Implement the following problem in a MATLAB *script* m-file.

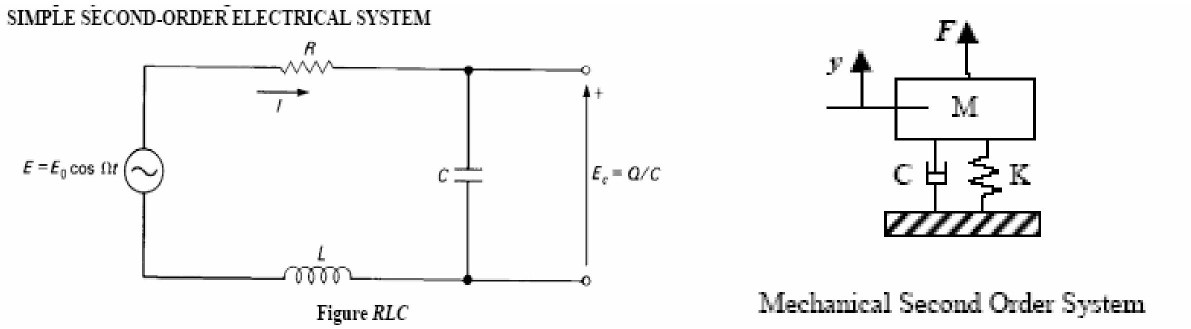

Figure 1: *Left*: Series RCL circuit. *Right*: Mass-spring-damper system. In this problem, at time $t = 0$, voltage V is applied (i.e., a switch is closed), or force F is suddenly applied. These input signals are held constant "for all time."

The following discussion is based on ***Figure 1***: a series RCL circuit, and a mass-spring-damper system. The two systems are mathematically equivalent to each other. We will assume that the system is ***underdamped***… which means when subjected to a constant input, it will oscillate about its final configuration for some time before reaching equilibrium.

The system is initially at "rest" (current is zero; velocity is zero) and there is no energy stored (no current through the inductor and no voltage across the capacitor; no extension in the spring). A step input is suddenly applied at time $t = 0$ (a switch is closed to apply DC voltage V_s; a constant force F is suddenly applied).

The response of an ***underdamped*** system to a step input is:

$$x(t) = u_0\left[1 - e^{-\zeta\omega_n t}(\cos\omega_d t - \gamma\sin\omega_d t)\right]$$

Response:	$x(t)$ (capacitor voltage v_c, displacement $y(t)$)
Size of step:	u_0 (applied voltage V_s, or static displacement, F/k)
Undamped Natural Frequency:	$\omega_n = \sqrt{\dfrac{1}{LC}}$ (electrical), $\omega_n = \sqrt{\dfrac{k}{m}}$ (mechanical)
Damping Ratio:	$\zeta = \dfrac{R}{2}\sqrt{\dfrac{\mathrm{C}}{\mathrm{L}}}$ (electrical), $\zeta = \dfrac{c}{2\sqrt{km}}$ (mechanical)
Damped Natural Frequency:	$\omega_d = \omega_n\sqrt{1-\zeta^2}$
Normalizing Constant:	$\gamma = \dfrac{\zeta}{1-\zeta^2}$

The equation for $x(t)$ is only valid for ***underdamped*** system. For an underdamped system: $\boxed{\zeta < 1}$
The function $x(t)$ is a constant (the response's final value), plus an exponentially decaying sinusoid. The system oscillates about it final value; the oscillation eventually dies down.

For **electrical systems with the components in series** (***Figure 1***), u_o is the applied DC voltage V_s, while $x(t)$ is the voltage measured across the capacitor, $v_c(t)$, so:

$$v_c(t) = V_s\left[1 - e^{-\zeta\omega_n t}(\cos\omega_d t - \gamma\sin\omega_d t)\right]$$

The final value of v_c it V_s. In your electrical engineering textbooks, you may have used the ***neper frequency***, or ***damping factor*** $\alpha = \zeta\omega_n$ in the exponential term. A circuit is underdamped when $\alpha < \omega_n$ or $\zeta < 1$.

