

# به نام خدا

گزارش بخش نهایی فاز دوم پروژه تئوری مدارهای الکتریکی  
دکتر فاطمی زاده

سید محمد محسنی زنوزی

۹۹۱۰۲۱۷۸

میتوانید کد مربوط به پروژه را در فایل PH2\_II\_99102178.m مشاهده کنید

## نحوه کار با نرم افزار:

ابتدا نت لیست مربوط به مدار مورد نظر خود را در فایل .txt ذخیره کنید (فرمت ورودی در ادامه آورده خواهد شد). هنگام اجرای برنامه از شما خواسته خواهد شد نام فایل نت لیست و همچنین نام فایلی که میخواهید خروجی در آن ذخیره شود (نیازی نیست از قبل ایجاد شده باشد) را به ترتیب وارد کنید:

```
Command Window
fx Enter netList file name :
```

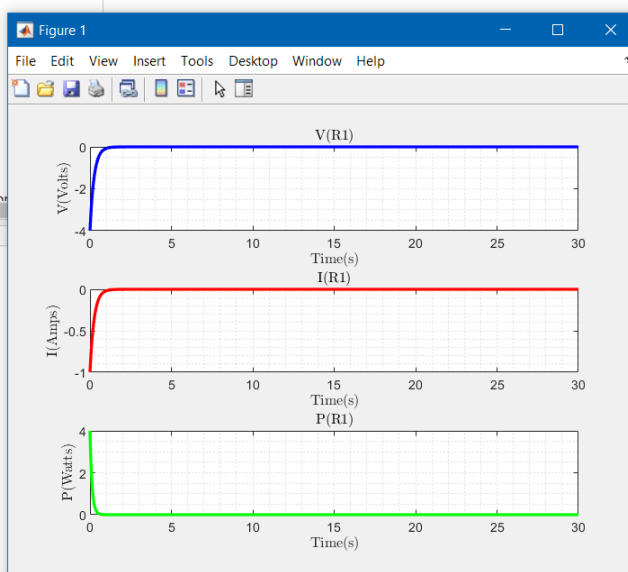
```
Command Window
Enter netList file name : netList2.txt
Enter output file name : output2.txt
output file generated ...
fx enter name of the element or stop simulation using "end" :
```

پس از دادن نام این فایل ها، در صورت درست بودن ورودی ها مشاهده میکنید که مدار حل شده و فایل خروجی ساخته میشود، سپس از شما پرسیده میشود که نام یکی از المان هایی که در مدار بود را وارد کنید تا نمودار ولتاژ و جریان و توان آن را بکشد، پس از وارد کردن نام المان در صورت وجود المان مورد، برنامه از شما درخواست میکند به بازه زمانی شبیه سازی را وارد کنید تا نمودارهای این المان را در آن بازه رسم کند:

```
25 if strcmp(currentComponent(1), 'ML')
26     if ~(any(Nodes(:) == str2double(currentComponent(6))))
27         Nodes(end + 1) = str2double(currentComponent(6));
28     end
29
30     if ~(any(Nodes(:) == str2double(currentComponent(7))))
31         Nodes(end + 1) = str2double(currentComponent(7));
32     end
33 end
34
35 if (strcmp(currentComponent(1), 'R') || strcmp(currentComponent(1), 'C'))
```

Command Window

```
Enter netList file name : netList2.txt
Enter output file name : output2.txt
output file generated ...
enter name of the element or stop simulation using "end" : R1
Enter the time interval (ex : "0 30") : 0 30
fx enter name of the element or stop simulation using "end" :
```



همانطور که مشاهده میکنید پس از این مراحل نمودار های المان داده شده برای شما رسم میشوند و برنامه دوباره از شما میپرسد که آیا میخواهید المان دیگری را نیز بررسی کنید یا خیر، در صورتی که دیگر نیازی به شبیه سازی نداشتید کافیت "end" را در ورودی بنویسید تا برنامه به پایان برسد. اگر هم المان دیگری را میخواستید بررسی کنید نام آن و سپس بازه زمانی را وارد کنید تا شبیه سازی شود، ولی در نظر داشته باشید چون امکان داشت نیاز داشته باشید که چند نمودار را همزمان مشاهده کنید نمودار های قبلی بسته نمیشوند و میتوانستیم برای اینکه بسته شوند در ابتدای while یکبار all close میکردیم ولی اینکار را به دلیلی که گفته شد نکردیم. **الگوریتم استفاده شده MNA معرفی شده در فصل ۱۲\* کتاب نظریه اساسی مدارها و شبکه ها نوشته آقای جبه دار مارالانی است.**

فرمت ورودی ها از این قرار است:

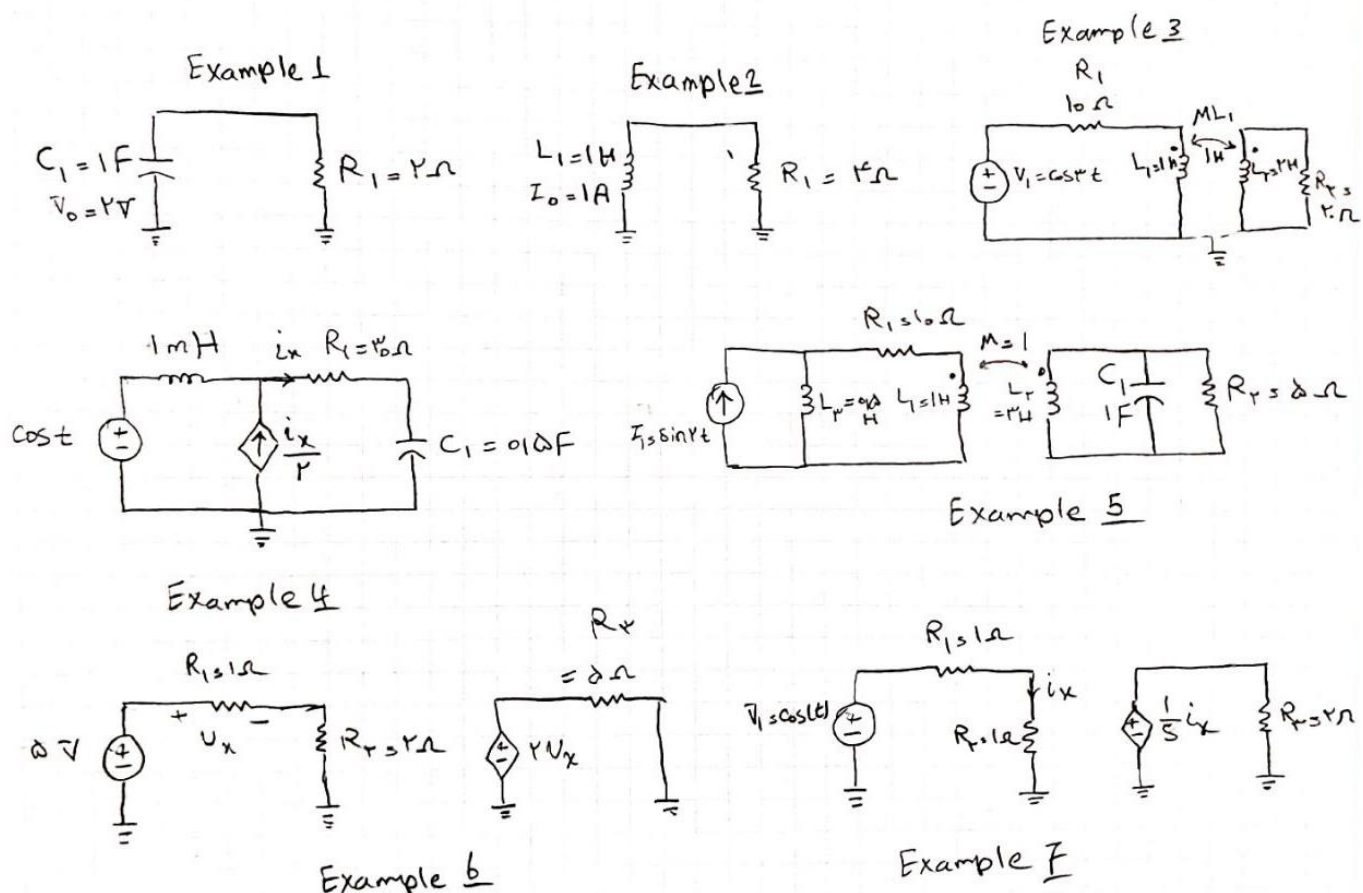
C,name,n1,n2,capacity,initialCondition  
 L,name,n1,n2,inductance,initialCondition  
 R,name,n1,n2,resistance  
 V,name,n1,n2,voltage  
 I,name,n1,n2,current  
 ML,name,n1,n2,inductance1,n3,n4,inductance2,mutualinductance  
 Z,name,n1,n2,sourceNode1,sourceNode2,gain  
 H,name,n1,n2,sourceNode1,sourceNode2,gain  
 Y,name,n1,n2,sourceNode1,sourceNode2,gain  
 Y,name,n1,n2,sourceNode1,sourceNode2,gain

که به این ها توجه داشته باشید شرایط اولیه خازن و سلف را نیز نرم افزار ما هندل میکند. متغیر های gain هم میتوانند تابع شبکه طور باشند (مثلا  $s/(s+1)$ )، ورودی های ولتاژ و جریان برای منابع نابسته هم میتوانند تابعی از زمان باشند (مثل  $\cos(3*t)$ ). **(خلاصه کار خفنی بوده و همه جزئیات را هندل میکند و لایق نمره اضافه است ۹۵)**

همچنین دقت کنید که sourceNode1 و sourceNode2 همان گره هایی هستند که منبع وابسته، به جریان اتصال کوتاه بین آن دو یا ولتاژ بین آن دو بستگی دارد. بدیهی است اگر بخوایم جریان یک شاخه را بسنجیم دو گره در مسیر ورودی شاخه که جریان از این مسیر وارد میشود را در نظر میگیریم تا همان جریان شاخه به عنوان متغیری که منبع وابسته به آن بستگی دارد شناسایی شود.

