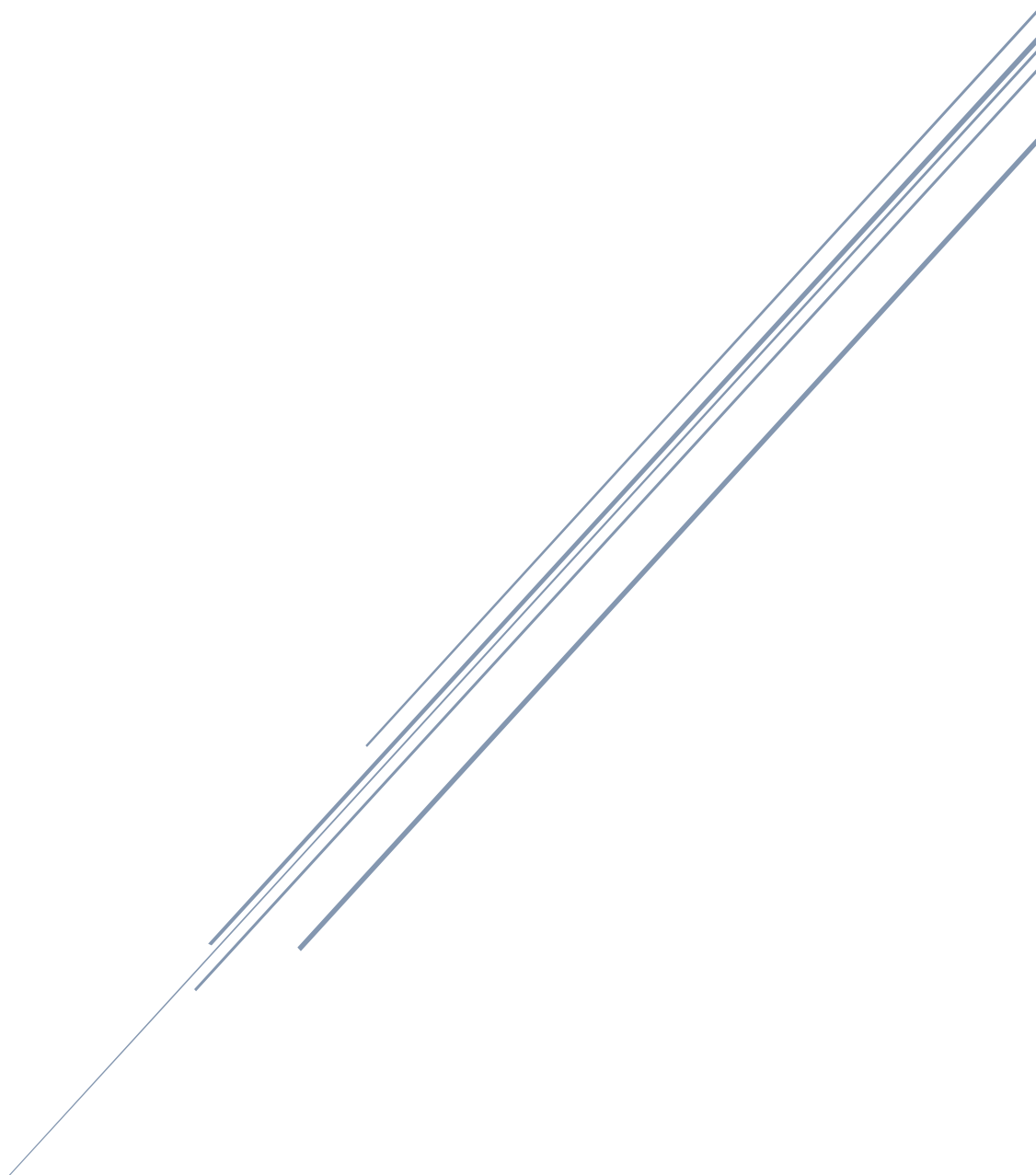


# SERIES RLC CIRCUIT SIMULATION & ANALYSIS

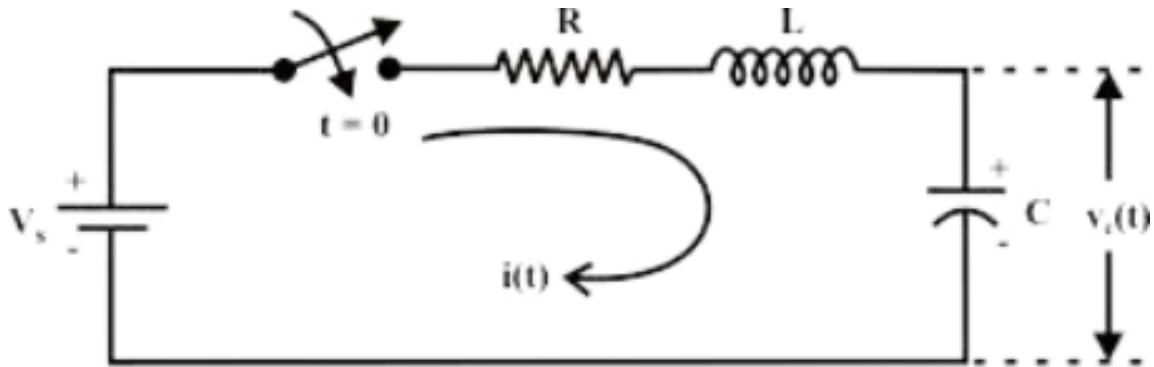


Nariman Ziaie  
December 2023

## فهرست مطالب

صفحه	عنوان
	[ تحلیل مدار RLC سری ]
2	۱-۱ محاسبات تئوری
4	۱-۲ نمایش بلوک دیاگرام و نمودار SFG
5	۱-۳ حالت های میرایی (Damping)
5	۱-۴ پاسخ پله مدار RLC
6	۱-۵ شبیه سازی در نرم افزار پروتئوس (Proteus)
8	۱-۶ شبیه سازی در نرم افزار مولتی سیم (Multisim)
10	۱-۷ شبیه سازی در نرم افزار متلب و سیمولینک (MATLAB & Simulink)
13	۱-۸ شبیه سازی به کمک کد پایتون (Python GUI)
15	[ فهرست منابع ]

