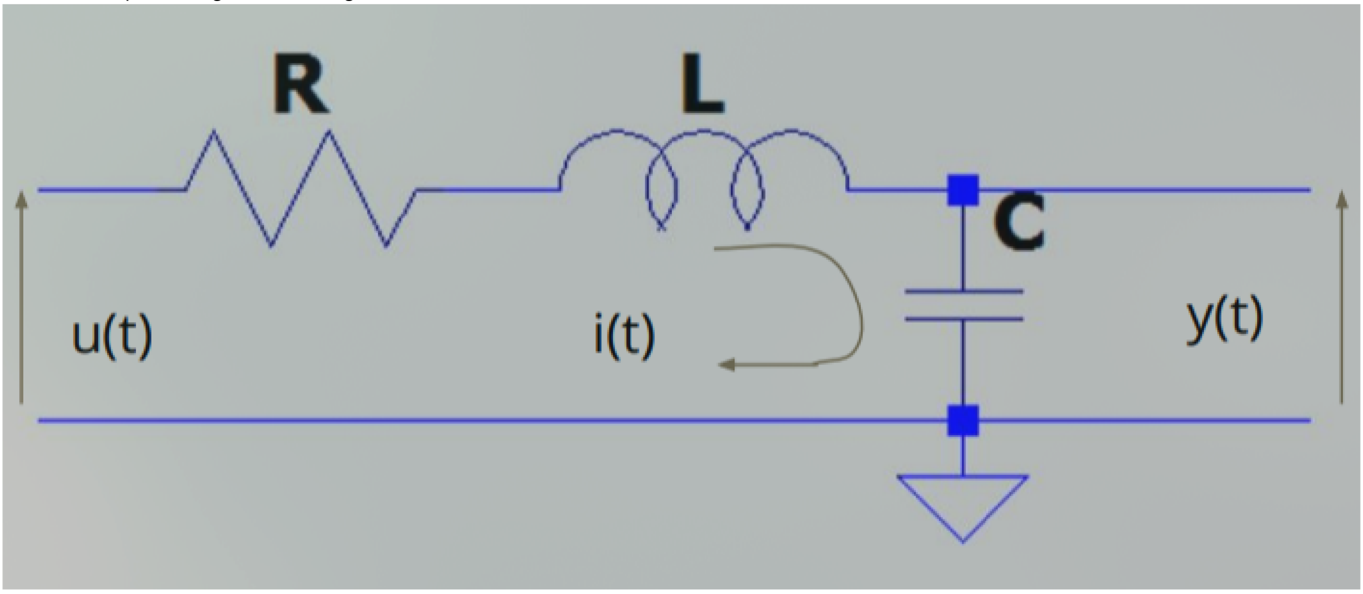
Task

Simulate output voltage of following RLC Circuit:



The equation for the statespace and block diagram can be worked out using Kirchoff's Voltage Law and Controller Conical Form from control system theory.

