

**Bangladesh University of Engineering and Technology**

**Depertment of Electrical and Electronic Engineering**

**Course No:** EEE 212

**Course Name:** Numerical Technique Laboratory

**Project Title:** Designing first order RLC circuit from given transient response and figuring out the system to generate output for any input

**Group Members:**

Tasnim Nishat Islam (1706092)

Kazi Ishrak Ahmed (1706077)

# Introduction

We want to design first order RL, RC and second order RLC circuits of three conditions (under damped, over damped, critically damped) from given transient response by plotting the curve through regression. Once we get the circuit we will figure out the system, so that for any given input the output can be found through convolution.

# Methodology

Designing First or second order circuit given response of the circuit as the time passes by,an equivalent series or parallel RLC circuit will be designed.

The procedures—

1. First the data will be passed through non-linear regression

2.The curve gotten will be curve fitted with the most suited exponentially responded graph.

3. Alpha, omega, tau will be obtained which will give the value of resistance, capacitor and inductor for both series and parallel combination

4. The circuit will be provided through diagram

5. Then the transfer function of the system will be obtained

7. Once we get the transfer function, we will be able to find any output by convoluting the input with the system

8. Some examples will be provided of output for different input for the given circuits.

# Steps

**Finding the exact transient response**

1. Browse the “.xlxs” file
2. Fetch the data and divide them to t and y(t)
3. 4 transient response is introduced
4. Compare them and get GOF(Parameters showing how good the fit is) values.
5. User finds the best fitted curve
6. the curve is fitted with the given data.

**RLC circuit design**

1. Get the value of R form user
2. Get L, C value
   1. In the function we pass the constant values, preferred equation number and R
   2. If preferred equation is 1, circuit is either RC or RL
      1. RC: C=tau/R(tau from constant values obtained from the fit curve)
      2. RL: ττL=tau\*R
   3. Otherwise circuit can be either Series or Parallel RLC, as user wishes it to be. For that we obtain the value of alpha and omega from constant values and preferred equation number
      1. For overdamped: α=(s1+s2)/-2; ω=root(s1\*s2)
      2. For underdamped: α=ω =a;
      3. For critically damped: α=a; ω =root(w2+a2)
   4. If parallel
      1. C=1/(2\* α \*R)
      2. L=1/(C\*ω2)
   5. If series:
      1. L=R/(2\* α)
      2. C=1/(L\*ω2)

**Get any output for any input**

1. First we get the input signal. Our project is able to read the elementary functions like unit step & ramp function through the process\_signal function.
2. To find the Laplace transform of the signal define\_signal() is used. If periodic time period is asked from the user.
3. To plot the graphs RC, RL,RLC series, RLC parallel are considered, we first find the laplace transform of the input signal, then convolute it with the transfer function and get the output response.
4. For modifying the transfer function, we have applied proper voltage divider rule and current divider rule is applied.

**Algorithm**

1.get the file

2. fullpathname

3.read the file (data only)

4.global variable t,y,eqn

5.first row=t; second row=y

6. eqn = ["init\*exp(-x/tau)";"A1\*exp(s1\*x)+A2\*exp(s2\*x)";"(A2+A1\*x)\*exp(-a\*x)";"exp(-a\*x)\*(A1\*cos(w\*x)+A2\*sin(w\*x))"];

7. my\_plot function

a. inputs eqn,t,y

b. gofMat is a structure type variable

c. fit the equations with t and y

d. sse, rsquare, adjrsquare, rmse gofMat(1)-gofMat(4)

e. end for

8. call my\_plot

9. convert gof\_Mat to table

10.gofMat\_table=convert gofMat to array

11.show gofMat\_table

12. get selected\_equation

13. fit t,y,the number of equation is selected

14. plot f,t,y

15. get selected\_circuit\_type

16. get R

17. findlc function includes

1. inputs input, selected equation number, R, selected\_circuit
2. if selected\_equation\_number=1
   1. case ‘RC’
      1. C = f.tau/R;
      2. L=0;
   2. case 'RL'
      1. L = f.tau\*R;
      2. C=0;
3. else
   1. find\_aw includes
      1. switch idx
         1. case 2
            1. a=(f.s1+f.s2)/-2;
            2. w=sqrt(f.s1\*f.s2);
         2. case 3
            1. a = f.a;
            2. w = a;
         3. case 4
            1. a=f.a;
            2. w=sqrt(f.w^2+a^2);
   2. end switch
   3. switch x
      1. case 'RLC series'
         1. L=R/(2\*a);
         2. C=1/(L\*w^2);
      2. case 'RLC parallel'
         1. C=1/(2\*a\*R);
         2. L=1/(C\*w^2);
   4. end switch
4. end if