## تحليل مدار RLC سري



۱-۱ محاسبات تئوری (  $t \geq t_0$  )

$$kvl: -v_i(t) + Ri(t) + l \frac{d_i(t)}{dt} + \frac{1}{c} \int_0^t i(\tau) dt = 0$$

$$\rightarrow v_i(t) = Ri(t) + l \frac{d_i(t)}{dt} + \frac{1}{c} \int_0^t i(\tau) dt$$

$$\rightarrow v_o(t) = \frac{1}{c} \int_0^t i(\tau) dt$$

**Derivative:**

$$\rightarrow Dv_i(t) = \left( D^2l + DR + \frac{1}{c} \right) \times i(t) = \left( -\frac{1}{l} (RI_0 + v_o) \right)$$

$$\rightarrow Dv_o(t) = \frac{1}{c} \times i(t)$$

**Laplace transform:**

$$\rightarrow CSv_i(s) = (LCS^2 + RCS + 1) \times I(s)$$

$$\rightarrow v_i(s) = RI(s) + SLI(s) + v_o(s)$$

$$* \rightarrow I(s) = \frac{CS}{LCS^2 + RCS + 1} = \frac{v_i(s) - v_o(s)}{R + SL}$$

$$\rightarrow Sv_o(s) = \frac{1}{C} \times I(s)$$

$$** \rightarrow v_o(s) = \frac{1}{CS} \times I(s)$$

$$\text{From } (*), (**): v_o(s) = \frac{1}{CS} \times \frac{CS}{LCS^2 + RCS + 1} \times v_i(s)$$

$$\rightarrow \frac{v_o(s)}{v_i(s)} = \frac{1}{LCS^2 + RCS + 1}$$

$$S = \left(-\frac{R}{2L} \pm \sqrt{\left(\frac{R}{2L}\right)^2 - \frac{1}{LC}}\right)$$

$$\rightarrow S = -\alpha^2 \pm \sqrt{a^2 + \omega_0^2}$$

$$\rightarrow \alpha = \frac{R}{2L}$$

$$\rightarrow \omega_0 = \frac{1}{\sqrt{LC}}$$

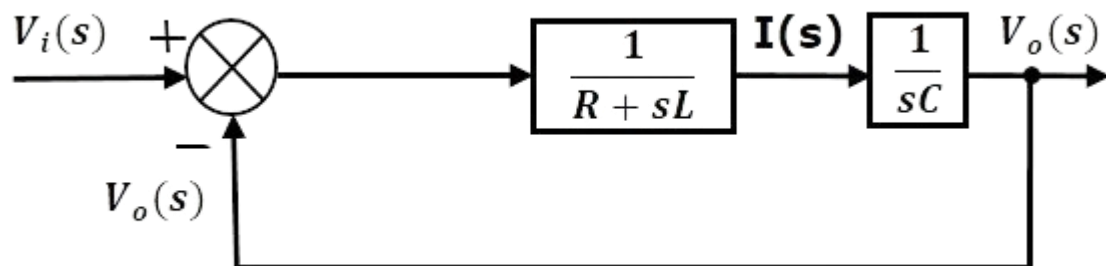
$$\rightarrow \tau = \frac{2L}{R}$$

**نکته:** از آنجایی که ریشه‌های  $s_1$  و  $s_2$  مربوط به پاسخ فرکانسی مدار هستند، به آن‌ها فرکانس‌های طبیعی (Natural frequencies) می‌گوییم که برحسب نپر بر ثانیه  $(N_p/s)$  اندازه‌گیری می‌شوند.

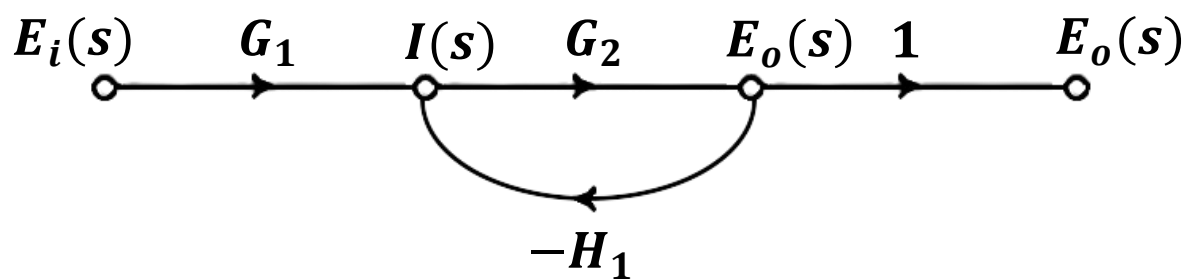
$\omega_0$  را فرکانس تشدید یا رزونانس (Resonant response) یا فرکانس طبیعی نامیرا می‌نامند و برحسب رادیان بر ثانیه  $(rad/s)$  بیان می‌کنند.

