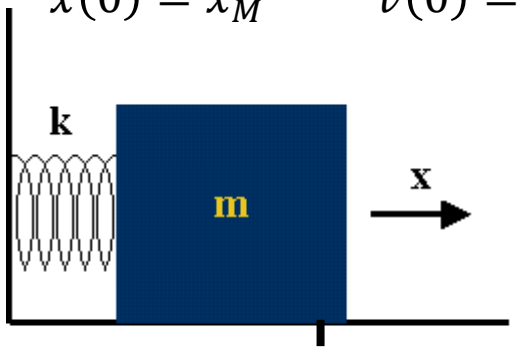
MEDICIONES DE
Circuito RLC
Respuesta transitoria
Clase 11

LABORATORIO 3
2do cuatrimestre 2020

Oscilador armónico simple

Resorte

$x(0) = x_M \qquad v(0) = 0$



x=0, posición natural del resorte

$$m\,\ddot{x} = -kx$$

$$x = x_M \cos(\omega_0 t)$$

Frecuencia natural de oscilación

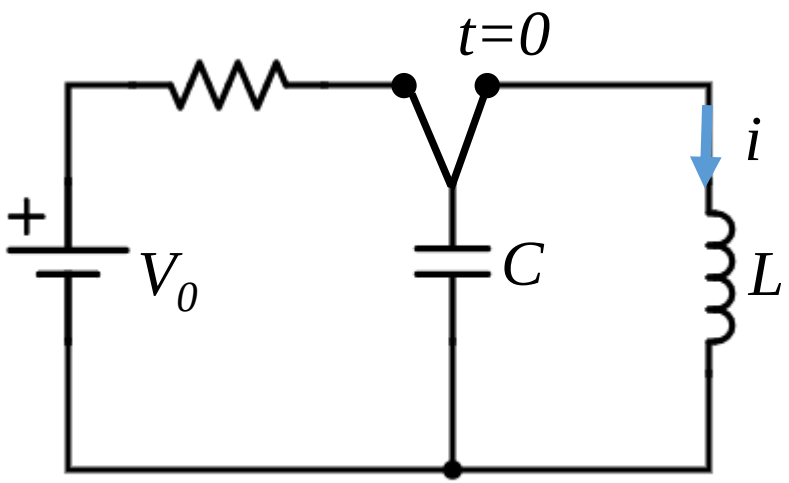
$$\omega_0 = \sqrt{k/m}$$

$$E_{mec} = E_{cin} + E_{res} = cte.$$

$$E_{mec} = \frac{1}{2}mv^2 + \frac{1}{2}kx^2 = cte.$$

Oscilador armónico simple

Circuito LC



$i(t=0)=0$ $\qquad$ $V_C(t=0)=V_0$

$$L\frac{di}{dt}+V_C=0 \Rightarrow L\ddot{q}+\frac{q}{C}=0$$

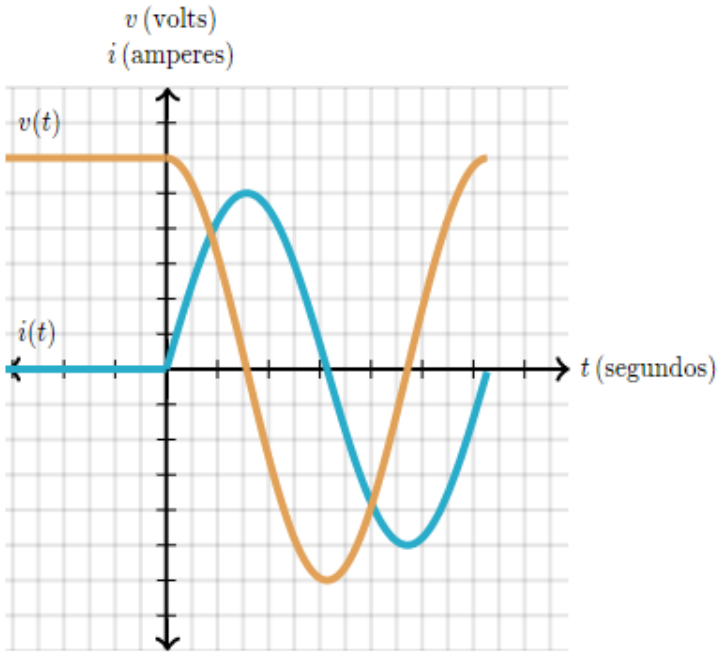
$$q=q_0\cos(\omega_0 t+\delta)$$

Frecuencia natural de oscilación

$$\omega_0=\sqrt{1/_{LC}}$$

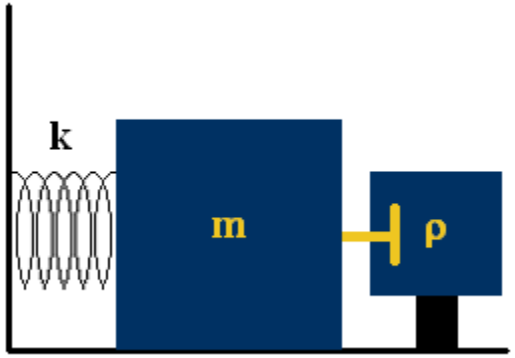
$$E=E_C+E_L=cte.$$

$$E=\frac{1}{2}\frac{q^2}{C}+\frac{1}{2}Li^2=cte.$$



Oscilador armónico amortiguado

Resorte amortiguado



$$m\ddot{x} = -kx - \rho\dot{x}$$

$$\ddot{x} + \frac{\rho}{m}\dot{x} + \frac{k}{m}x = \ddot{x} + 2\gamma\dot{x} + \omega_0^2 x = 0$$

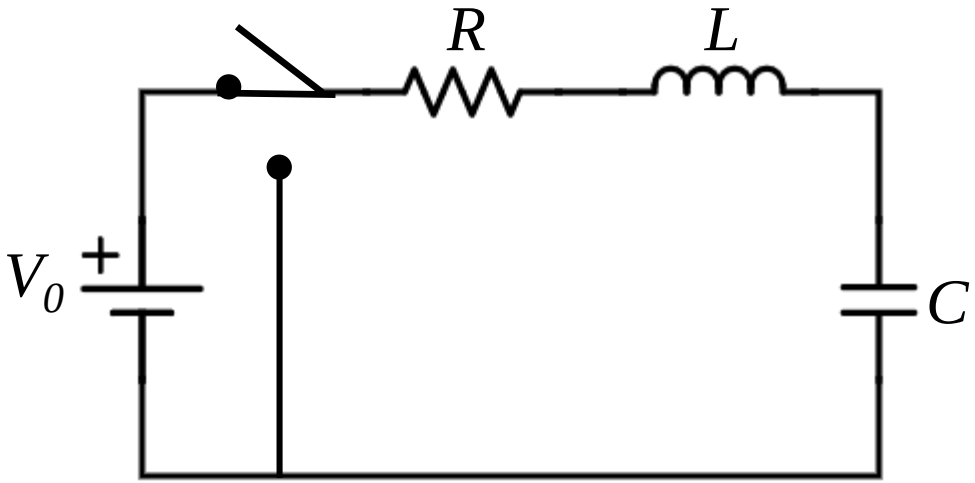
$$x = A_1 e^{\lambda_1 t} + A_2 e^{\lambda_2 t} \qquad \lambda_{1,2} = -\gamma \pm \sqrt{\gamma^2 - \omega_0^2}$$

Casos de amortiguamiento

$\omega_0^2 > \gamma^2$	$\omega_0^2 = \gamma^2$	$\omega_0^2 < \gamma^2$
subamortiguado	Amortiguado crítico	sobremortiguado