18. Fetch R, L, C values

19. In the proper axes handle show the proper image

switch selected\_circuit\_type

case 'RC'

imshow('RC.jpg');

case 'RL'

imshow('RL.jpg');

case 'RLC series'

imshow('RLC series.jpg');

case 'RLC parallel'

imshow('RLC parallel.jpg');

end

20. define\_signal includes

input xt\_input, timefinal\_input, var, time\_period (var=periodic/aperiodic)

syms t real s

xt = xt\_input;

tf = tf\_input;

xt = process\_signal(xt);

process signal includes

input(xt)

syms t;

u is replaced with ‘heaviside’

seg = extractBetween(xt,'r',')');

n = length(seg);

new = string();

for i = 1:n

k = seg(i);

new(i) = strjoin(string({k ')\*heaviside'}));

endfor

for i = 1:n

pos = strfind(xt,'r');

pos = pos(i);

if pos

xt = insertAfter(xt,pos,new(i));

end

end

xt = str2sym(erase(xt,{'amp';'r'}));

switch var

case 'Aperiodic/Triagonometric'

fun = xt laplaced then simplified

case 'Periodic'

T = timeperiod;

fun = xt laplaced then simplified

/(1-exp(-s\*T)));

xt = ilaplace(fun);

if (hastype(xt,'ilaplace'))

msgbox("Inverse Laplace can't be found");

endif

endswitch

21. var,xt, timefinal, timeperiod fetched

22. define\_signal called

23. function picture includes

function picture(xt,fun,R,L,C,x,tf,p)

syms t real,s;

switch x

case 'RC'

I = simplify(fun/(R+1/(s\*C)));

switch p

case 'Ic'

yt = ilaplace(I);

case 'Vr'

yt = ilaplace(I\*R);

case 'Vc'

yt = ilaplace(I/(s\*C));

case 'Ir'

yt = ilaplace(I);

end

case 'RL'

I = simplify(fun/(R+s\*L));

switch p

case 'Il'

yt = ilaplace(I);

case 'Vr'

yt = ilaplace(I\*R);

case 'Vl'

yt = ilaplace(I\*s\*L);

case 'Ir'

yt = ilaplace(I);

end

case 'Series'

I= simplify(fun/(R+s\*L+1/(s\*C)));

switch p

case 'Ir'

yt = ilaplace(I);

case 'Vr'

yt = ilaplace(I\*R);

case 'Vl'

yt = ilaplace(I\*s\*L);

case 'Vc'

yt = ilaplace(I/(s\*C));

case 'Il'

yt = ilaplace(I);

case 'Ic'

yt = ilaplace(I);

end

case 'Parallel'

switch p

case 'Vc'

yt = xt;

case 'Ir'

yt = ilaplace(fun/R);

case 'Il'

yt = ilaplace(fun/(s\*L));

case 'Ic'

yt = ilaplace(fun\*s\*C);

case 'Vl'

yt = xt;

case 'Vr'

yt = xt;

end

end

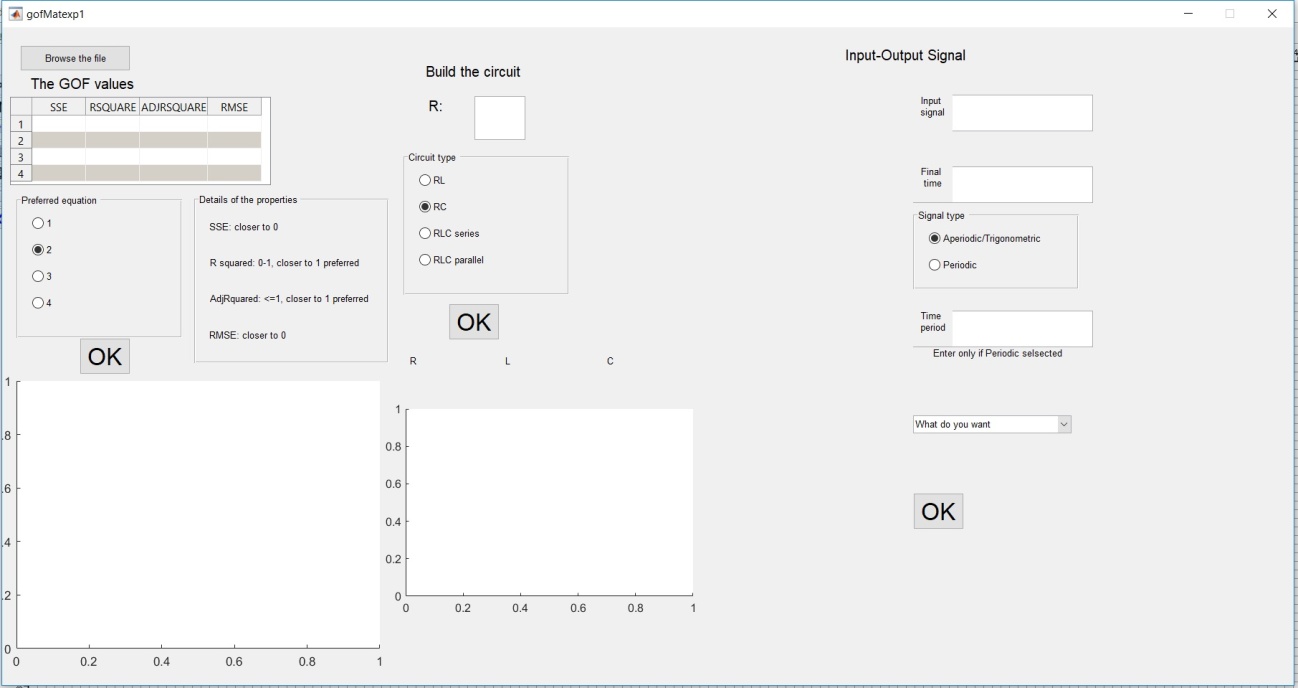
xt=vpa(simplify(xt,'IgnoreAnalyticConstraints',true,'Steps',50),2);