پارامتر  $\alpha$ ، فرکانس نپر (Neper frequency) یا ضریب میرایی (Damping factor) نامیده می‌شود و واحد آن، نپر بر ثانیه است.

**Block Diagram:**

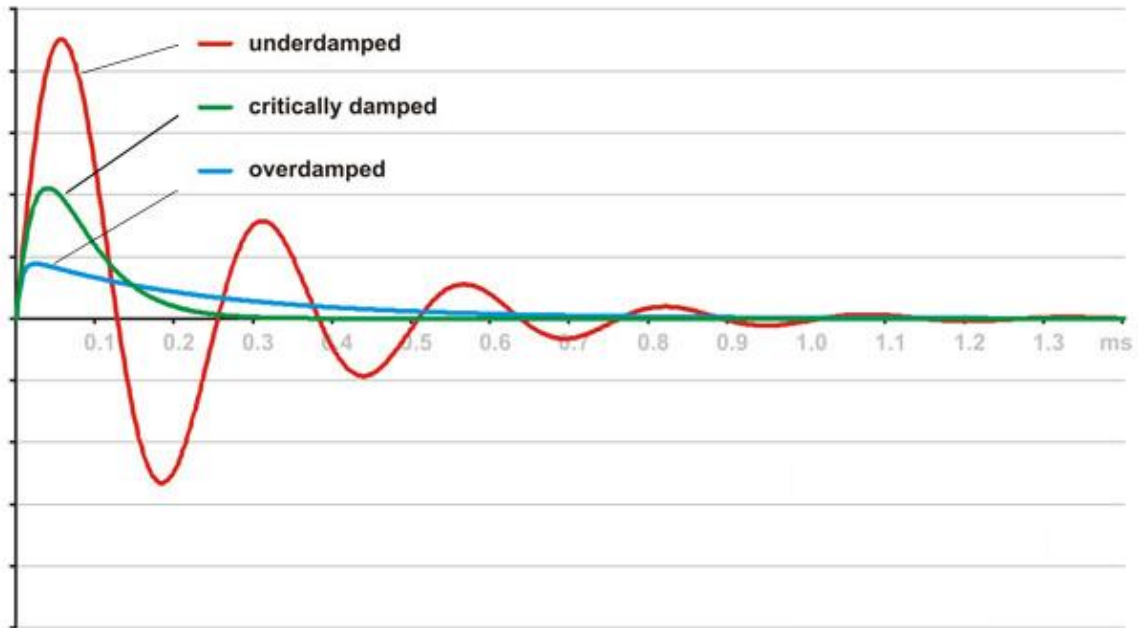


**Signal Flow Graph (SFG):**



### ۱-۳ حالت های میرایی

- اگر  $\alpha > \omega_0$  شود، پاسخ فرامیرا یا میرایی شدید (Overdamped) است.
- اگر  $\alpha = \omega_0$  شود، پاسخ میرایی بحرانی (Critically damped) است.
- اگر  $\alpha < \omega_0$  شود، پاسخ فرومیرا یا میرایی ضعیف (Underdamped) است.



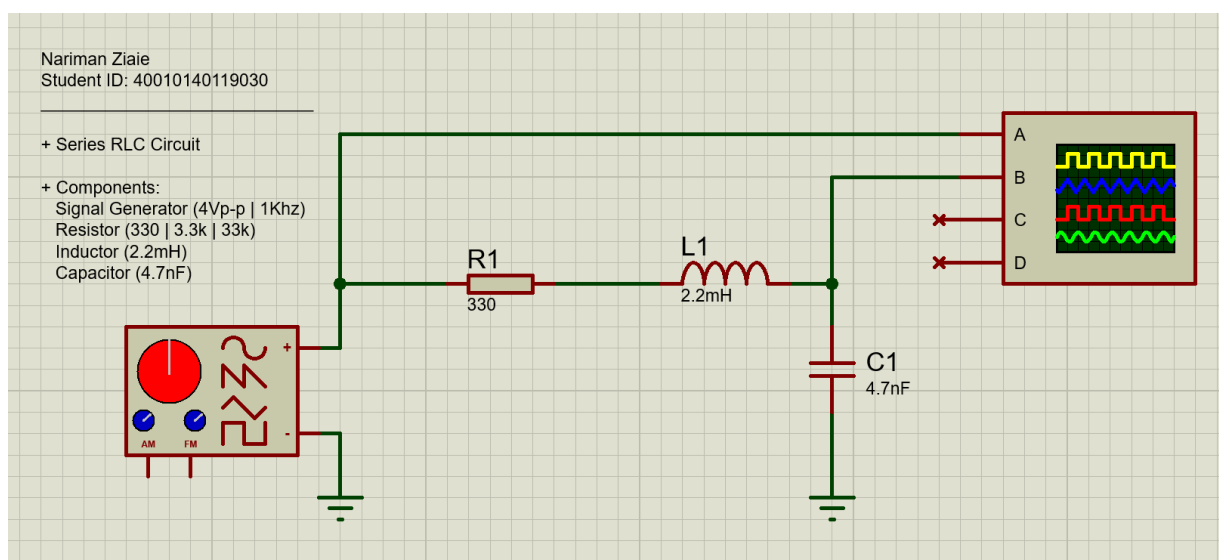
### ۱-۴ پاسخ پله مدار RLC

$v(t) = V_s + A_1 e^{s_1 t} + A_2 e^{s_2 t}$	(الف) فرامیرا
$v(t) = V_s + (A_1 + A_2 t) e^{-\alpha t}$	(ب) میرای بحرانی
$v(t) = V_s + (A_1 \cos \omega_d t + A_2 \sin \omega_d t) e^{-\alpha t}$	(ج) فرومیرا