Oscilador armónico amortiguado

Circuito RLC serie – transitorio eléctrico



$$\gamma = \frac{R}{2L} \qquad \omega_0 = \frac{1}{\sqrt{LC}}$$

$$V_0 = Ri + L\frac{di}{dt} + \frac{1}{C}q(t)$$

$$V_0 = R\dot{q} + L\ddot{q} + \frac{1}{C}q$$

$$\frac{V_0}{L} = \ddot{q} + 2\frac{R}{2L}\dot{q} + \frac{1}{LC}q$$

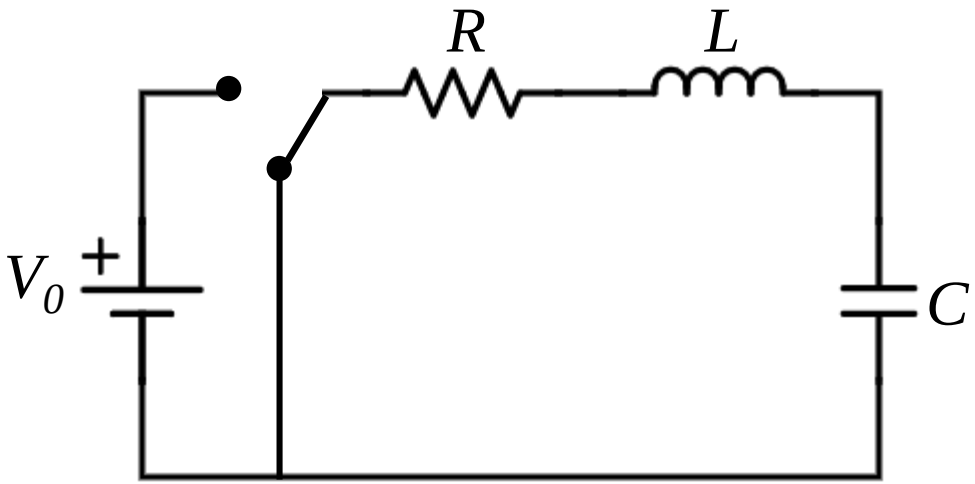
Condiciones iniciales

$$q(t=0) = 0 \qquad \dot{q}(t=0) = i(t=0) = 0$$

Solución particular

$$q_p(t) = V_0 C$$

Circuito RLC serie – transitorio eléctrico



$$0 = Ri + L\frac{di}{dt} + \frac{1}{C}q(t)$$

$$0 = R\dot{q} + L\ddot{q} + \frac{1}{C}q$$

$$0 = \ddot{q} + 2\frac{R}{2L}\dot{q} + \frac{1}{LC}q$$

$$\gamma = \frac{R}{2L} \qquad \omega_0 = \frac{1}{\sqrt{LC}}$$

Condiciones iniciales

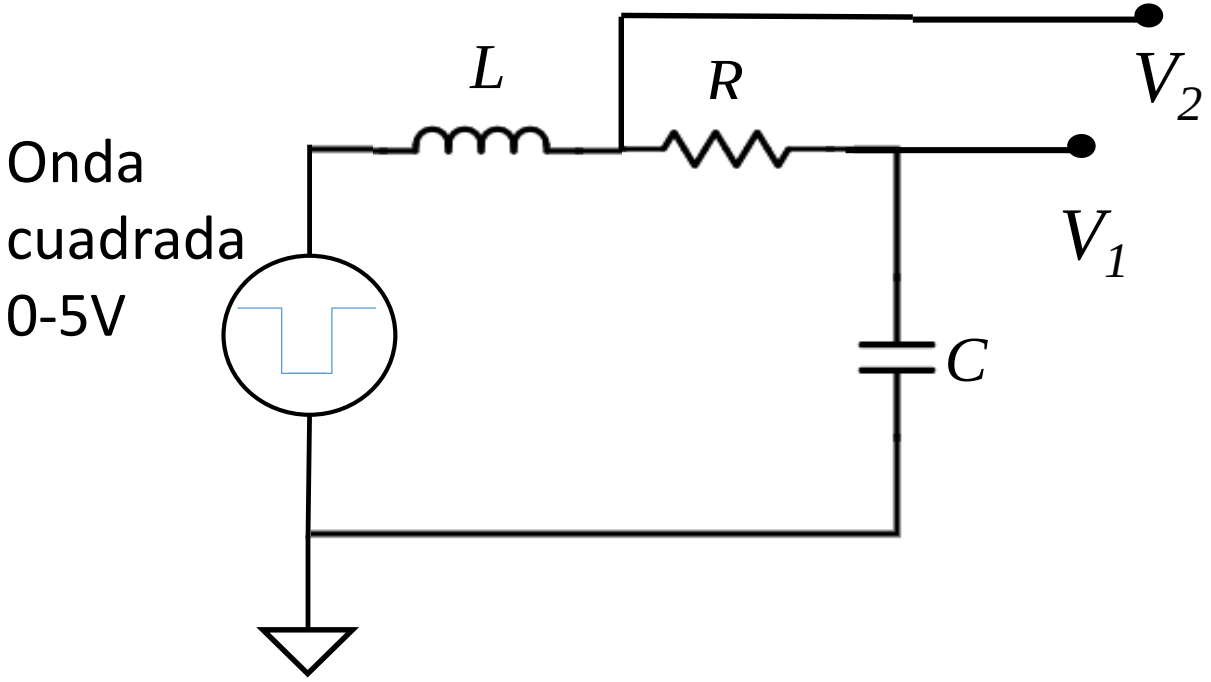
$$q(t=0) = V_0 C \qquad \dot{q}(t=0) = i(t=0) = 0$$

$$\omega_0^2 > \gamma^2 \qquad \omega_0^2 = \gamma^2 \qquad \omega_0^2 < \gamma^2$$

subamortiguado | Amortiguado crítico | sobremortiguado

Experimento propuesto

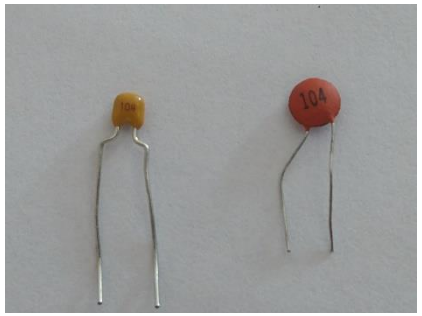
Circuito RLC serie – transitorio eléctrico