figure;

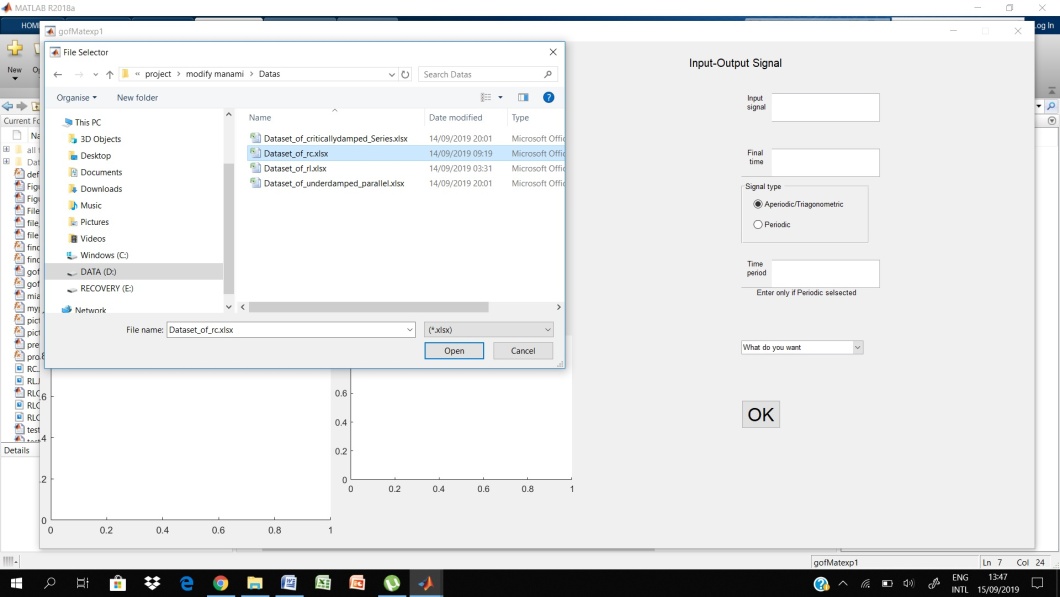
plot xt, yt

**How the GUI works**

1. The blank GUI



1. We get the output from a data file.



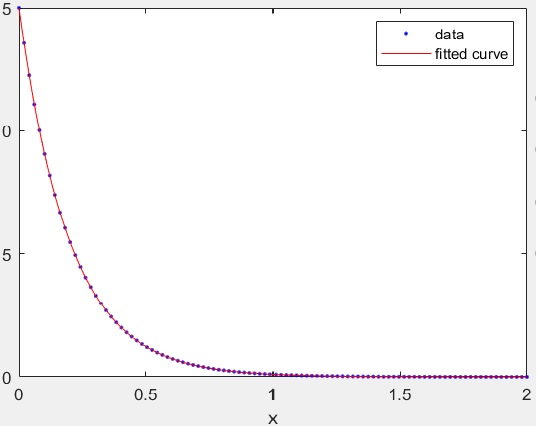
1. The GUI shows the GOF values. GOF is a MATLab function which demonstrates which curve will give the best fit. User has to mainly choose the value.