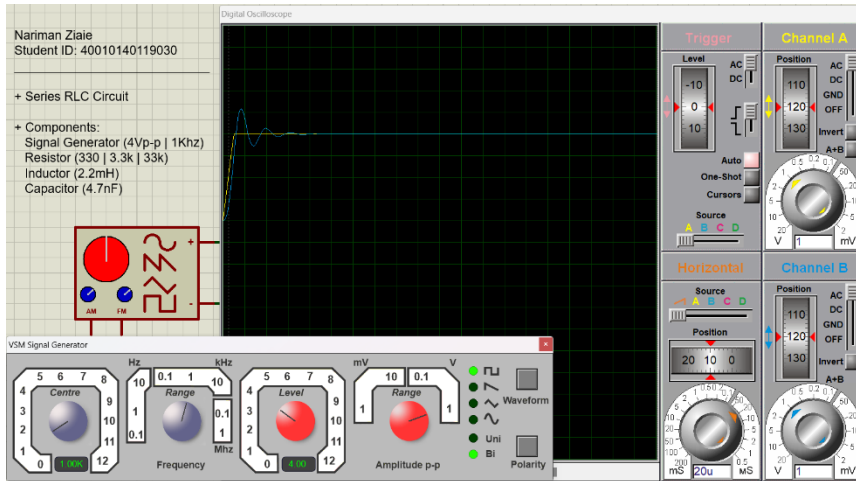
## ۱-۵ شبیه سازی در نرم افزار پروتئوس (Proteus)



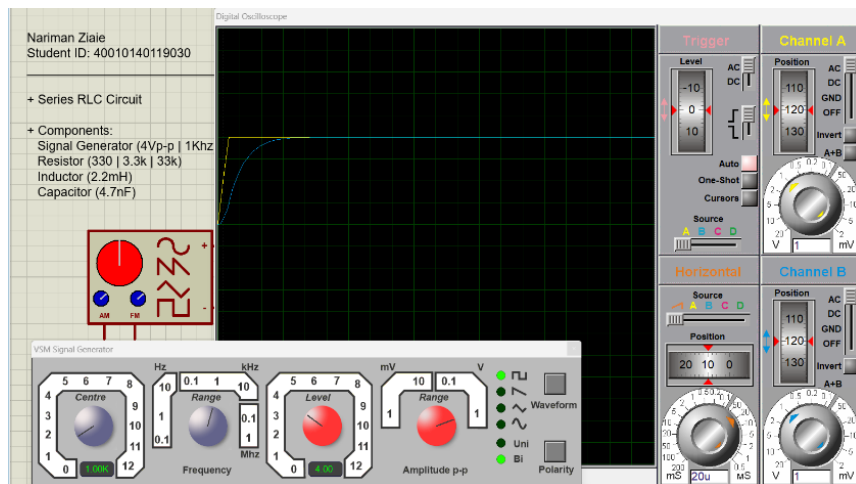
# PROTEUS



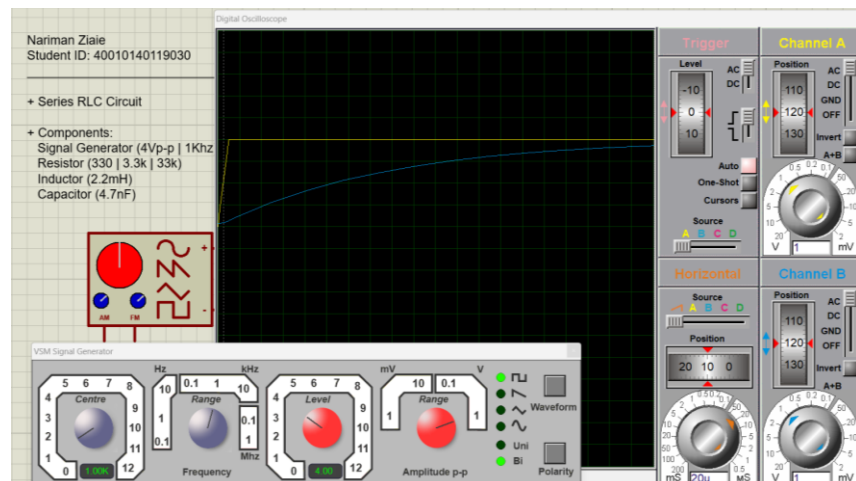
\*مدار بسته شده در محیط نرم افزار پروتئوس



✱ حالت میرایی ضعیف ( $R=330\Omega$ )



✱ حالت میرایی بحرانی ( $R=3.3k\Omega$ )