$$V_C = V_1$$

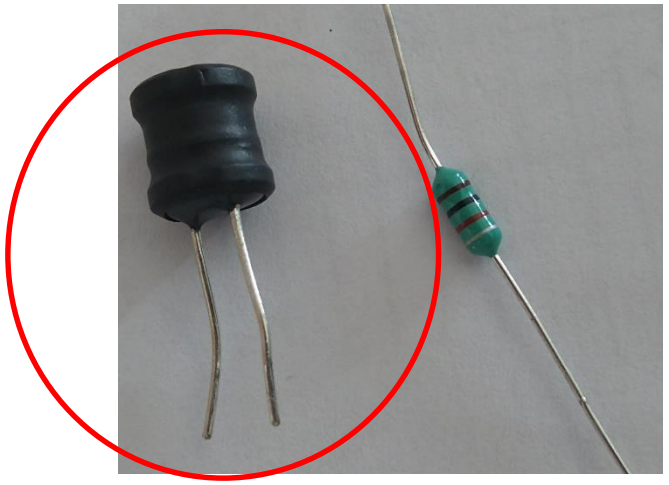
$$I = \frac{V_2 - V_1}{R}$$

Experimento Circuito RLC serie
Componentes del circuito



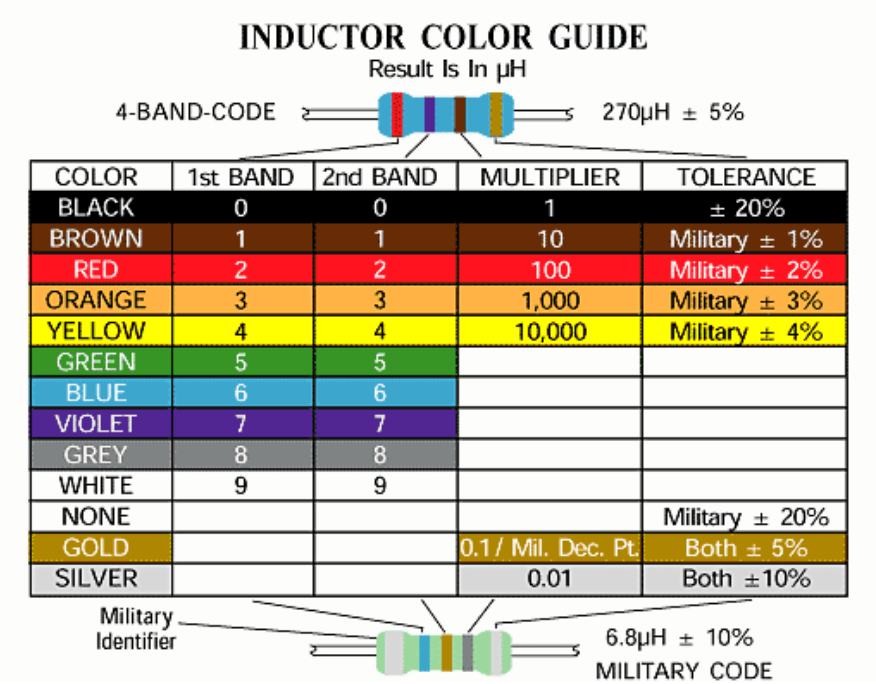
Capacitor cerámico

104 ➔ $10*10^4$pF= 100nF



Inductancias

Capacitor de poliéster



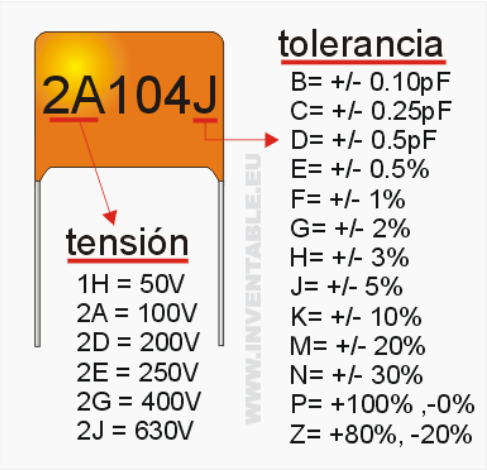
INDUCTOR COLOR GUIDE

Result Is In µH

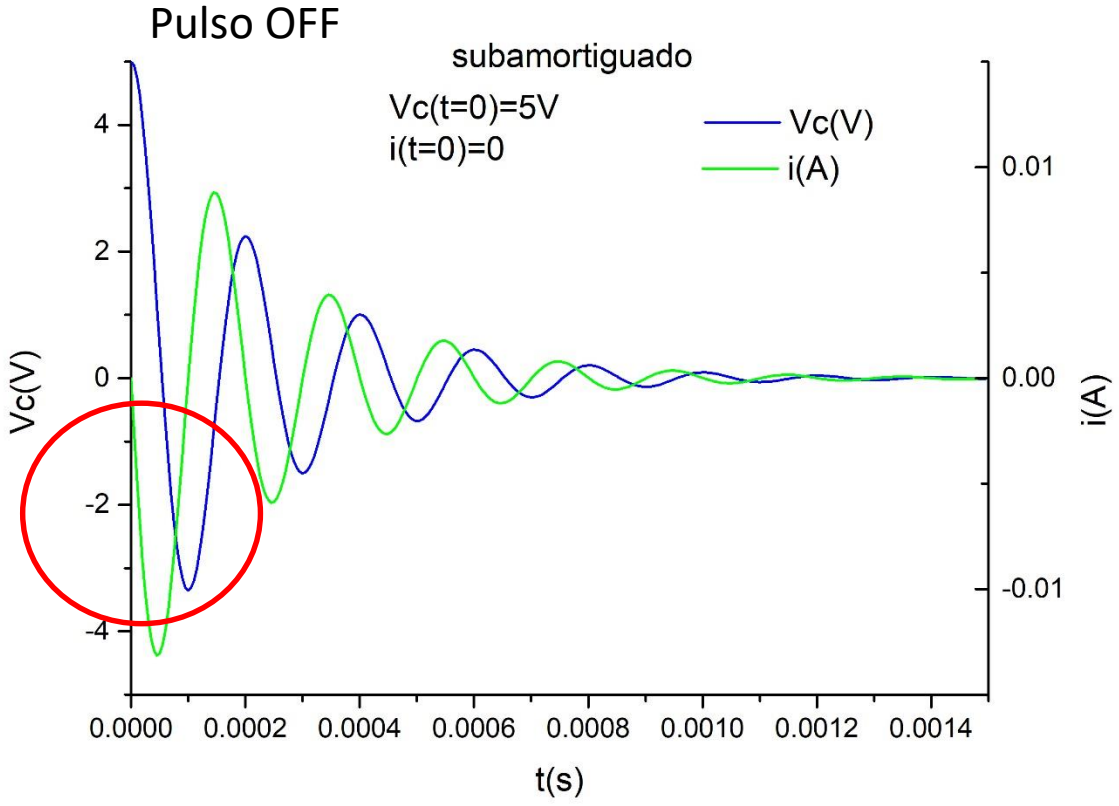
4-BAND-CODE 270µH ± 5%

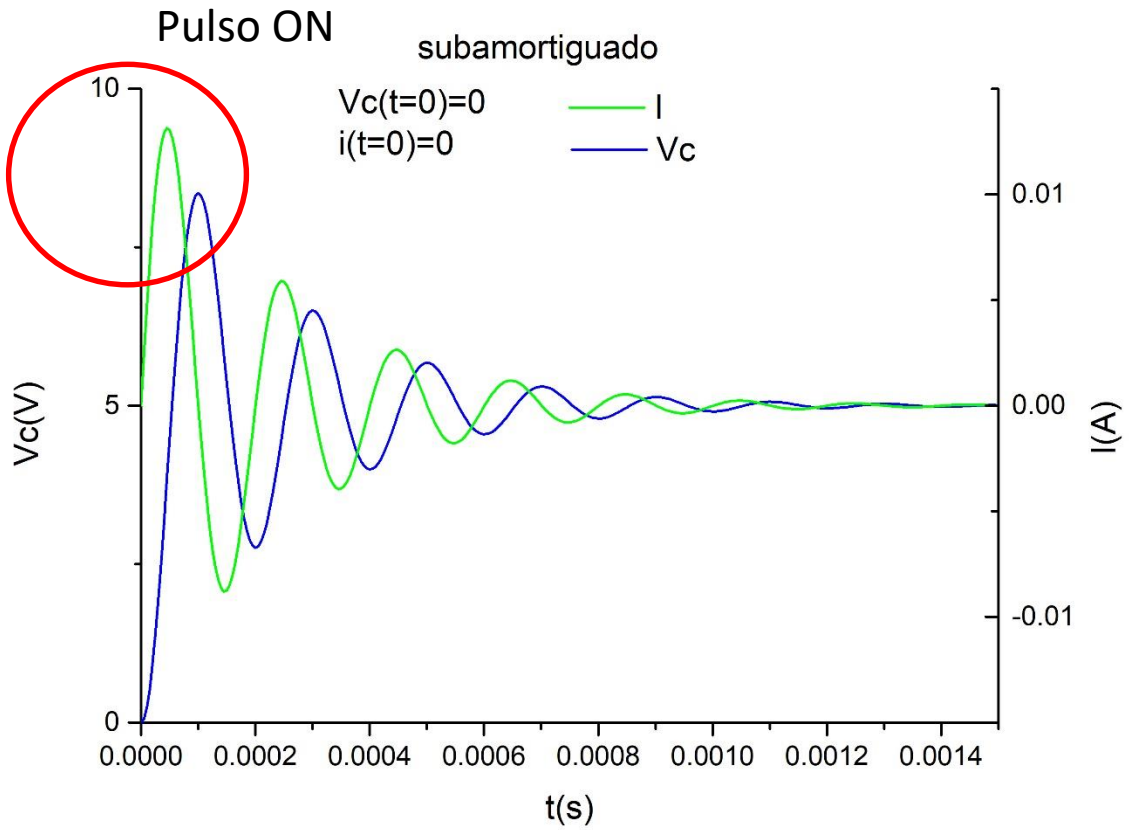
COLOR	1st BAND	2nd BAND	MULTIPLIER	TOLERANCE
BLACK	0	0	1	± 20%
BROWN	1	1	10	Military ± 1%
RED	2	2	100	Military ± 2%
ORANGE	3	3	1,000	Military ± 3%
YELLOW	4	4	10,000	Military ± 4%
GREEN	5	5		
BLUE	6	6		
VIOLET	7	7		
GREY	8	8		
WHITE	9	9		
NONE				Military ± 20%
GOLD			0.1 / Mil. Dec. Pt.	Both ± 5%
SILVER			0.01	Both ±10%