means $\frac{d}{dt}$

$\rightarrow \; y(LCs^2 + RCs + 1) = u$

$\therefore \; LC\ddot{y} + RC\dot{y} + y = u(t)$

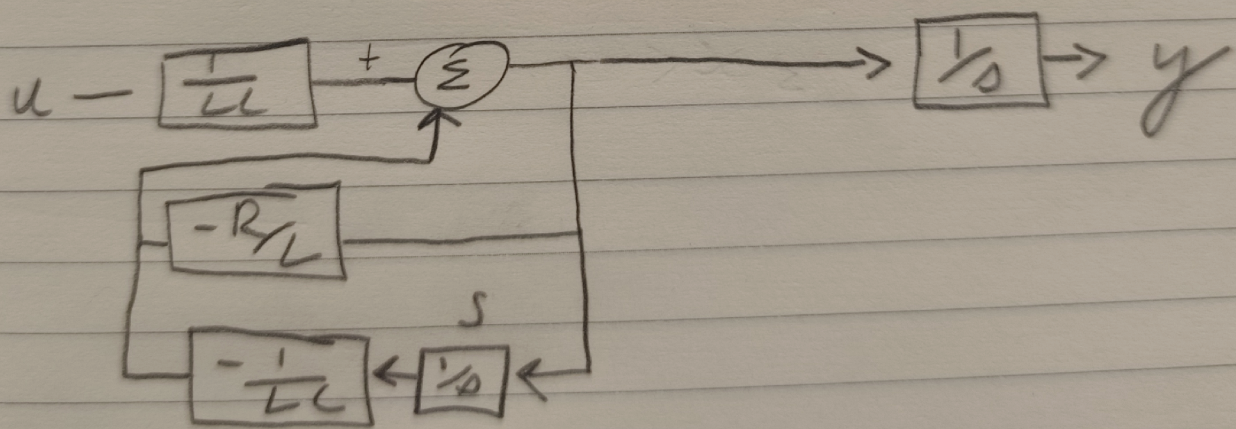
① Turn into controllable canonical form

$x_1 = y \qquad x_2 = \dot{y} = \dot{x}_1$

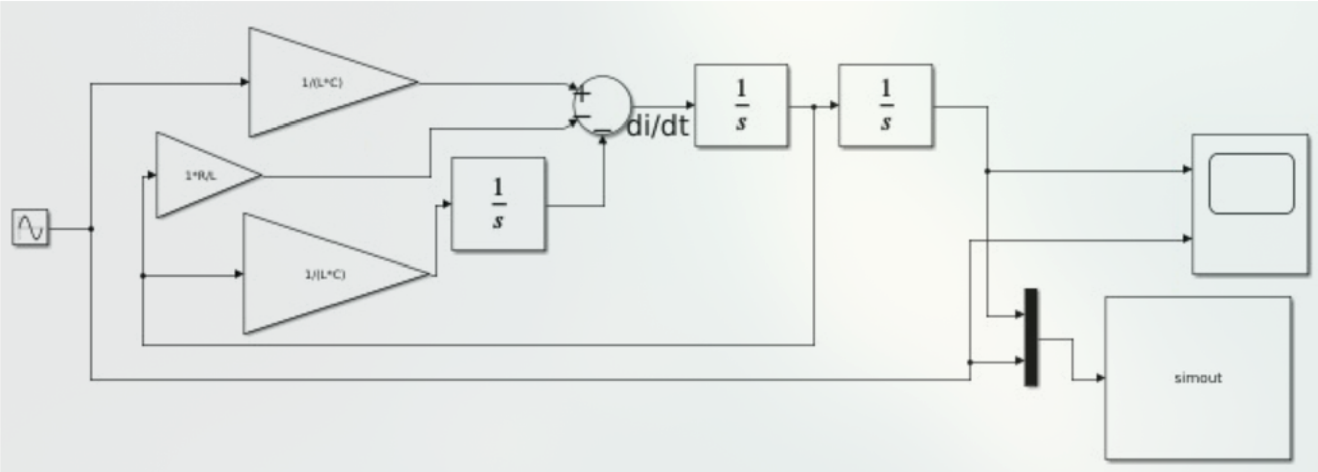
$\therefore \; y = \frac{1}{s} x_2$

② Write in cc form.

$$\dot{x}_2 = -\frac{1}{LC} x_1 - \frac{R}{L} x_2 + \frac{1}{LC} u$$



- Hence the simulation diagram becomes, where the multiplier block values from top to bottom are $\frac{1}{LC}$, $\frac{R}{L}$ and $\frac{1}{LC}$ respectively.