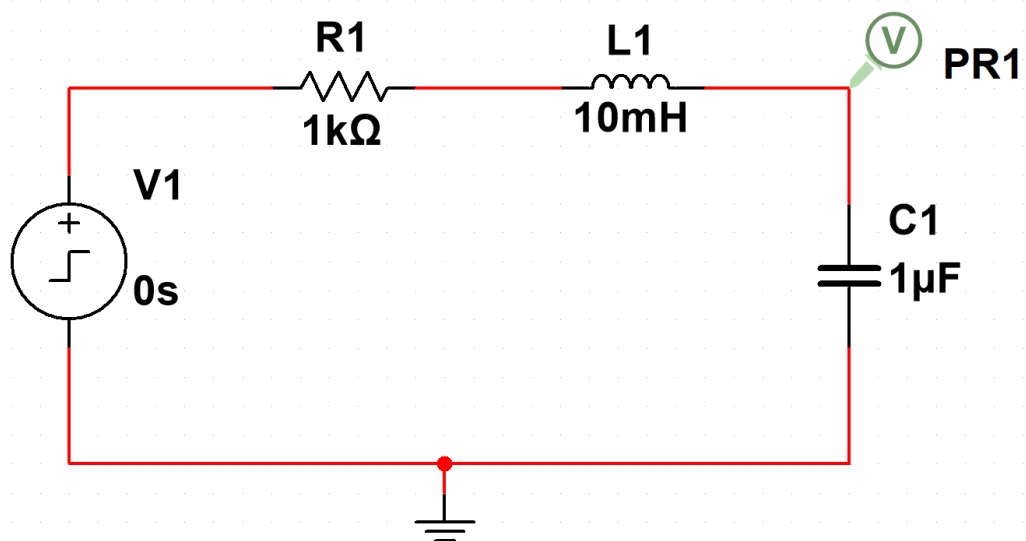
✱ حالت میرایی شدید ( $R=33k\Omega$ )



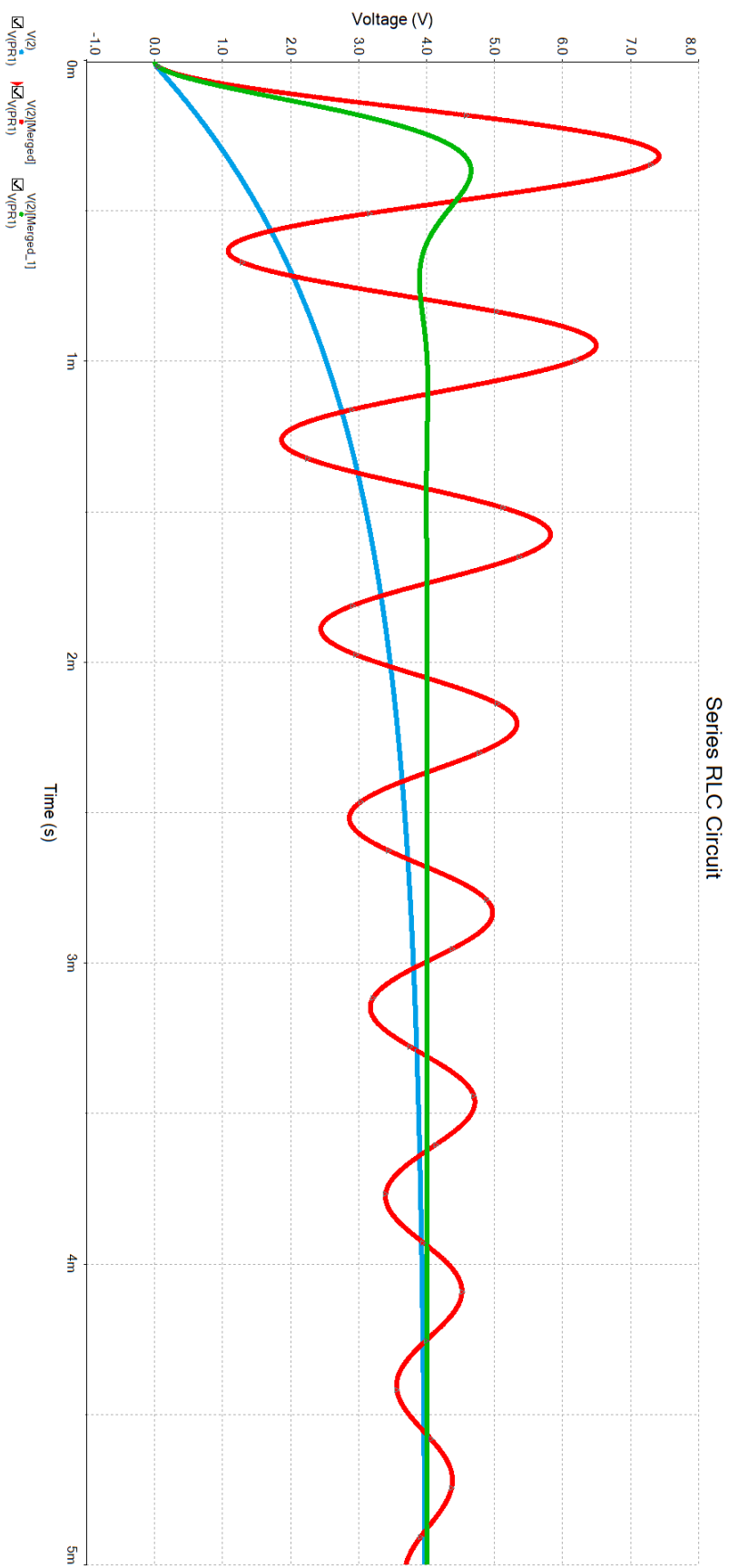
۱-۶ شبیه سازی در نرم افزار مولتی سیم (Multisim)