For **mechanical systems** (***Figure 1***), u_o is the static deflection of the spring, F/k, the applied force divided by the spring constant. $x(t)$ is the spring displacement, $y(t)$, so:

$$y(t) = \frac{F}{k}\left[1 - e^{-\zeta\omega_n t}(\cos\omega_d t - \gamma\sin\omega_d t)\right]$$

The final value of y is F/k.

Systems that are ***overdamped*** ($\zeta > 1$ or $\alpha > \omega_n$) or ***critically damped*** ($\zeta = 1$ or $\alpha = \omega_n$) have different responses, beyond the scope of this exercise…. Perhaps it is for a future assignment.

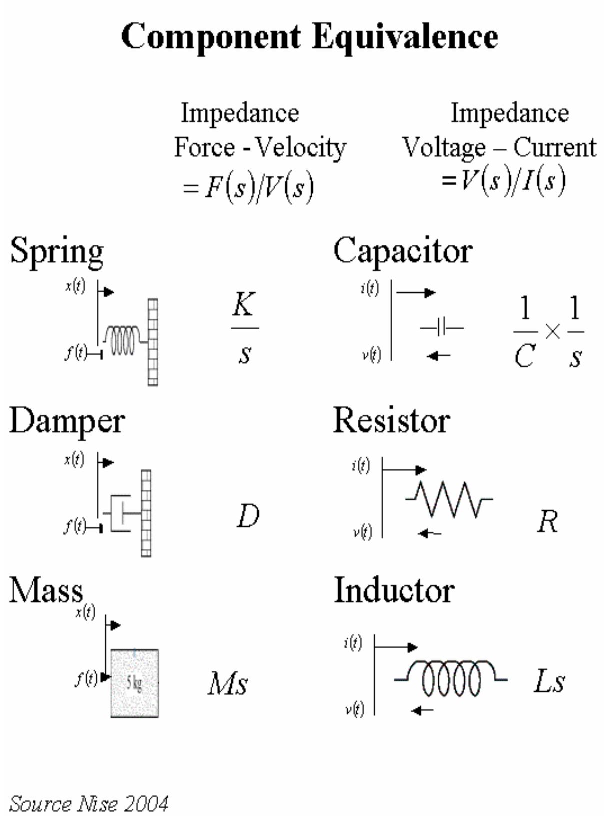

Figure 2. Component Equivalence.

EXERCISE:
Evaluate the system response for:

$\omega_n = 104$ rad/sec $\qquad \zeta = 0.3$

$u_0 = 1$ unit (Volts or Force/stiffness) $\qquad 0 \le t \le 0.4$ (sec).

Sample code is given below.

Physical question:
What improvements could be made to the system response? How would you change the parameters?

Programming Question:
Can you develop a code where you interactively input *R*, *C* and *L* (or *c*, *k* and *m*?)
Can you use `if-ifelse-end` statements to create a program that can analyze ***underdamped***, ***overdamped*** and ***critically damped*** systems for any (practical) combinations of *R*, *C* and *L* (or *D*, *K* and *M*).

```
%Engr 179 Allan Hancock College
%Sherman Wiggin
%Spring 2005
%Response of an underdamped second-order system to a step input

omegan=104  %The natural frequency
zeta = 0.3  %The dampening ratio
u0=1        %Input step size = unity
omegad=omegan*sqrt(1-zeta^2)%The natural damped frequency
gamm_a=zeta/sqrt(1-zeta^2)%non-dimensional parameter (simplifies
          notation)
t=linspace(0,0.4)   %Creates 100 data points 0 to 0.4
x=u0*(1-exp(-zeta*omegan*t).*(cos(omegad*t)-
          gamm_a*sin(omegad*t)))
plot(t,x)
```