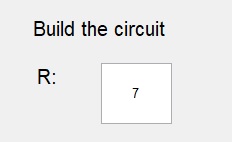
1. Selecting the preferred equation: By close observation of four values, comparing them user chooses a value.



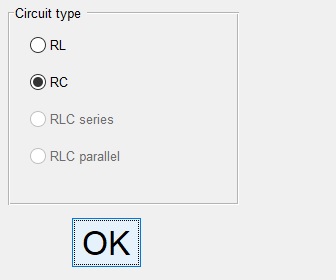
1. User gets the fitted curve from the given equation of the code.



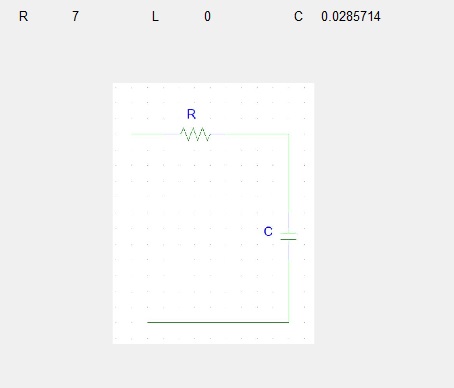
1. To build the circuit a random value of resistance is taken.



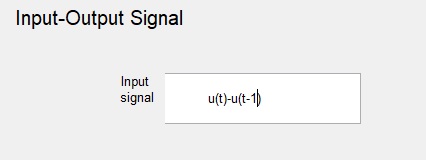
1. The user will be given the choice which circuit he wants.



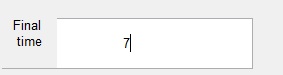
1. The R, L, C value is shown along with the equivalent circuit diagram.



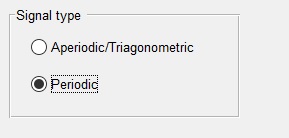
1. User inputs an input signal



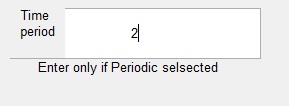
1. Final time is given by the user (second).



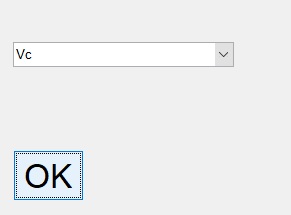
1. Periodic or Aperiodic/Trigonometric can be chosen.



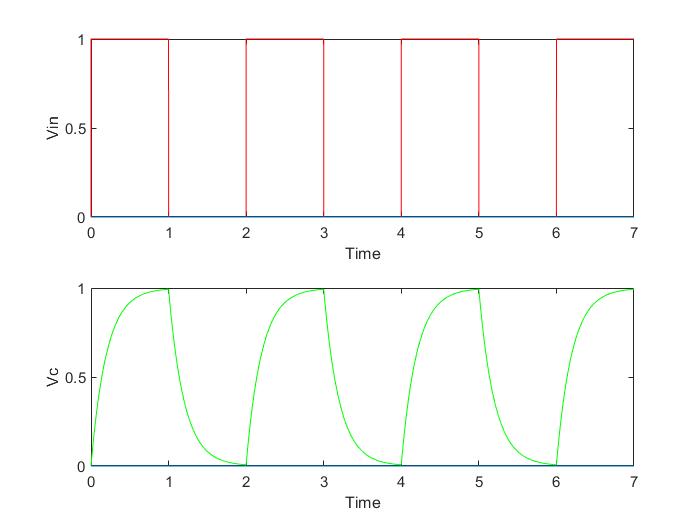
1. If Periodic user gives time period. (This property is not applicable otherwise)



1. The required signal we want to see, will be seen from the pop-up menu.



1. The input signal and output signal are shown in the figure file.



**Discussion:**

While giving value of transient response, the user is encouraged to put more and more values to get correct curve. The real curve can have maxima to minima at some points. So, we should put enough values at that region to detect it. A random start point is selected for fitting the curve. So, the user is encouraged to run the code several times to get desired values.

The system response is a design problem. So, it can be obtained by various combinations of RLC. So, the user is asked to choose the value of R wisely.

The function “ilaplace” may not be able to find inverse Laplace transform of standard form always. Then a numerical or higher order approach can be made like Euler and Talbot inversion which is not our objective in this project. So, for the time being we ignored that.

Meanwhile, the GUI and the modified functions were done in MATLAB 2018 and the main code in MATLAB 2019a, so required output for some inputs in the third part of the code could not be shown due to unavailability of some functions in the MATLAB 2018. Sorry for the inconvenience. By the way, the user can get his output from the main code instead.