Military Identifier 6.8µH ± 10% MILITARY CODE



Circuito RLC serie - Transitorio Subamortiguado





$$\gamma = \frac{R}{2L}$$

$$Vc(t) = V_0 e^{-\gamma t}(\cos \omega t + \frac{\gamma}{\omega} \sin \omega t)$$

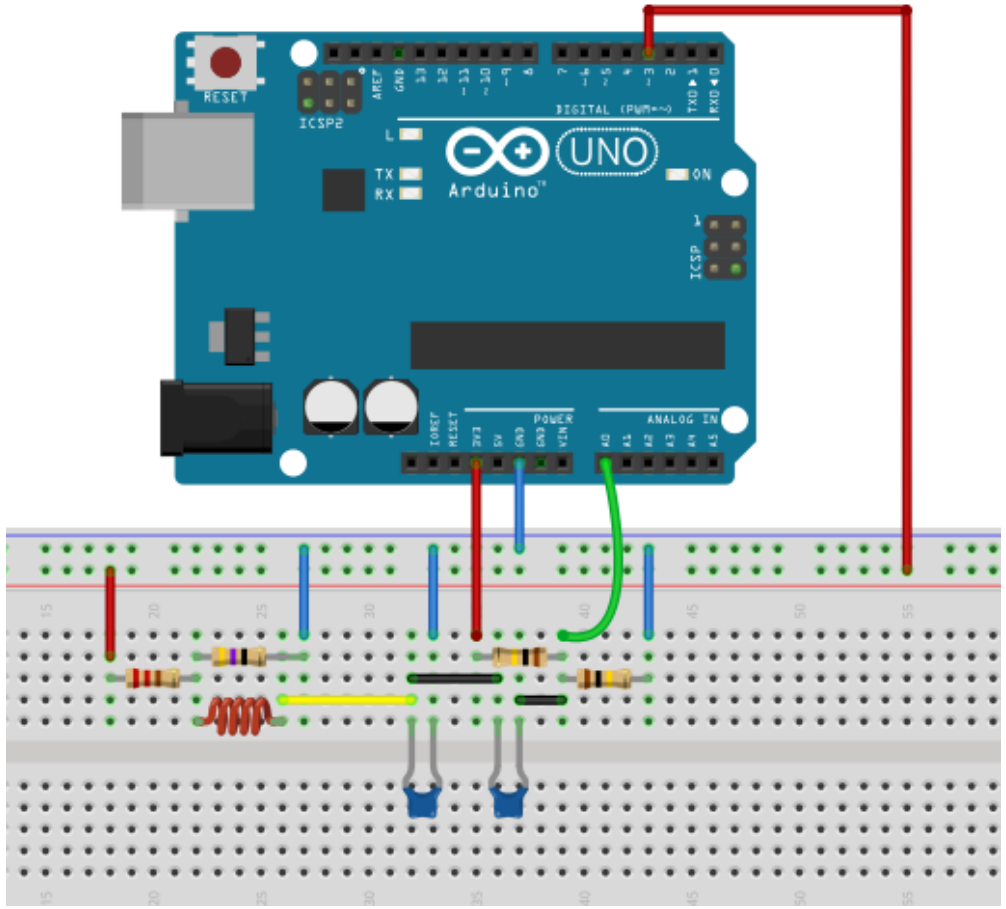
$$Vc(t) = V_0 \left(1 - e^{-\gamma t}(\cos \omega t + \frac{\gamma}{\omega} \sin \omega t)\right)$$

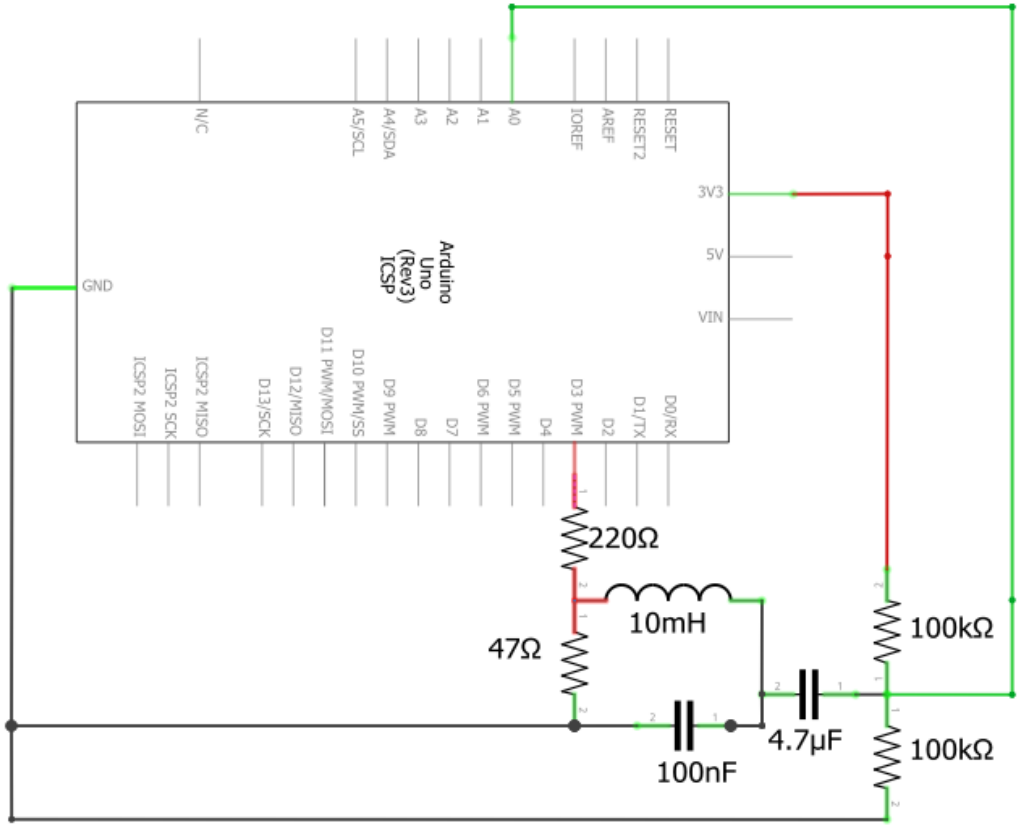
$$i(t) = -\frac{V_0}{L\omega} e^{-\gamma t} \sin \omega t$$