Time based experiments

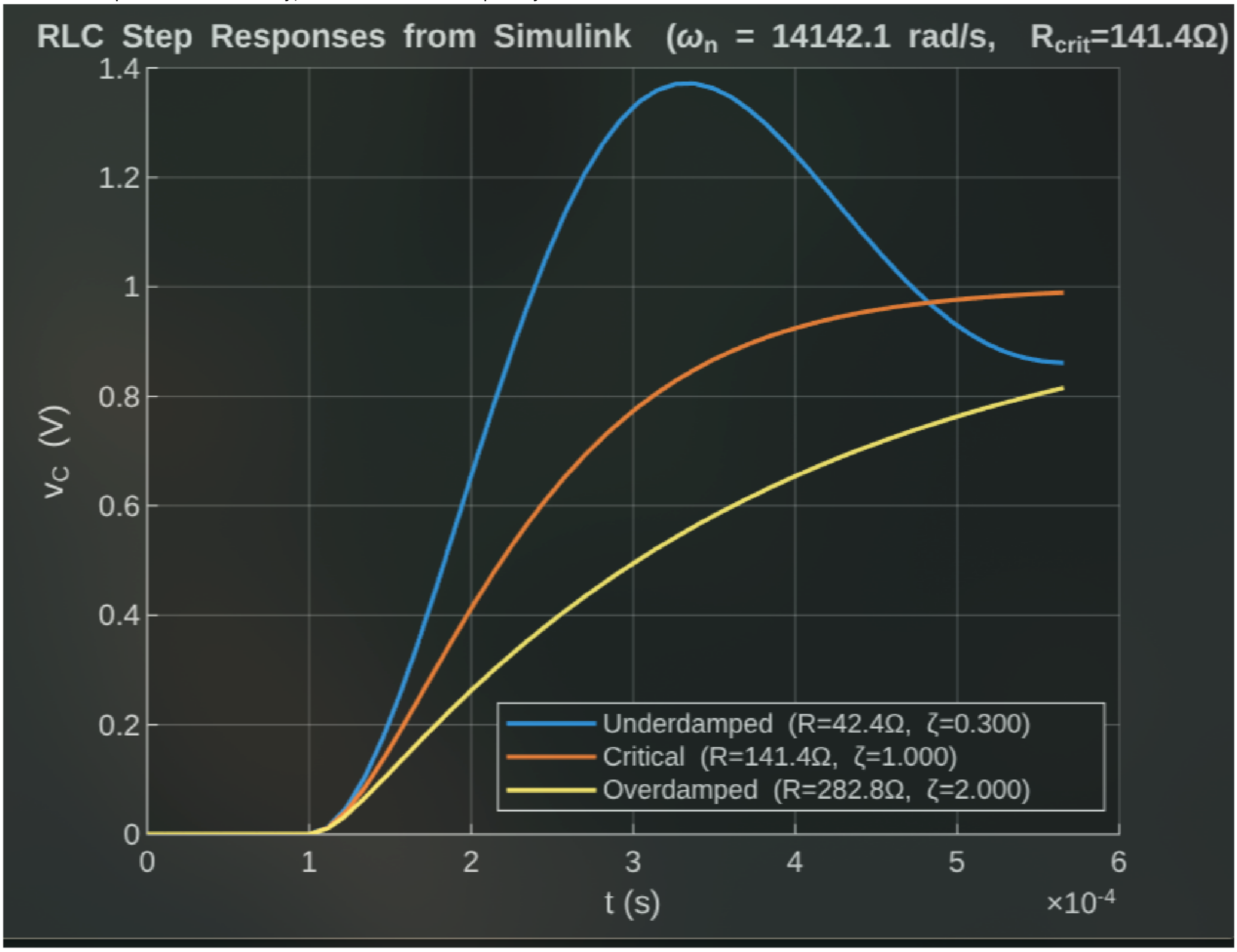
The transfer function can also be written:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{1}{LC\,s^2 + RC\,s + 1} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}.$$

Which means the natural frequency, damping ratio, quality factor, critical R can be defined