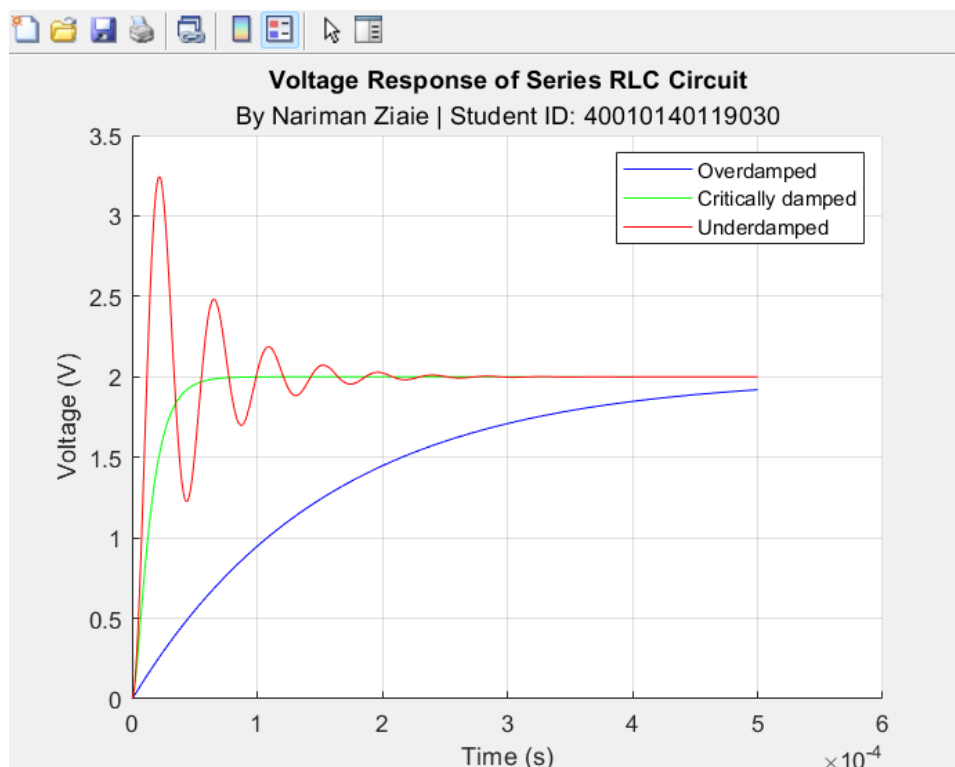
Multisim



\*مدار بسته شده در محیط نرم افزار مولتی سیم



\*\*\*حالت های میرایی ضعیف، بحرانی و شدید مدار RLC بسته شده به ازاء مقدار مقاومت ( $10\Omega$ ,  $100\Omega$ ,  $1k\Omega$ )