$$\omega = \sqrt{\frac{1}{LC} - \gamma^2}$$

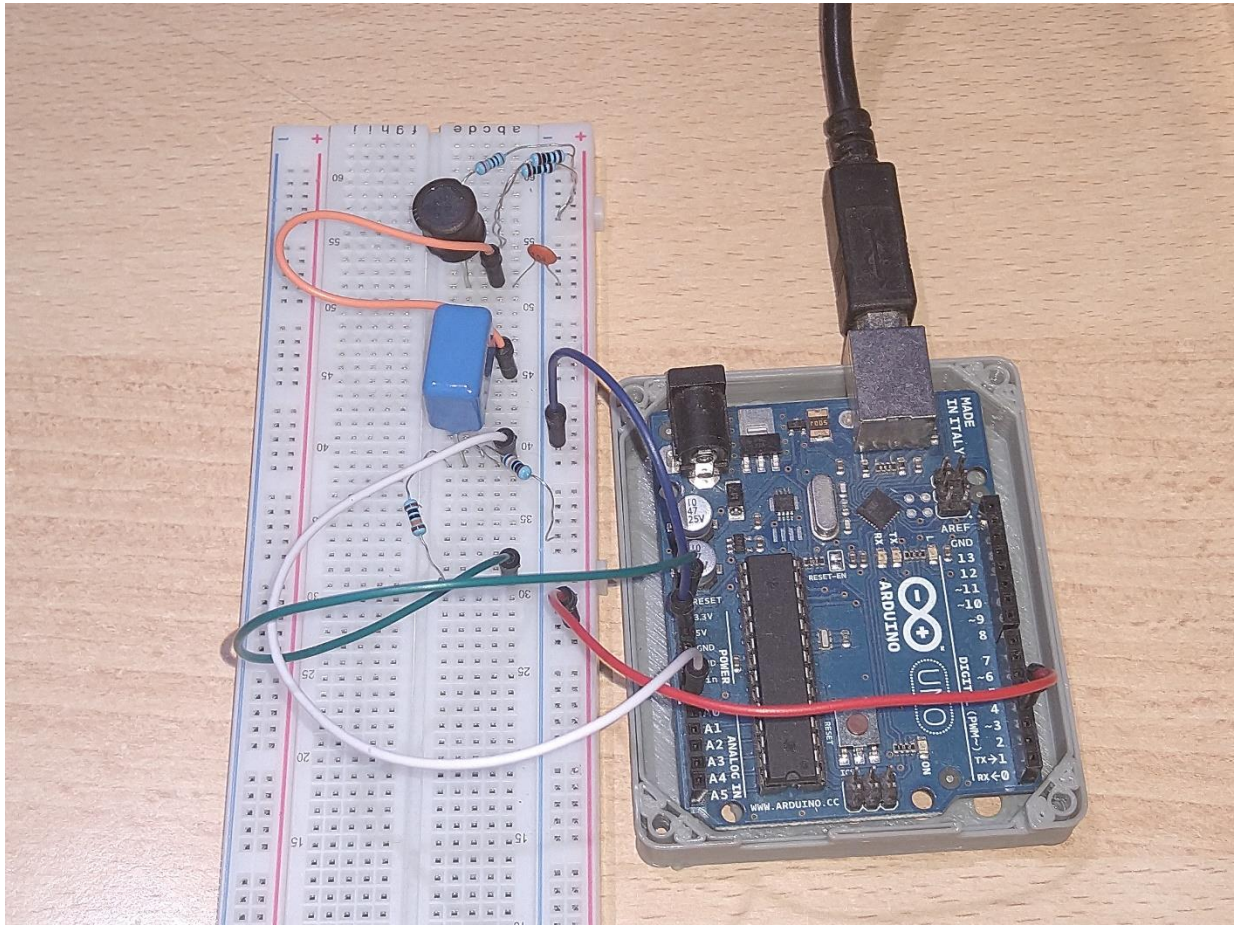
$$i(t) = \frac{V_0}{L\omega} e^{-\gamma t} \sin \omega t$$

Experimento Circuito RLC serie Transitorio Subamortiguado





Experimento Circuito RLC serie - Montaje Transitorio Subamortiguado



Emplear el IDE RLCtransitorio_subamortiguado

Se registra la caída de tensión sobre el capacitor

Restamos la tensión de offset

Tenemos dos columnas de salida: tiempo y Vc

Registramos y guardamos la señal con Serial Plot

RLCtransitorio_subamortiguado Arduino 1.8.13

Archivo Editar Programa Herramientas Ayuda

RLCtransitorio_subamortiguado

```
int Voltaje0[250];
int tiempo[250];
int out1=3; //la señal escalon la tomo del PIN 3
float V0;


void setup() {
  // put your setup code here, to run once:
Serial.begin(57600);
// Set the internal reference and mux.

ADCSRA = (ADCSRA & 0xf8) | 0x03; // clear prescaler | set prescaler to /8
ADCSRA |= 1<<ADEN; // make sure ADC is enabled
pinMode(out1,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
//coloco la salida del pin 3 en ON
//registro la tensión en A0 y el tiempo
float tiempo_ini=micros();
digitalWrite(out1, HIGH);
for(int i=0;i<250;i++){
    Voltaje0[i] = analogRead(A0);
    tiempo[i]= micros()-tiempo_ini;
  }
  for (int i=0; i<250; i++){
    Serial.print(tiempo[i]/1000000,6); //paso el tiempo a segundos
    Serial.print(',');
    V0=Voltaje0[i]*5.0/1023-3.3/2; //resto la tensión continua del divisor colodado a la salida
    Serial.println(V0,3);
```

Compilado

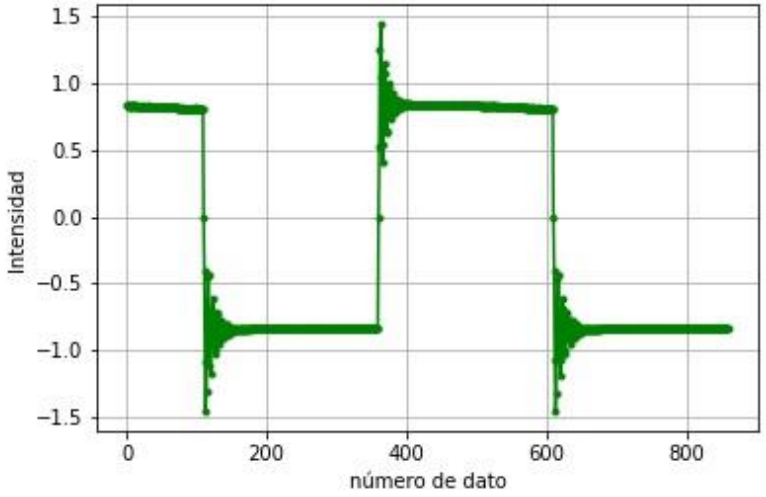
Analizamos los datos en Python

Script RLCtransitoriosubamortiguado

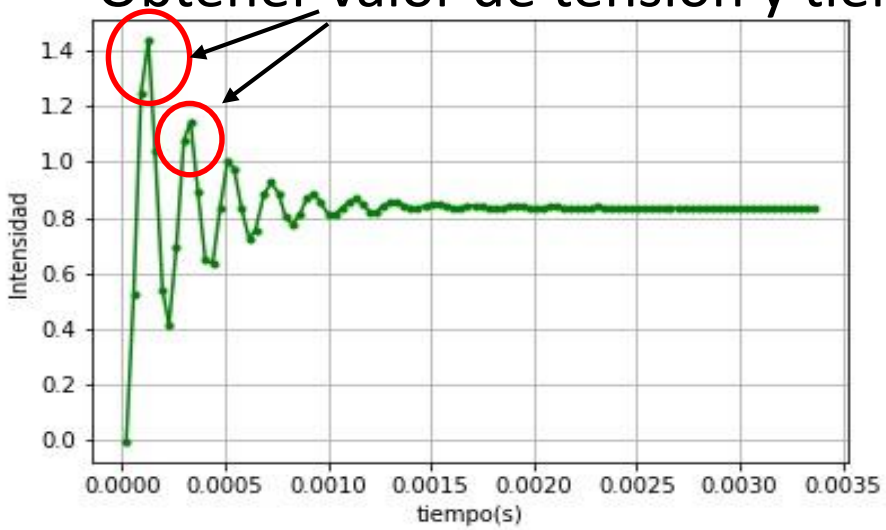
Editor - C:\Users\User\Desktop\Adriana\Laboratorio 3\Analisis phyton\RLCtansitorio\RLCtransitoriosubamortiguado.py

temp.py | RLCtransitoriosubamortiguado.py

```python

#%%
#Loading modules
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import least_squares
import os
from IPython import get_ipython

#%%


#selecciono el grafico en Terminal (inline) o en ventana emergente (qt5)
#get_ipython().run_line_magic('matplotlib', 'inline')
get_ipython().run_line_magic('matplotlib', 'qt5')

#Elijo el directorio donde se encuentran los archivos que voy a analiza
os.chdir (r'C:\Users\User\Desktop\Adriana\Laboratorio 3\RLCtransitorio_UNO\25sept')

print("nombre del archivo completo con terminación .txt incluida")
file = input()

#Importing our data
data = np.loadtxt(file,dtype=float,delimiter = ',',skiprows= 1)



x1i = data[:,0]
y1i = data[:,1]
#construyendo un vector con número de datos
numdatos=np.arange(len(y1i),step=1,dtype=int)


#Graficando nuestros datos en función del número de dato
plt.ion()
plt.close("all")
plt.figure(1)
plt.plot(numdatos, y1i, '-g');
```