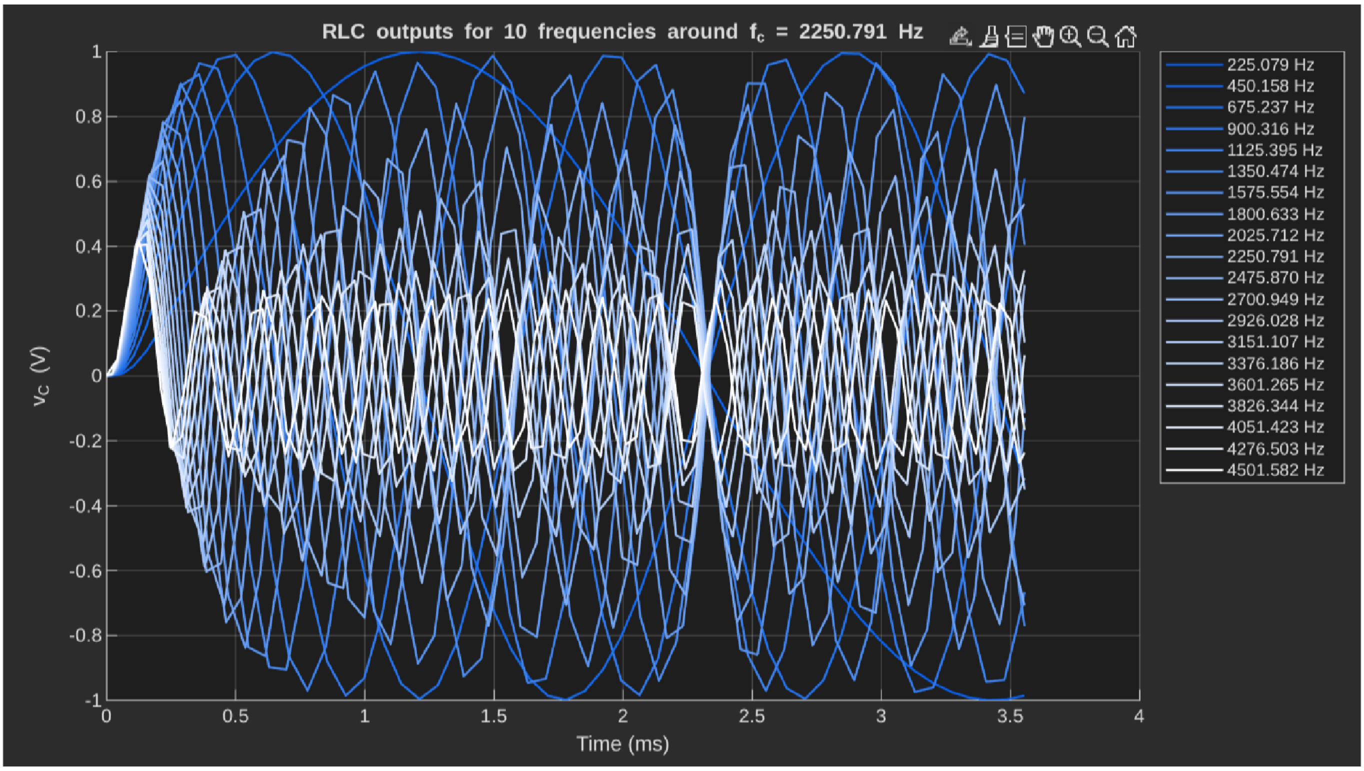
$$\omega_n = \frac{1}{\sqrt{LC}}, \qquad \zeta = \frac{R}{2}\sqrt{\frac{C}{L}}, \qquad Q = \frac{1}{2\zeta} = \frac{1}{R}\sqrt{\frac{L}{C}}, \qquad R_{\text{crit}} = 2\sqrt{\frac{L}{C}}.$$

Thus the response for a critically, under and over damped system can be simulated:



Frequency based Experiments

Since this circuit represents a low pass filter, we can calculate the cut-off frequency using the above and then see the effect of simulating at various input frequencies to see the effect of the filter. The simulation decreases the blue hue as the frequency increase and we can clearly see the time based output signal amplitude decrease, indicating the filter is working. It is also clear that the signal is suppressed as the frequency increase indicating that it is a low pass filter indeed.