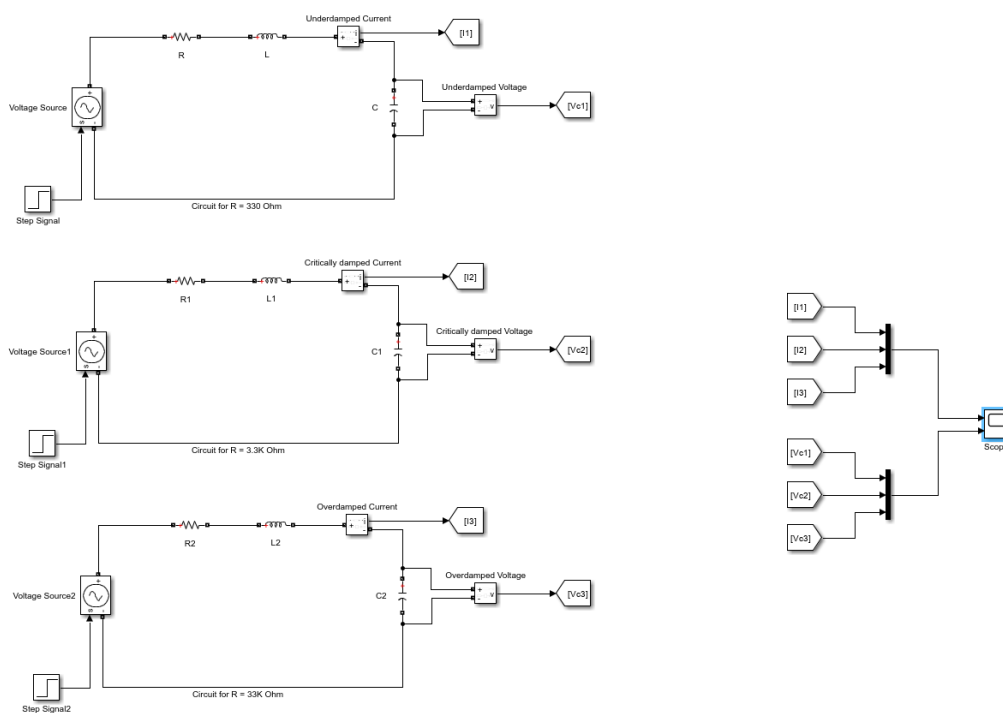


\*حالت های میرایی ضعیف، بحرانی و شدید مدار RLC سری

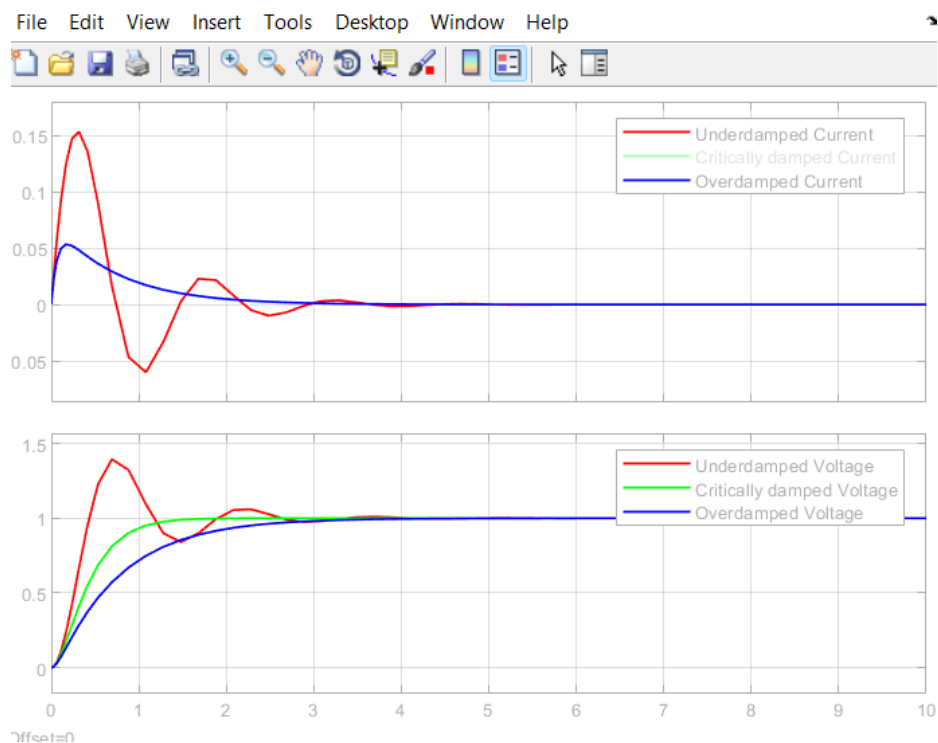
```

1. clc;
2. clear all;
3.
4. vb = 2; % Volts
5. r = 10; % Ohms
6. l = 10e-3; % Henries
7. c = 4.7e-9; % Farads
8. dt = 1e-6; % Second
9.
10. il_initial = 0;
11. vc_initial = 0;
12. n_iterations = 500;
13.
14. R_overdamped = 33e3;
15. R_critical = 3.3e3;
16. R_underdamped = 330;
17.
18. [time1, il, vl, V_overdamped] = simulate_rlc(vb, R_overdamped, l, c, dt, il_initial, vc_initial, n_iterations);
19. [time1, il, vl, V_critical] = simulate_rlc(vb, R_critical, l, c, dt, il_initial, vc_initial, n_iterations);
20. [time1, il, vl, V_underdamped] = simulate_rlc(vb, R_underdamped, l, c, dt, il_initial, vc_initial, n_iterations);
21.
22. figure;
23. title('Voltage Response of Series RLC Circuit');
24. subtitle('By Nariman Ziaie | Student ID: 40010140119030');
25. hold on;
26. plot(time1, V_overdamped, 'b');
27. hold on;
28. plot(time1, V_critical, 'g');
29. hold on;
30. plot(time1, V_underdamped, 'r');
31.
32. xlabel('Time (s)');
33. ylabel('Voltage (V)');
34. legend('Overdamped', 'Critically damped', 'Underdamped');
35.
36. grid on;
37. hold off;
38.
39. function [time1, il, vl, vc] = simulate_rlc(vb, r, l, c, dt, il_initial, vc_initial, n_iterations)
40.     time1 = zeros(n_iterations + 1, 1);
41.     il = zeros(n_iterations + 1, 1);
42.     vl = zeros(n_iterations + 1, 1);
43.     vc = zeros(n_iterations + 1, 1);
44.
45.     time1(1) = 0;
46.     il(1) = il_initial;
47.     vc(1) = vc_initial;
48.
49.     for n = 2:(n_iterations + 1)
50.         time1(n) = time1(n-1) + dt;
51.         l_dt = l / dt;
52.         c_dt = c * dt;
53.         il(n) = ((il(n-1) * (l_dt - r / 2 - dt / (2 * c))) + vb - vc(n-1)) / (l_dt + r / 2 + dt / (2 * c));
54.         vl(n) = ((il(n) - il(n-1)) / dt) * l;
55.         vc(n) = vc(n-1) + (((il(n) + il(n-1)) / 2) * dt) / c;
56.     end
57. end