Obtener valor de tensión y tiempo



Para estimar γ, ω y amplitud inicial

Spyder (Python 3.7)

Archivo Editar Buscar Código fuente Ejecutar Depurar Terminales Proyectos Herramientas Ver Ayuda

Editor - C:\Users\User\Desktop\Adriana\Laboratorio 3\Analisis phyton\RLCtansitorio\RLCtransitoriosubamortiguado.py

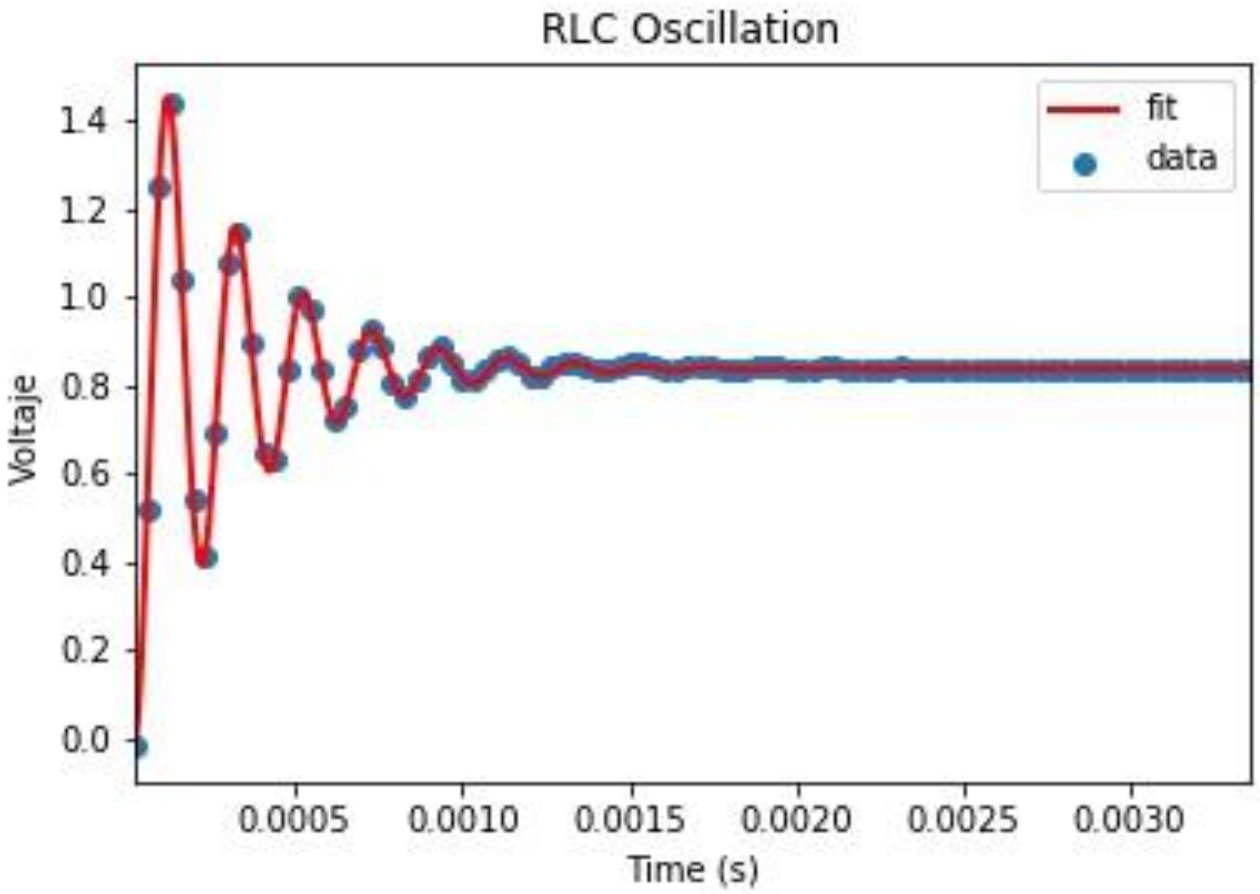
temp.py | RLCtransitoriosobreamortiguado.py | Ley de ohm.py | RLCtransitoriosubamortiguado.py*

```python
#definimos la función que vamos a usar para el ajuste, una función senoidal amortigauda + una consta

def cos_fit_fun_damped(parameters,time):
  a = parameters[0]
  omega = parameters[1]
  offset = parameters[2]
  phi = parameters[3]
  alpha = parameters[4]
  y = a * np.cos(omega * time + phi) * np.exp(-alpha*time) + offset
  return y

def get_residuals(parameters, position_data, time_data):
  theoretical_function = cos_fit_fun_damped(parameters,time_data )
  residuals = np.abs(theoretical_function - position_data)
  return residuals

#Completar con los valores estimados para la señal a partir del gráfico
print("Tiempo correspondiente al primer máximo ")
T1 = float(input())
print("Tiempo correspondiente al segundo máximo ")
T2 = float(input())
print("Valor de tensión  del primer máximo medido ")
Amp1 = float(input())
print("Valor de tensión del segundor máximo medido ")
Amp2 = float(input())
vm = y1[90]


#Guess parameters
guess_amplitude = Amp1-vm #tomada de los datos cargados
guess_omega = 2*np.pi/(T2-T1) #a partir de los datos cargados
guess_offset = vm
guess_phi = np.pi
guess_alpha = -np.log((Amp2-vm)/(Amp1-vm))/(T2-T1)
guess_parameters = [guess_amplitude, guess_omega, guess_offset, guess_phi, guess_alpha]

print(guess_parameters)

#Performing the fit
```

