB® R2021b