See the attached code as reference.

```
clear      % ワークスペースからすべての変数を消去
close all  % すべてのFigureを消去
clc        % コマンド ウィンドウのクリア

%回路素子のパラメータ
R = 100;          %抵抗[Ω]
C = 1e-6;         %静電容量[C]
L = 5e-3;         %インダクタンス[H]
f_h = 2000;       % Hz
f_w = f_h * 2 *pi; % Rad/s
V = 1;            % Input V

% Landmarks
wn   = 1/sqrt(L*C);             % rad/s
Rcrit = 2*sqrt(L/C);            % critical damping (zeta = 1)

% Choose representative cases
R_under = 0.3*Rcrit;            % underdamped (zeta<1)
R_crit  = Rcrit;                % critically damped (zeta=1)
R_over  = 2.0*Rcrit;            % overdamped (zeta>1)
Rlist   = [R_under, R_crit, R_over];

%Simulinkの実行
Endtime = 1.0e-3;             %シミュレーション実行時間
mdl = 'Ex2_2_sim_1';       %ファイル名（拡張子なし）
open(mdl);               %Simulinkファイルを開く

tstop = 8/wn;    % enough time for settling
colors = lines(3);
figure('Name','Simulink step: damping cases'); hold on

labels = strings(1,3);
for k = 1:3
    R = Rlist(k);
    zeta = (R/2)*sqrt(C/L);

    si = Simulink.SimulationInput(mdl);
    si = si.setVariable('R',R);
    si = si.setVariable('L',L);
    si = si.setVariable('C',C);
    si = si.setVariable('V',V);
    si = si.setModelParameter('StopTime', num2str(tstop));

    out = sim(si);
    t = out.simout.time;
```

```
    y = out.simout.Data(:,1);   % assuming simout = [y u]

    plot(t, y, 'LineWidth',1.5, 'Color', colors(k,:));
    if k==1, tag="Underdamped"; elseif k==2, tag="Critical"; else, tag="Overdamped"; end
    labels(k) = sprintf("%s  (R=%.1fΩ,  ζ=%.3f)", tag, R, zeta);
end
grid on; xlabel('t (s)'); ylabel('v_C (V)');
title(sprintf('RLC Step Responses from Simulink  (\\omega_n = %.1f rad/s,  R_{crit}=%.1fΩ)', wn, Rcrit));
legend(labels, 'Location','best');
hold off;

% Frequency
R = 100;         % Ohm
L = 5e-3;        % H
C = 1e-6;        % F
V = 1;           % V

wn = 1/sqrt(L*C);
fn = wn/(2*pi);


steps = -9:10;
f_list = fn * (1 + 0.1*steps);         % Hz
omega_list = 2*pi*f_list;             % rad/s

mdl = 'Ex2_2_sim_2';
open(mdl);


Nc = 8;                              % show ~8 cycles at the cutoff frequency
tstop_common = Nc / fn;              % seconds

figure('Name','RLC outputs near cutoff'); hold on
baseBlue = [0, 0.35, 0.9];  % hue
white    = [1, 1, 1];
t = linspace(0,1,length(omega_list))';
colors = (1 - t).*baseBlue + t.*white;
leg = strings(1,numel(omega_list));

for k = 1:numel(omega_list)
    omega = omega_list(k);           % rad/s
    f     = f_list(k);               % Hz

    si = Simulink.SimulationInput(mdl);
    si = si.setVariable('R',R);
    si = si.setVariable('L',L);
    si = si.setVariable('C',C);
    si = si.setVariable('V',V);
    si = si.setVariable('omega',omega);
    si = si.setModelParameter('StopTime', num2str(tstop_common));

    out = sim(si);

    t = out.simout.time;
    y = out.simout.Data(:,1);

    plot(t*1e3, y, 'LineWidth', 1.2, 'Color', colors(k,:));
    leg(k) = sprintf('%.3f Hz', f);
end

grid on
xlabel('Time (ms)');
ylabel('v_C (V)');
title(sprintf('RLC outputs for 10 frequencies around f_c = %.3f Hz', fn));
legend(leg, 'Location','bestoutside');
```