Table of Contents

```
clc;
clear;
close all;
```

Load simulation data (skip header line)

```
fid = fopen('eecs215hw11p1.txt', 'r');
fgetl(fid);  % skip header
sim_data = textscan(fid, '%f %f %f %f');
fclose(fid);
t_sim = sim_data{1};
I_C1_sim = sim_data{2};
I_C2_sim = sim_data{3};
I_L1_sim = sim_data{4};
```

Define transfer functions from equations.txt

v_c1(s) = (12(90s^2-85s-2))/(1620s^3+180s^2+167s+9)

```
num1 = 12 * [90, -85, -2];
den1 = [1620, 180, 167, 9];

% v_c2(s) = (6(90s+11))/(1620s^3+180s^2+167s+9)
num2 = 6 * [90, 11];
den2 = [1620, 180, 167, 9];

% i_l(s) = (2(90s^2-s+18))/(s(90s^2+5s+9))
num3 = 2 * [90, -1, 18];
den3 = conv([1, 0], [90, 5, 9]);
```

Compute partial fraction decomposition

```
[r1, p1, k1] = residue(num1, den1);
[r2, p2, k2] = residue(num2, den2);
[r3, p3, k3] = residue(num3, den3);
```

Create time vector for analytical solution (dense for continuous plotting)

```
t = linspace(0, 50, 5000);
```

Compute inverse Laplace transforms (analytical)

For complex conjugate pairs, combine them properly If pole p has residue r, and p* has residue r*, then: r*exp(p*t) + r*exp(p**t) = 2*real(r*exp(p*t))

```
% v_c1(t)
v_c1 = zeros(size(t));
i = 1;
while i <= length(r1)
    if abs(imag(p1(i))) < 1e-10  % Real pole
        v_c1 = v_c1 + real(r1(i)) * exp(real(p1(i)) * t);
        i = i + 1;
    else  % Complex conjugate pair
        v_c1 = v_c1 + 2 * real(r1(i) * exp(p1(i) * t));
        i = i + 2;  % Skip conjugate
    end
end

% v_c2(t)
v_c2 = zeros(size(t));
i = 1;
while i <= length(r2)
    if abs(imag(p2(i))) < 1e-10  % Real pole
        v_c2 = v_c2 + real(r2(i)) * exp(real(p2(i)) * t);
        i = i + 1;
    else  % Complex conjugate pair
        v_c2 = v_c2 + 2 * real(r2(i) * exp(p2(i) * t));
        i = i + 2;  % Skip conjugate
    end
end

% i_l(t)
i_l = zeros(size(t));
i = 1;
while i <= length(r3)
    if abs(imag(p3(i))) < 1e-10  % Real pole
        i_l = i_l + real(r3(i)) * exp(real(p3(i)) * t);
        i = i + 1;
    else  % Complex conjugate pair
        i_l = i_l + 2 * real(r3(i) * exp(p3(i) * t));
        i = i + 2;  % Skip conjugate
    end
end
```