در صفحه بعد چند نمونه ورودی و خروجی آورده شده است که آنها را بررسی میکنیم:

نمونه ها:



همه این نمونه ها در فایل های `netlist(i).txt` که شماره نمونه است آورده شده اند و میتوانید شبیه سازی کرده و خروجی هرکدام را مشاهده کنید، در کل نیز بهتر است که از نمودار برای آنالیز استفاده کنید چون پاسخ ها کامل هستند و در فایل `.txt` مربوط به خروجی ممکن است با نوشته های طولانی مواجه شوید. چند تا از نمونه هارا در اینجا بررسی میکنیم:

نمونه ۱:

فایل نت لیست را مشاهده میکنید:

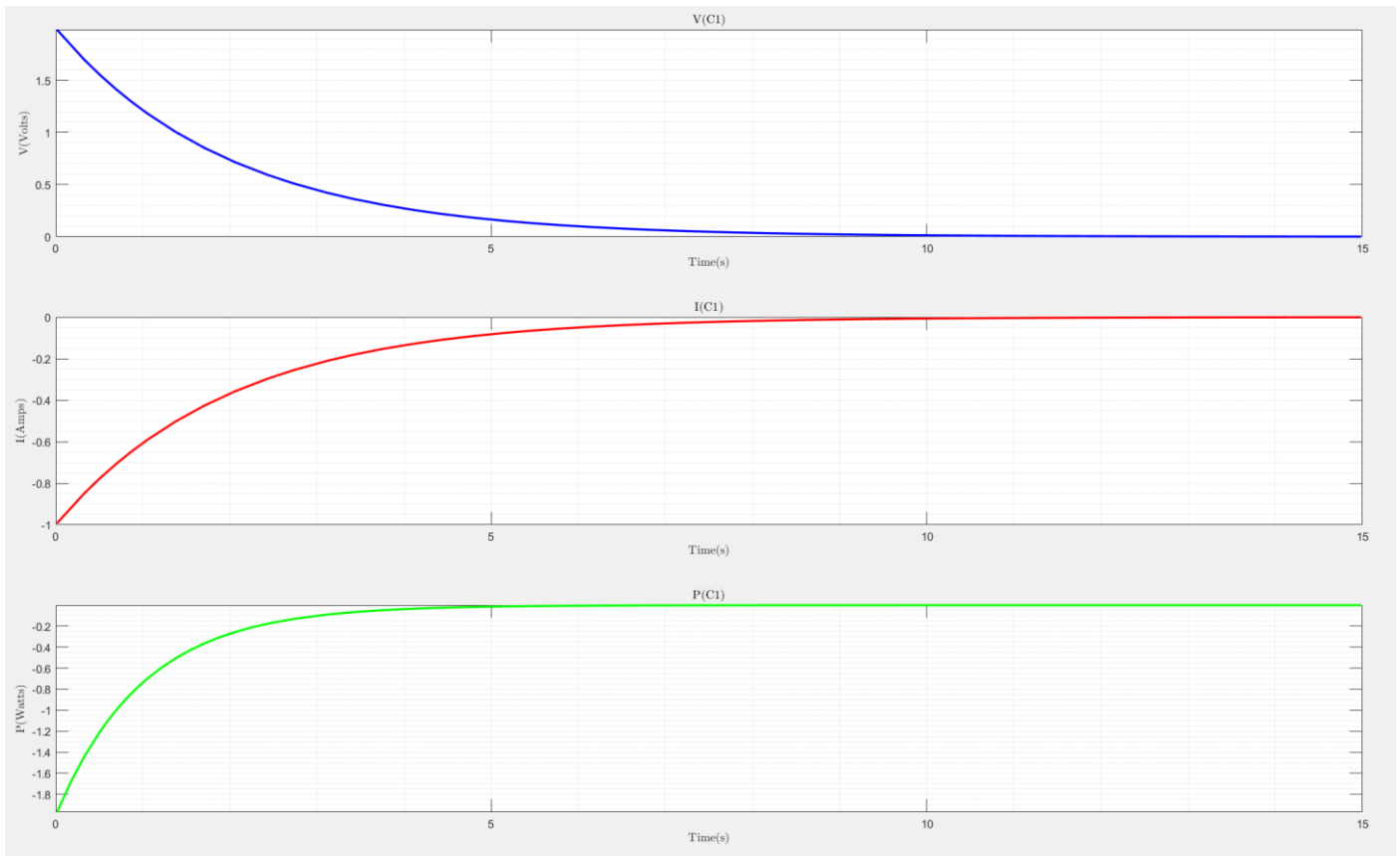
```
C,C1,1,0,1,2
R,R1,1,0,2
```

فایل خروجی:

```
<C1><2*exp(-t/2)><2*dirac(t) - exp(-t/2)><4*dirac(t) - 2*exp(-t)>
<R1><2*exp(-t/2)><exp(-t/2)><2*exp(-t)>
```

نمودار مربوط به C1:

enter name of the element or stop simulation using "end" : C1  
Enter the time interval (ex : "0 3") : 0 15



