EE3331C Feedback Control Systems Assignment 1

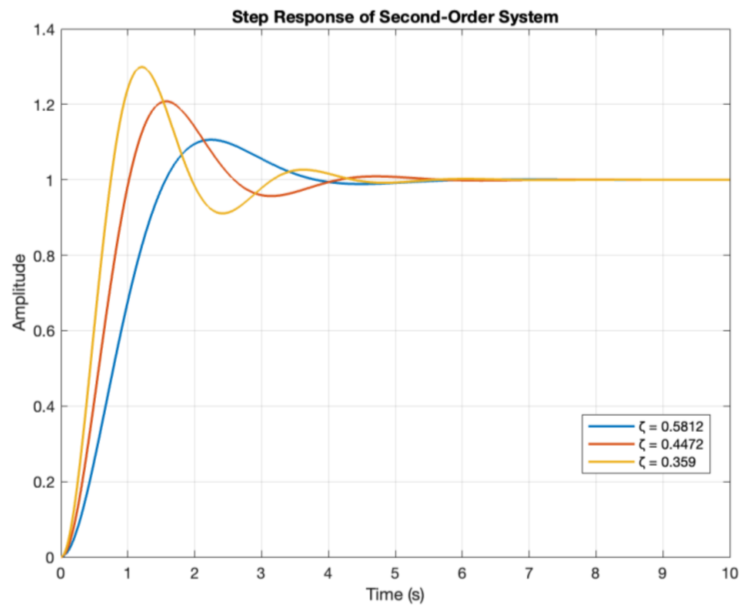
Code

```
%% SET MODE!
% a=1   b=2   c=3
MODE = 1;

% Define constants
syms s
a = 1;
b = 2;
c = 1;

% Set up system vars
if MODE == 1
    a_values = [a, a, a];
    b_values = [0.7*b, 1*b, 1.3*b];

    wn_results = zeros(1, 3);
    zeta_results = zeros(1, 3);
elseif MODE == 2
    a_values = [0.7*a, 1*a, 1.3*a];
    b_values = [b, b, b];

    wn_results = zeros(1, 3);
    zeta_results = zeros(1, 3);
elseif MODE == 3
    a_values = [a, a, a];
    b_values = [b, b, b];
    wn_values = [0.7*b, 1*b, 1.3*b];
    zeta = 0.1*a;

    wn_results = zeros(1, 3);
    zeta_results = zeros(1, 3);
end

% Solve for wn and zeta
for i = 1:3
    if MODE == 3
        term1 = s + zeta*wn_values(i) + wn_values(i)*sqrt(1-zeta^2)*1i;
        term2 = s + zeta*wn_values(i) - wn_values(i)*sqrt(1-zeta^2)*1i;
    else
        term1 = s + a_values(i) + b_values(i)*1i;
        term2 = s + a_values(i) - b_values(i)*1i;
    end

    % Multiply the terms
    result = term1 * term2;
```

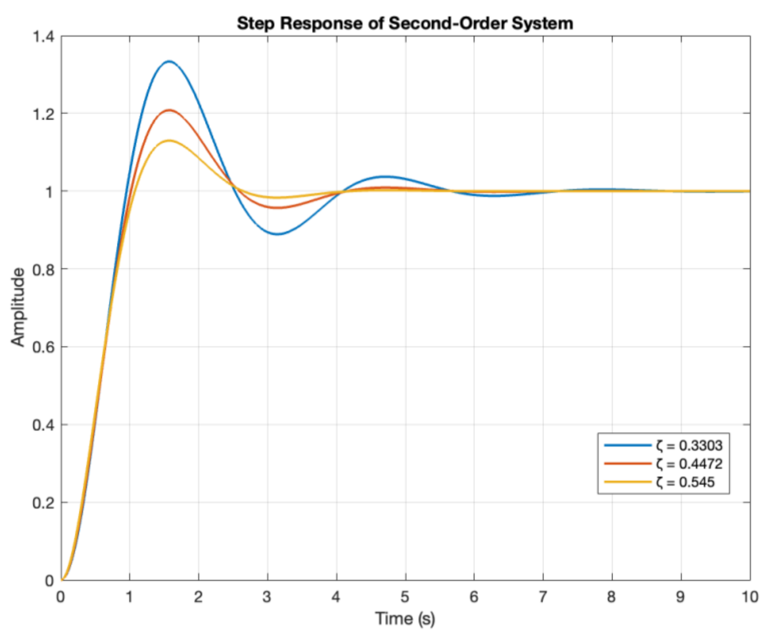
```
    % Expand resulting polynomial
    expanded_result = expand(result);

    % Get coefficients for x^1 and x^0
    coeffs_result = coeffs(expanded_result);
    coeff_x1 = coeffs_result(2);
    coeff_x0 = coeffs_result(1);

    % Calculate and store wn and zeta
    wn_results(i) = sqrt(coeff_x0)
    zeta_results(i) = coeff_x1 / (2 * wn_results(i))
end

% Create a time vector for simulation and a figure for plotting
t = 0:0.01:5;
figure;

% Loop through values of wn and zeta
for i = 1:3

    % Local vars
    zeta = zeta_results(i);
    wn = wn_results(i);
    K = c * wn;

    % Calculate the closed-loop transfer function
    num = K * wn;
    den = [1, 2 * zeta * wn, wn * wn];
    sys = tf(num, den)

    % Simulate the step response
    y = step(sys, t);

    % Plot the step response
    plot(t, y, 'LineWidth', 1.5, 'DisplayName', ['ζ = ', num2str(zeta)]);
    hold on;
end

% Format the figure
xlabel('Time (s)');
ylabel('Amplitude');
title('Step Response of Second-Order System');
legend('Location', 'best');
grid on;
hold off;
```

Graphs

a. In this one, since b is increasing between trials and a and c are being held constant, the w_n will increase drastically while the damping factor will decrease. This means that the poles are moving across just the imaginary axis. Because of that, the exponential envelope should remain the same between the three.



b. Since a is increasing between trials, and b and c are being held constant, the w_n will only increase slightly, while the damping factor will increase across runs. This means that the poles are moving across just the real axis. Because of that, the frequency should appear around the same between runs, but they will die out at different points.



c. In this example, since w_n and zeta are increased with time, it means the poles will move across both the real and imaginary axis. Because of that, the overshoot will be the same between runs, and the steady state is reached quickly because the pole is farther to the left of the imaginary axis.