Salida del programa

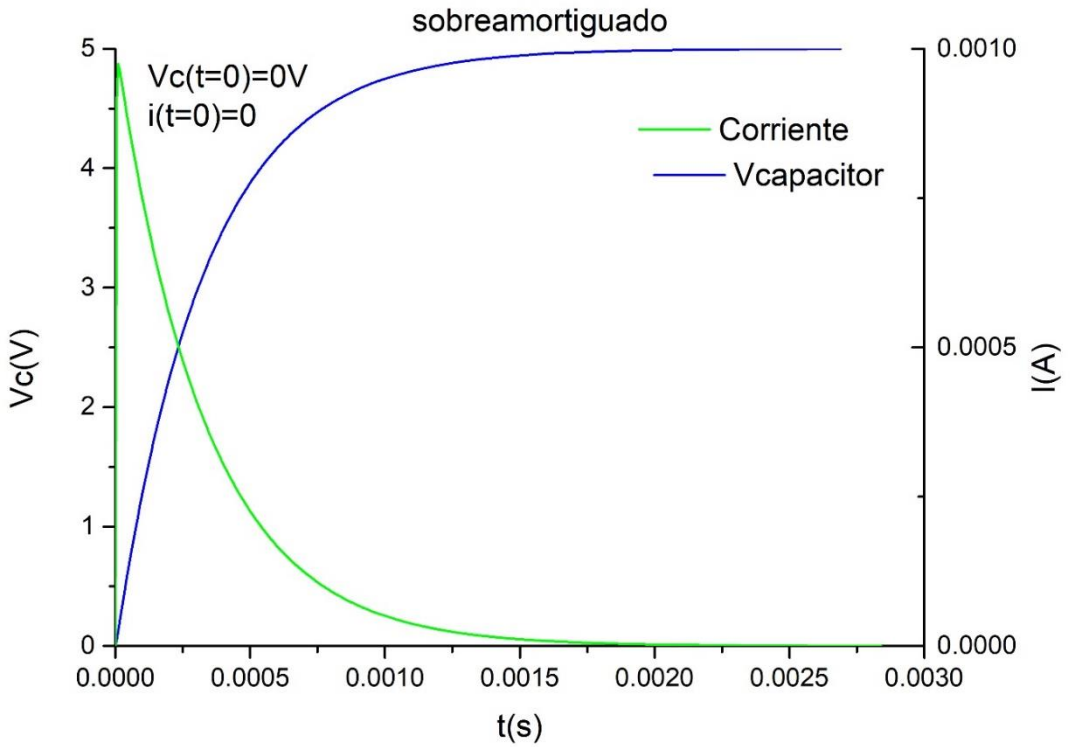
Se carga el valor de C en el script

Calculamos R y L

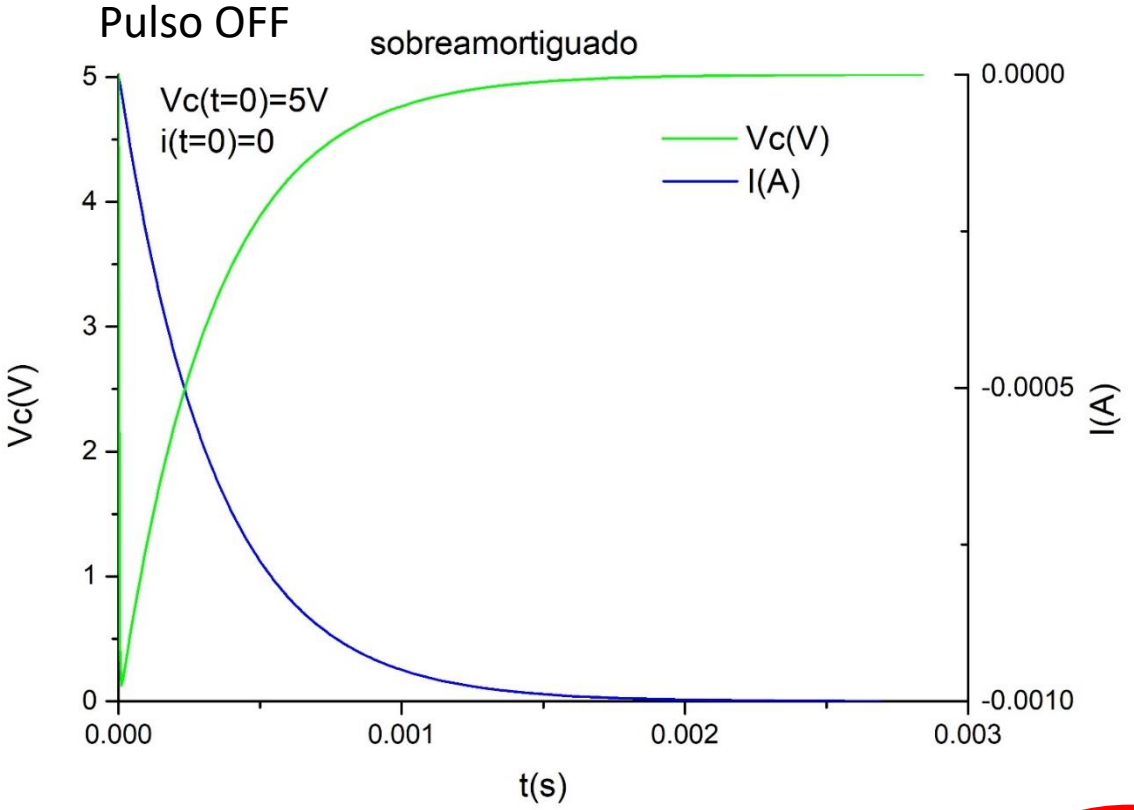


Circuito RLC serie - Transitorio Sobreamortiguado

Pulso ON



Pulso OFF



$$Vc(t) = V_0\left(1 - \frac{e^{-\gamma t}}{2\beta}\left((-\gamma+\beta)e^{-\beta t} + (\gamma+\beta)e^{\beta t}\right)\right)$$

$$\gamma = \frac{R}{2L}$$

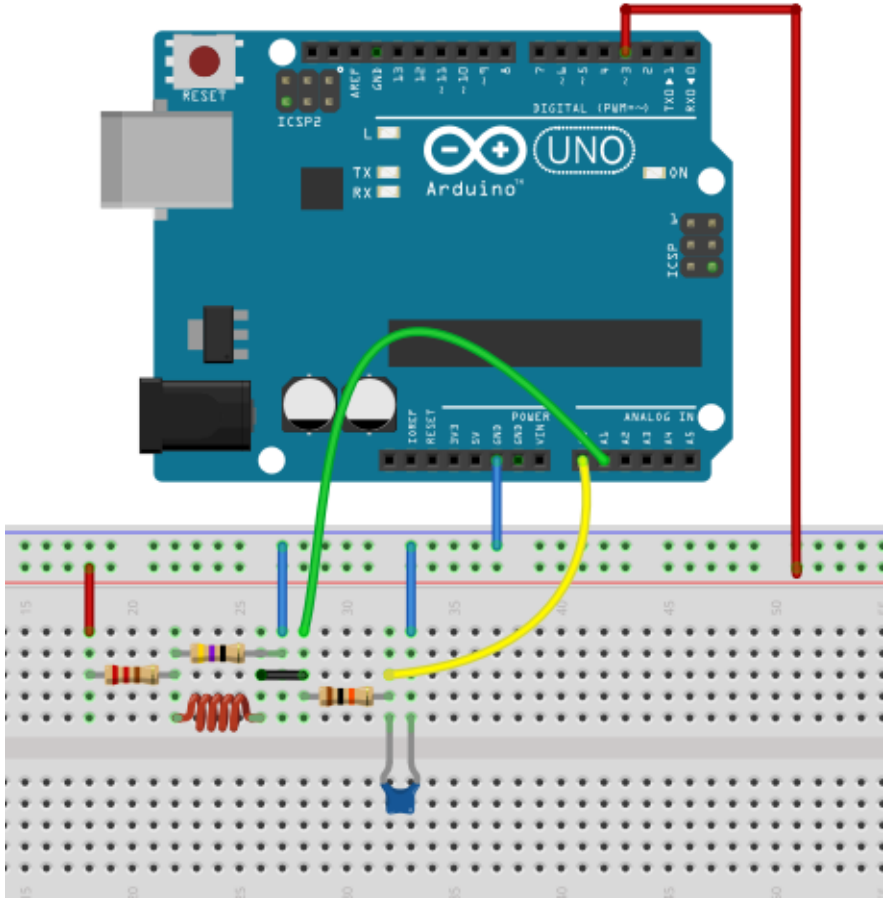
$$Vc(t) = \frac{V_0}{2\beta}e^{-\gamma t}\left((-\gamma+\beta)e^{-\beta t} + (\gamma+\beta)e^{\beta t}\right)$$

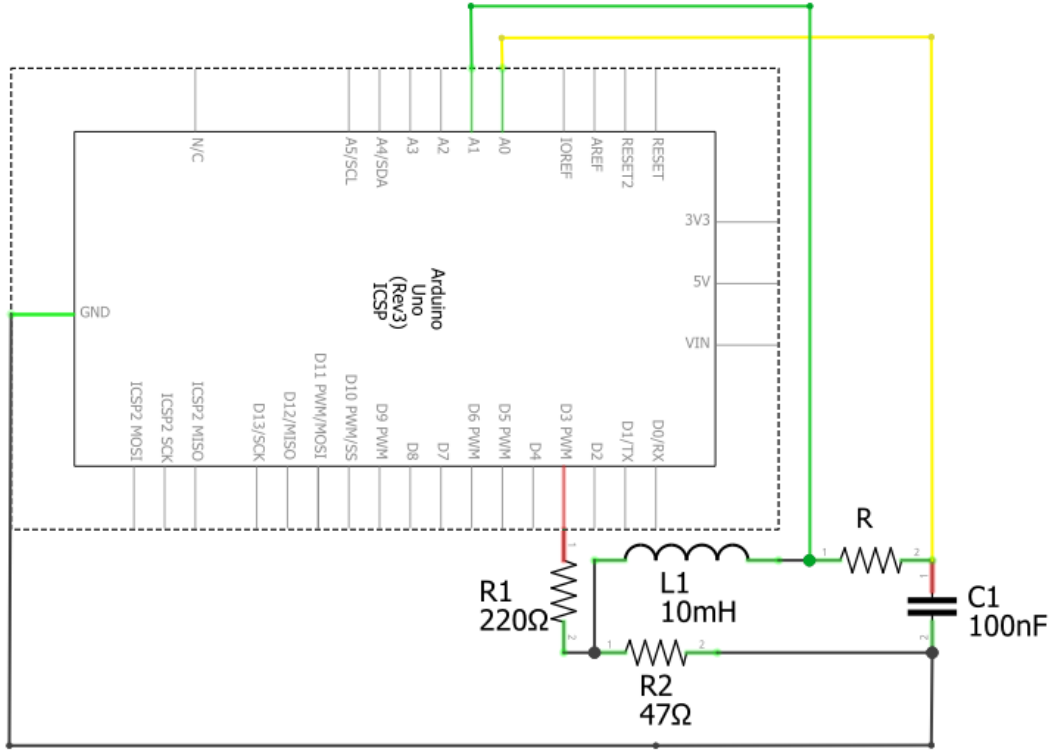
$$i(t) = \frac{V_0}{L\beta}e^{-\gamma t}\frac{\left(e^{\beta t} - e^{-\beta t}\right)}{2}$$