```

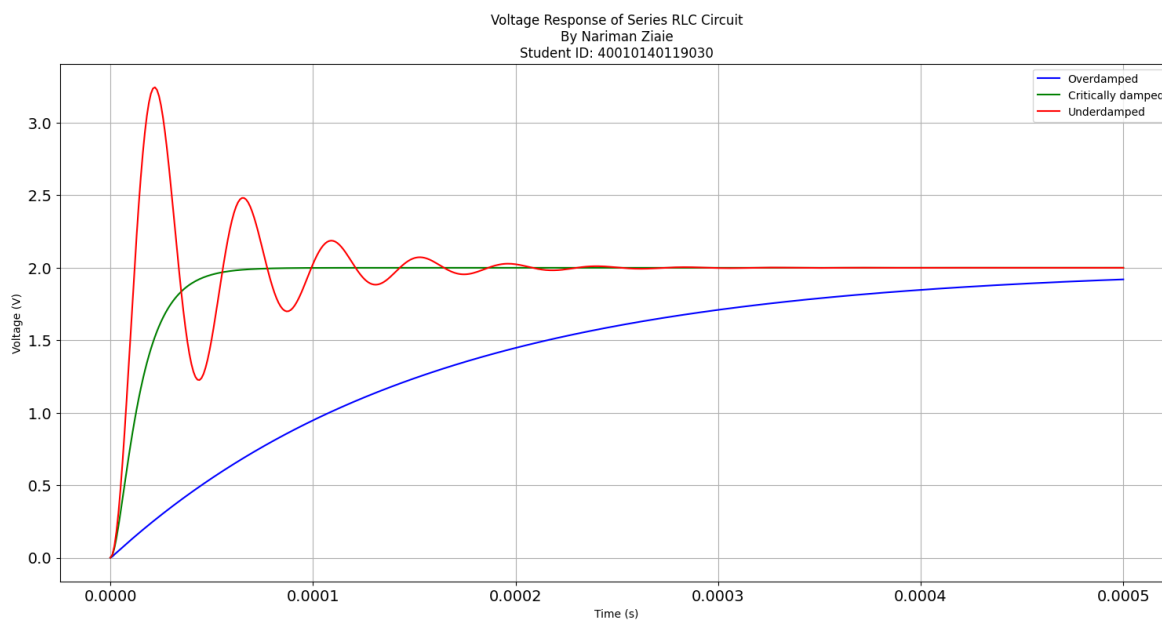


※مدار شبیه سازی شده در نرم افزار سیمولینک



※حالت های میرایی ضعیف، بحرانی و شدید مدار RLC سری

## ۸-۱ شبیه سازی به کمک کد پایتون (Python GUI)



\*حالت های میرایی ضعیف، بحرانی و شدید مدار RLC سری

```

1. import numpy as np
2. import matplotlib.pyplot as plt
3.
4. # Define circuit parameters and input voltage
5. vb = 2 # Volts
6. r = 10 # Ohms
7. l = 10e-3 # Henries
8. c = 4.7e-9 # Farads
9. dt = 1e-6 # Second
10.
11. # Initial conditions
12. i1_initial = 0
13. vc_initial = 0
14.
15. # Define damping resistance
16. R_overdamped = 33e3 # Overdamped
17. R_critical = 3.3e3 # Critically damped
18. R_underdamped = 330 # Underdamped
19.
20. # Define simulation function
21. def simulate_rlc(vb, r, l, c, dt, i1_initial, vc_initial, n_iterations):
22.     time1 = np.zeros(n_iterations + 1)
23.     i1 = np.zeros(n_iterations + 1)
24.     v1 = np.zeros(n_iterations + 1)
25.     vc = np.zeros(n_iterations + 1)
26.
27.     time1[0] = 0
28.     i1[0] = i1_initial
29.     vc[0] = vc_initial
30.
31.     # Calculate values for each iteration
32.     for n in range(1, n_iterations + 1):
33.         time1[n] = time1[n-1] + dt
34.         l_dt = l / dt
35.         c_dt = c * dt
36.         i1[n] = ((i1[n-1] * (l_dt - r / 2 - dt / (2 * c))) + vb - vc[n-1]) / (l_dt + r / 2 + dt / (2 * c))
37.         v1[n] = ((i1[n] - i1[n-1]) / dt) * l
38.         vc[n] = vc[n-1] + (((i1[n] + i1[n-1]) / 2) * dt) / c
39.
40.     return time1, i1, v1, vc
41.
42. # Simulate the circuit
43. n_iterations = 500
44. time1, i1, v1, vc = simulate_rlc(vb, r, l, c, dt, i1_initial, vc_initial, n_iterations)
45.
46. time1, i1, v1, V_overdamped = simulate_rlc(vb, R_overdamped, l, c, dt, i1_initial, vc_initial, n_iterations)
47. time1, i1, v1, V_critical = simulate_rlc(vb, R_critical, l, c, dt, i1_initial, vc_initial, n_iterations)
48. time1, i1, v1, V_underdamped = simulate_rlc(vb, R_underdamped, l, c, dt, i1_initial, vc_initial, n_iterations)
49.
50. # Plot results
51. plt.figure(figsize=(12, 8))
52. plt.title("Voltage Response of Series RLC Circuit\nBy Nariman Ziaie\nStudent ID: 40010140119030")
53.
54. # Overdamped
55. plt.plot(time1, V_overdamped, label="Overdamped", color="blue")
56.
57. # Critically damped
58. plt.plot(time1, V_critical, label="Critically damped", color="green")
59.
60. # Underdamped
61. plt.plot(time1, V_underdamped, label="Underdamped", color="red")
62.
63. # Label the axes
64. plt.xlabel("Time (s)")
65. plt.ylabel("Voltage (V)")
66.
67. plt.legend(loc="upper right")
68.
69. # Increase the font size of the labels
70. plt.xticks(fontsize=14)
71. plt.yticks(fontsize=14)
72.
73. # Show the plot
74. plt.grid(True)
75. plt.tight_layout()
76. plt.show()

```

## فهرست منابع

- کتاب تحلیل مهندسی مدار - ویلیام هیت - ترجمه محمود دیانی
- کتاب سیستم های کنترل خطی - خاکی صدیق
- تحلیل مدار RLC سری - فرادرس
- MATLAB documentation
- Python documentation
- Stack overflow
- GitHub

---

برای مشاهده فایل های پروژه روی تصویر زیر کلیک کنید یا آن را اسکن نمایید.