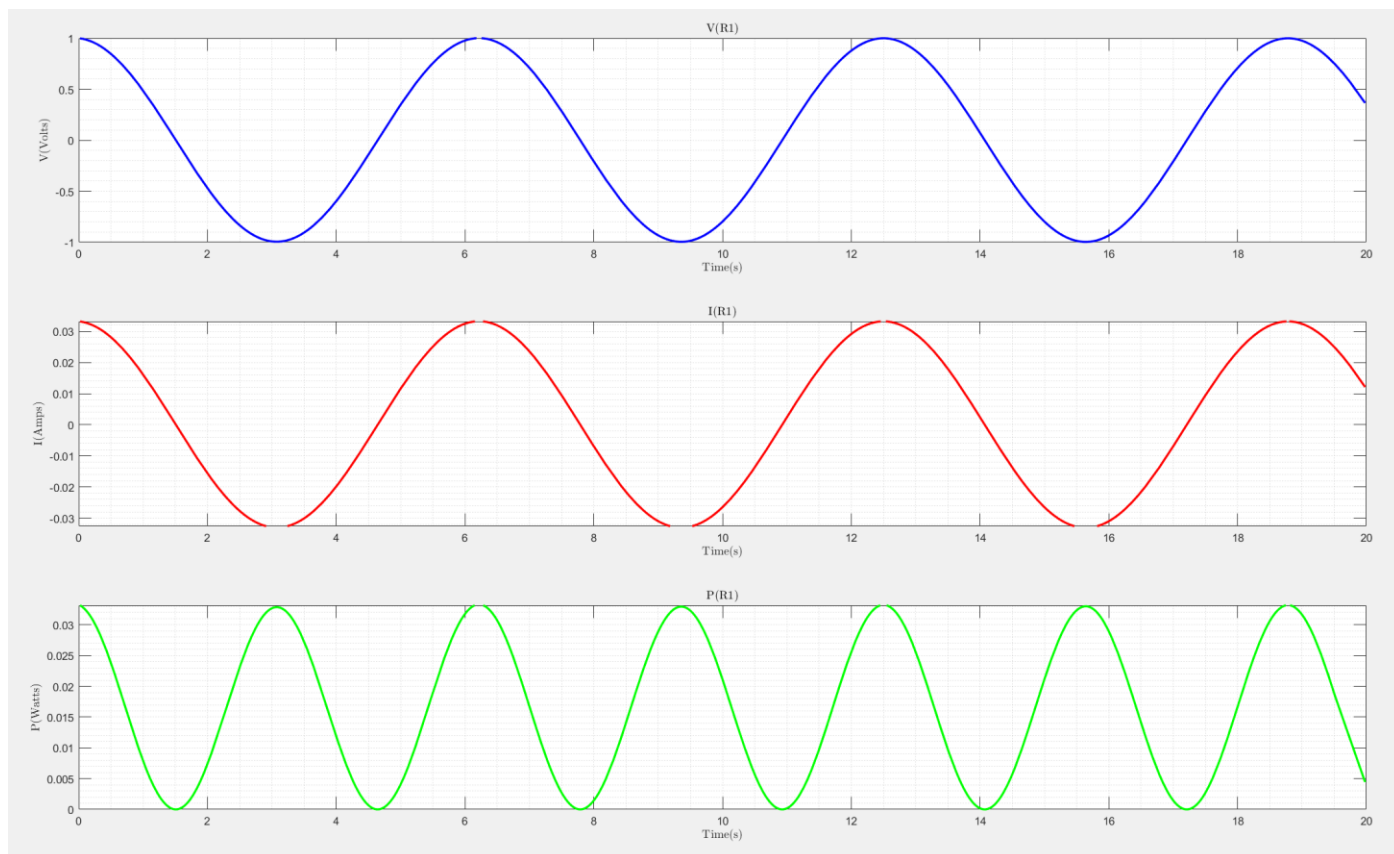
نمونه ۴:

نت لیست:

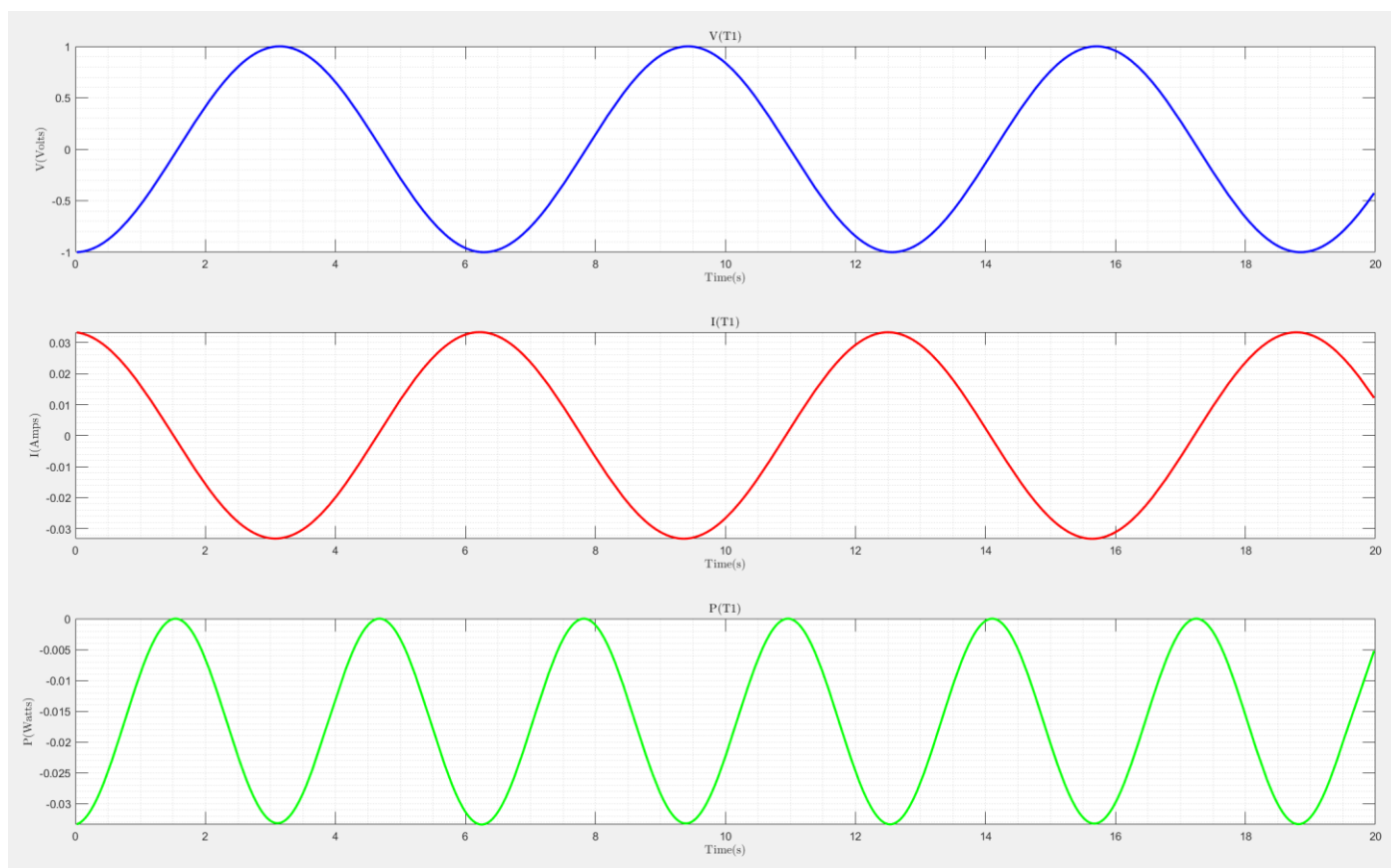
```
V,V1,1,0,cos(t)
L,L1,1,2,0.001,0
T,T1,0,2,2,3,0.5
R,R1,3,4,30
C,C1,4,0,0.5,0
```