$$\beta = \sqrt{\gamma^2 - \frac{1}{LC}}$$

$$i(t) = -\frac{V_0}{L\beta}e^{-\gamma t}\frac{\left(e^{\beta t} - e^{-\beta t}\right)}{2}$$

Experimento Circuito RLC serie Transitorio Sobremortiguado





R= 4.7kΩ; 10kΩ

Emplear el IDE RLCtransitorio_sobreamortiguado

Se registra la caída de tensión sobre el Capacitor V0 y a la entrada de la resistencia V1

RLCtransitorio_sobreamoriguado Arduino 1.8.13

Archivo Editar Programa Herramientas Ayuda

RLCtransitorio_sobreamoriguado

```
int Voltaje0[150];
int Voltaje1[150];
long tiempo[150];
int out1=3; //alimento el circuito con el PIN 3
float V0;
float V1;
float V;
float t;


void setup() {
  // put your setup code here, to run once:
Serial.begin(57600);
// Set the internal reference and mux.

ADCSRA = (ADCSRA & 0xf8) | 0x04; // clear prescaler | set prescaler to /16
ADCSRA |= 1<<ADEN; // make sure ADC is enabled
pinMode(out1,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:


float tiempo_ini=micros();
digitalWrite(out1, HIGH);
for(int i=0;i<150;i++){
    Voltaje0[i] = analogRead(A0);
    Voltaje1[i] = analogRead(A1);
    tiempo[i]= micros()-tiempo_ini;
  }
```

Emplear el IDE RLCtransitorio_sobreamortiguado

Se registra la caída de tensión sobre el capacitor V0 y a la entrada de la resistencia V1

Obtenemos la caída de tensión en la resistencia V

Tenemos tres columnas de salida: tiempo, V0 y V

Registramos y guardamos la señal con Serial Plot

RLCtransitorio_sobreamoriguado Arduino 1.8.13

Archivo Editar Programa Herramientas Ayuda

RLCtransitorio_sobreamoriguado

```
    Serial.print(t,5);
    Serial.print(',');
    V0=Voltaje0[i]*5.0/1023;
    V1=Voltaje1[i]*5.0/1023;
    V=(V1-V0);
    Serial.print(V0,3) ;
    Serial.print(',');
    Serial.println(V,6) ;

  }

 delay(500);
tiempo_ini=micros()-tiempo[149];
digitalWrite(out1,LOW );
for(int i=0;i<150;i++){
    Voltaje0[i] = analogRead(A0);
    Voltaje1[i] = analogRead(A1);
    tiempo[i]= micros()-tiempo_ini;
  }
  for (int i=0; i<150; i++){
    t=tiempo[i]/1000000.00000;
    Serial.print(t,5);
    Serial.print(',');
    V0=Voltaje0[i]*5.0/1023;
    V1=Voltaje1[i]*5.0/1023;
    V=(V1-V0);
    Serial.print(V0,3) ;
    Serial.print(',');
    Serial.println(V,3) ;
  }
delay(500);
```

Analizamos los datos en Python

Script RLCtransitoriosobreamortiguado

Cargamos el valor nominal de L y R para estimar los parámetros para el ajuste

Archivo Editar Buscar Código fuente Ejecutar Depurar Terminales Proyectos Herramientas Ver Ayuda

Editor - C:\Users\User\Desktop\Adriana\Laboratorio 3\Analisis phyton\RLCtansitorio\RLCtransitoriosobreamortiguado.py

temp.py | RLCtransitoriosobreamortiguado.py* | Ley de ohm.py | RLCtransitoriosubamortiguado.py*

```python
#Loading modules
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import least_squares
from scipy.optimize import curve_fit
import os
from IPython import get_ipython

#%%

#selecciono el grafico en Terminal (inline) o en ventana emergente (qt5)
#get_ipython().run_line_magic('matplotlib', 'inline')
get_ipython().run_line_magic('matplotlib', 'qt5')

#Elijo el directorio donde se encuentran los archivos que voy a analiza
os.chdir (r'C:\Users\User\Desktop\Adriana\Laboratorio 3\RLCtransitorio_sobreamoriguado')

print("nombre del archivo completo con terminación .txt incluida")
file = input()

#Completar con los parámetros del circuito
print("Cargar valor de la Resistencia del cirucito en ohm")
R= float(input())
print("Cargar valor de la Inductancia del cirucito en H")
L= float(input())
#print("Cargar valor de la Capacidad del cirucito en F")
#C= float(input())

#Importing our data
data = np.loadtxt(file,dtype=float,delimiter = ',',skiprows= 1)

x1i = data[:,0]
y1i = data[:,1]
y2i = data[:,2]


#limitar los datos al rango de interés
j=0
datoinicial1=0
```

Ajustamos *Vc* e *I* con un modelo aproximado

$$Vc = a * e^{-bt} + Vfinal$$

$$i = c * e^{-dt}$$

Ajuste con modelo completo, estimando los valores de los parámetros a partir del modelo aproximado y usando L y R

Calculamos R , L y C

Spyder (Python 3.7)

Archivo Editar Buscar Código fuente Ejecutar Depurar Terminales Proyectos Herramientas

Editor - C:\Users\User\Desktop\Adriana\Laboratorio 3\Analisis phyton\RLCtansitorio\RLCtransitoriosobreamortiguado.py

temp.py | RLCtransitoriosobreamortiguado.py* | Ley de ohm.py | RLCtransitoriosubamortigua

```python
#Modelo aproximado para la corriente y la VC
Vfinal=y1[-1]
def aproximacionVC(time, a,b):
    y= a*np.exp(-b*time)+Vfinal
    return y

def aproximacionI(time, a,b):
    y= a*np.exp(-b*time)
    return y

# Ajuste de los datos
popt1, pcov01 = curve_fit(aproximacionVC, x1,y1)
popt2, pcov02 = curve_fit(aproximacionI, x1,y2)

funcredVC=aproximacionVC(x1,*popt1)
funcredI=aproximacionI(x1,*popt2)
# ESTIMO LOS ERRORES ( 1 sigma )
perr1 = np.sqrt(np.diag(pcov01))
perr2 = np.sqrt(np.diag(pcov02))
N=len(x1)

#modelo completo para la corriente
def modeloI(parameters1,time):
  A2= parameters1[0]
  landa12 = parameters1[1]
  landa22= parameters1[2]
  y = A2*(np.exp(landa12*time)-np.exp(landa22*time))
  return y

#Guess parameters
guess_landa12 = -popt2[1]
guess_landa22 = (-R/L- popt2[1]) #taken from theory
guess_A2 = popt2[0]

guess_parameters2 = [guess_A2, guess_landa12, guess_landa22]


#modelo completo para la caida sobre el capacitor
def modeloVC(parameters1,time):
```

Punto de control – RLC serie transitorio

- Circuito **subamortiguado** medir V capacitor
  Obtener L y R a partir del ajuste
  Comparar con los valores nominales

- Circuito **sobreamortiguado**, medir con los dos valores de R sugeridos
  Comparar modelo aproximado con modelo completo
  Comparar el comportamiento de las señales al variar R
  Obtener C, L y R a partir del ajuste
  Comparar con los valores nominales