Plot results

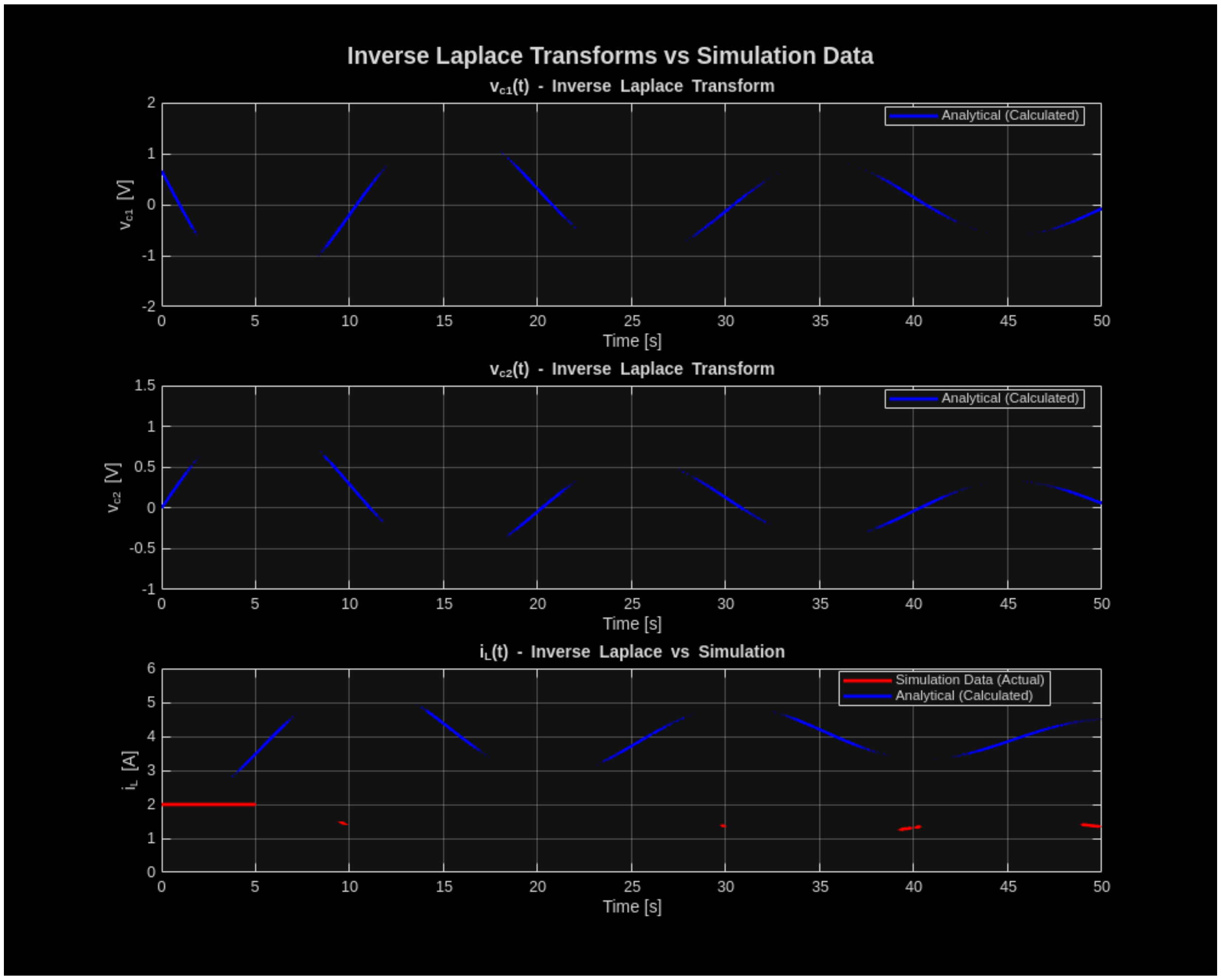
```
figure('Position', [100, 100, 1000, 800]);

% Plot v_c1
subplot(3, 1, 1);
plot(t, v_c1, 'b-', 'LineWidth', 2);
hold on;
grid on;
xlim([0 50]);
title('v_{c1}(t) - Inverse Laplace Transform');
xlabel('Time [s]');
ylabel('v_{c1} [V]');
legend('Analytical (Calculated)', 'Location', 'best');

% Plot v_c2
subplot(3, 1, 2);
plot(t, v_c2, 'b-', 'LineWidth', 2);
hold on;
grid on;
xlim([0 50]);
title('v_{c2}(t) - Inverse Laplace Transform');
xlabel('Time [s]');
ylabel('v_{c2} [V]');
legend('Analytical (Calculated)', 'Location', 'best');

% Plot i_l compared with simulation
subplot(3, 1, 3);
plot(t_sim, I_L1_sim, 'r-', 'LineWidth', 2);
hold on;
plot(t, i_l, 'b-', 'LineWidth', 2);
grid on;
xlim([0 50]);
title('i_L(t) - Inverse Laplace vs Simulation');
xlabel('Time [s]');
ylabel('i_L [A]');
legend('Simulation Data (Actual)', 'Analytical (Calculated)', 'Location',
'best');

sgtitle('Inverse Laplace Transforms vs Simulation Data', 'FontWeight',
'bold');
```



Print analytical expressions

```
fprintf('Inverse Laplace Transform Results:\n\n');

fprintf('v_c1(t) poles and residues:\n');
for i = 1:length(r1)
    if imag(p1(i)) == 0
        fprintf('  Pole: %.6f, Residue: %.6f\n', p1(i), r1(i));
    else
        fprintf('  Pole: %.6f %+.6fi, Residue: %.6f %+.6fi\n', real(p1(i)),
imag(p1(i)), real(r1(i)), imag(r1(i)));
    end
end

fprintf('\nv_c2(t) poles and residues:\n');
for i = 1:length(r2)
    if imag(p2(i)) == 0
        fprintf('  Pole: %.6f, Residue: %.6f\n', p2(i), r2(i));
    else
        fprintf('  Pole: %.6f %+.6fi, Residue: %.6f %+.6fi\n', real(p2(i)),
imag(p2(i)), real(r2(i)), imag(r2(i)));
```

```
    end
end

fprintf('\ni_l(t) poles and residues:\n');
for i = 1:length(r3)
    if imag(p3(i)) == 0
        fprintf('  Pole: %.6f, Residue: %.6f\n', p3(i), r3(i));
    else
        fprintf('  Pole: %.6f %+.6fi, Residue: %.6f %+.6fi\n', real(p3(i)),
imag(p3(i)), real(r3(i)), imag(r3(i)));
    end
end
```

```
Inverse Laplace Transform Results:

v_c1(t) poles and residues:
  Pole: -0.027778 +0.315005i, Residue: 0.222222 +1.077779i
  Pole: -0.027778 -0.315005i, Residue: 0.222222 -1.077779i
  Pole: -0.055556, Residue: 0.222222

v_c2(t) poles and residues:
  Pole: -0.027778 +0.315005i, Residue: -0.111111 -0.538889i
  Pole: -0.027778 -0.315005i, Residue: -0.111111 +0.538889i
  Pole: -0.055556, Residue: 0.222222

i_l(t) poles and residues:
  Pole: -0.027778 +0.315005i, Residue: -1.000000 +0.299819i
  Pole: -0.027778 -0.315005i, Residue: -1.000000 -0.299819i
  Pole: 0.000000, Residue: 4.000000
```

Published with MATLAB® R2025b