فایل خروجی: (چون پاسخ دقیق است طولانی است میتوانید فایل output4.txt را باز کنید و مشاهده کنید)

## نمودار های R1:



## نمودار های T1:



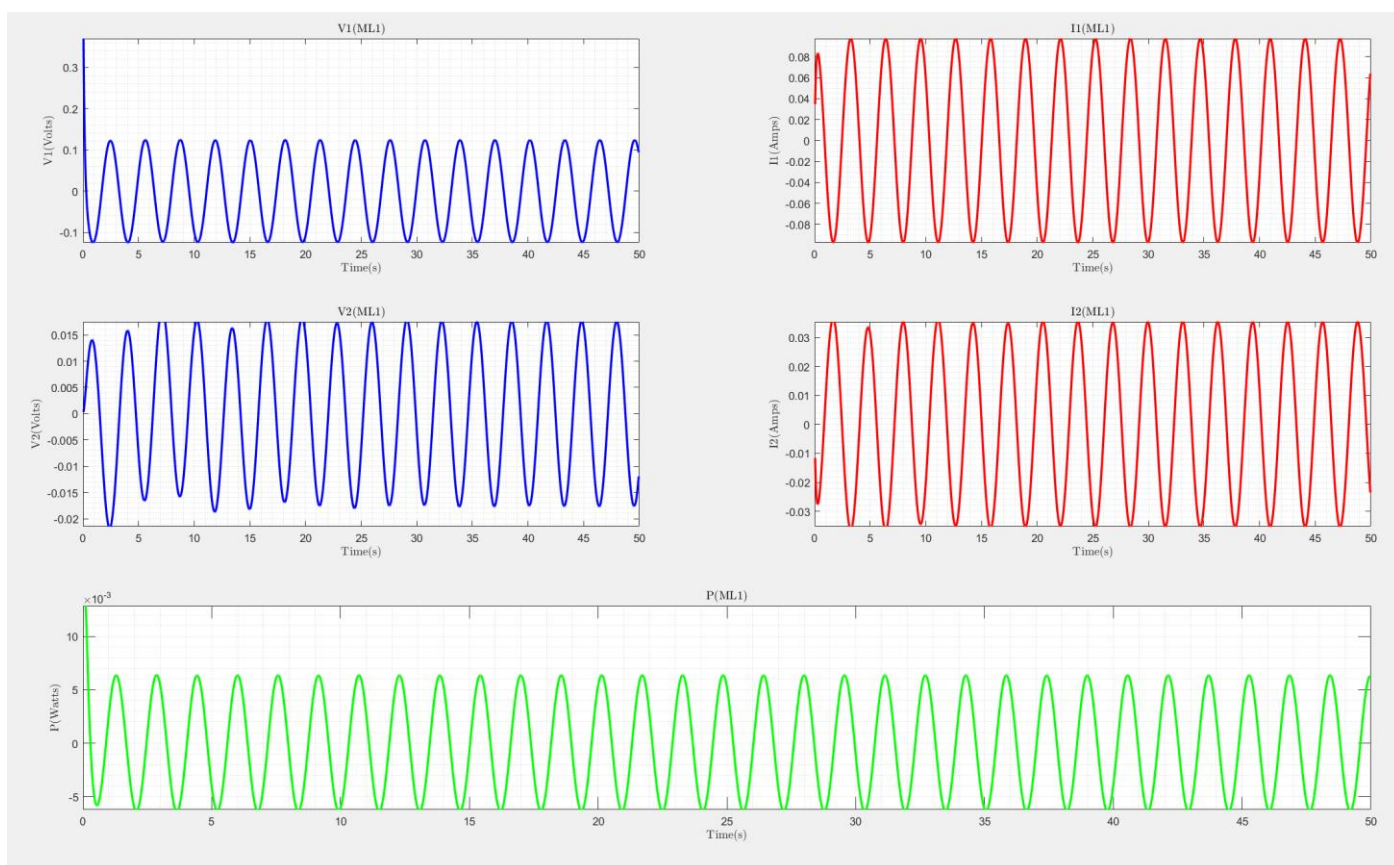
نمونه ۵:

نت لیست:

```
I,I1,0,1,sin(2*t)
L,L3,1,0,0.5,0
R,R1,1,2,10
ML,ML1,2,0,1,3,0,3,1
C,C1,3,0,1,0
R,R2,3,0,5
```

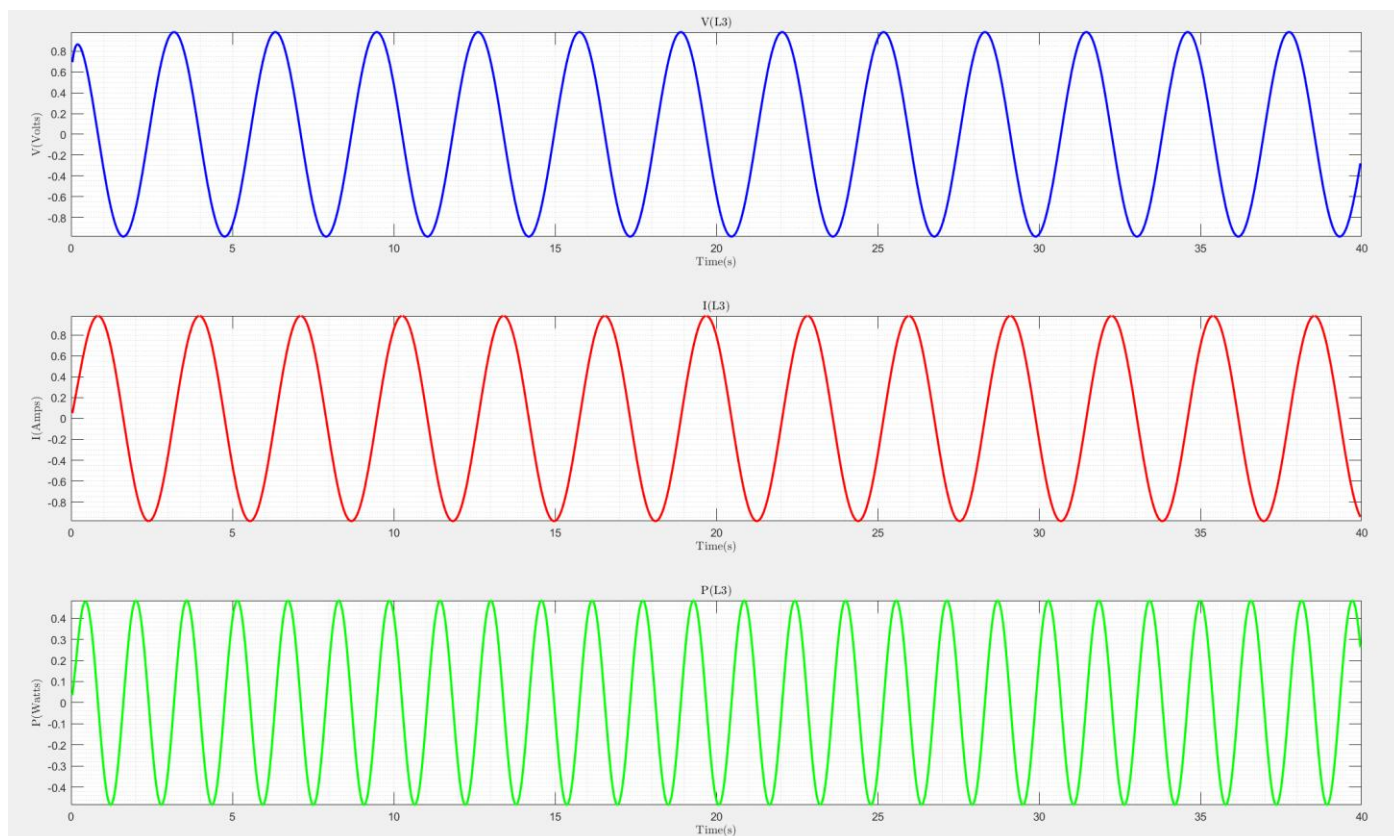
فایل خروجی: (به علت بزرگ بودن نوشته های فایل خروجی عکس را اینجا نمی آورم فایل output5.txt را باز کنید)

نمودار های ML1:

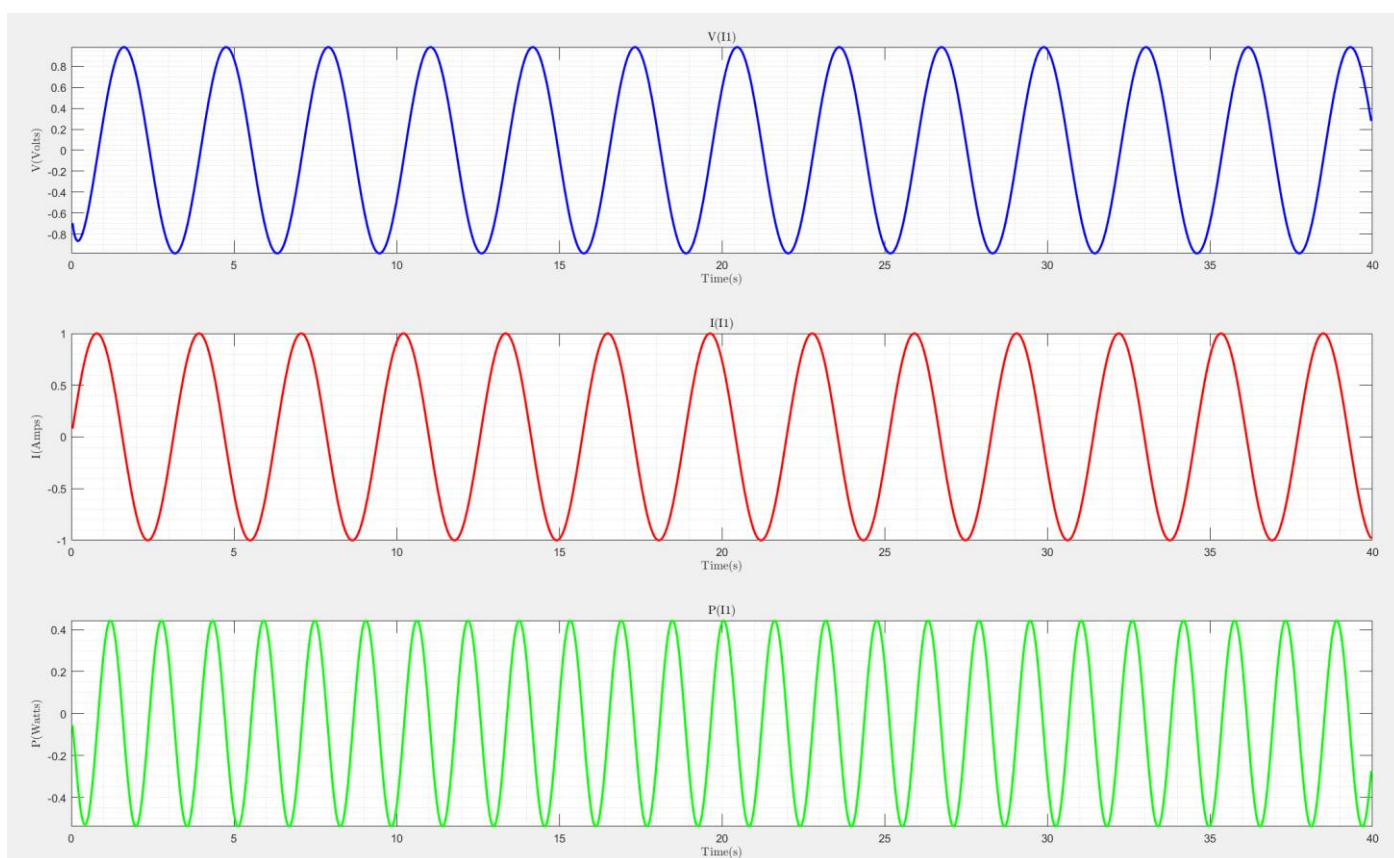




## نمودار های L3:



## نمودار های I1:





نمونه ۶:

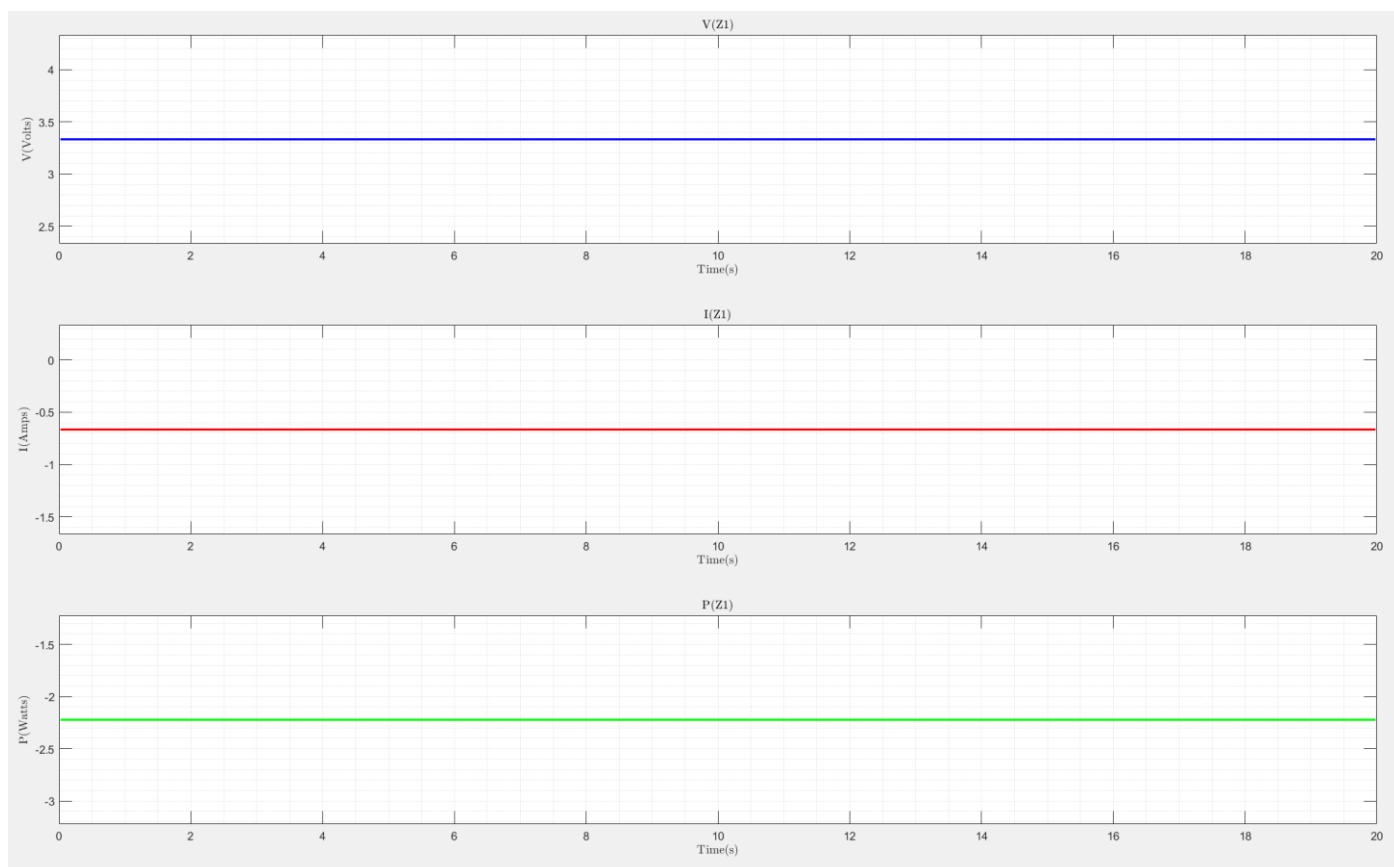
نت لیست:

V,V1,1,0,5  
R,R1,1,2,1  
R,R2,2,0,2  
Z,Z1,3,0,1,2,2  
R,R3,3,0,5

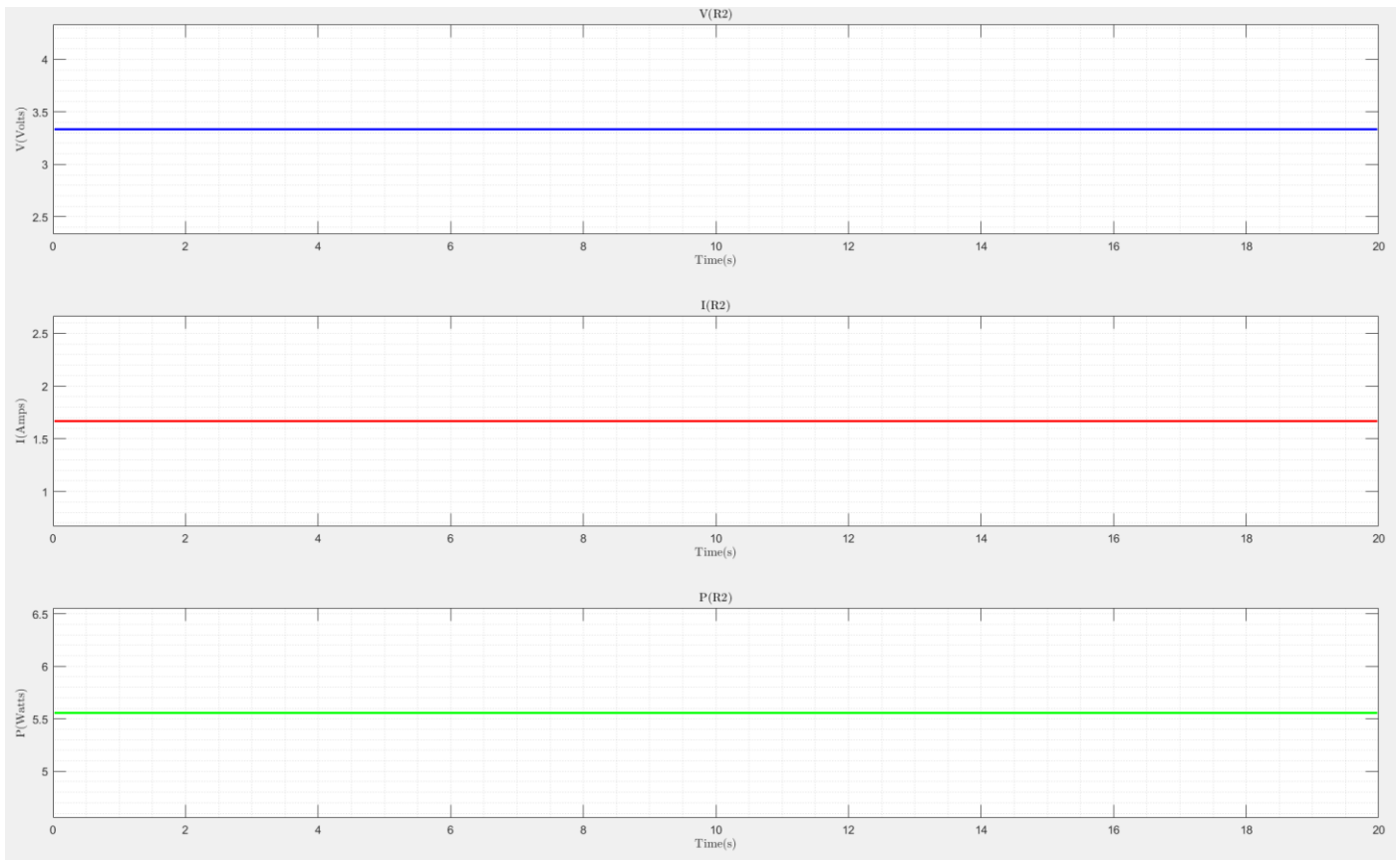
فایل خروجی:

<V1><5.0><-1.7><-8.3>  
<R1><1.7><1.7><2.8>  
<R2><3.3><1.7><5.6>  
<Z1><3.3><-0.67><-2.2>  
<R3><3.3><0.67><2.2>

نمودار های Z1:



نمودار های R2:



نمونه ۷:

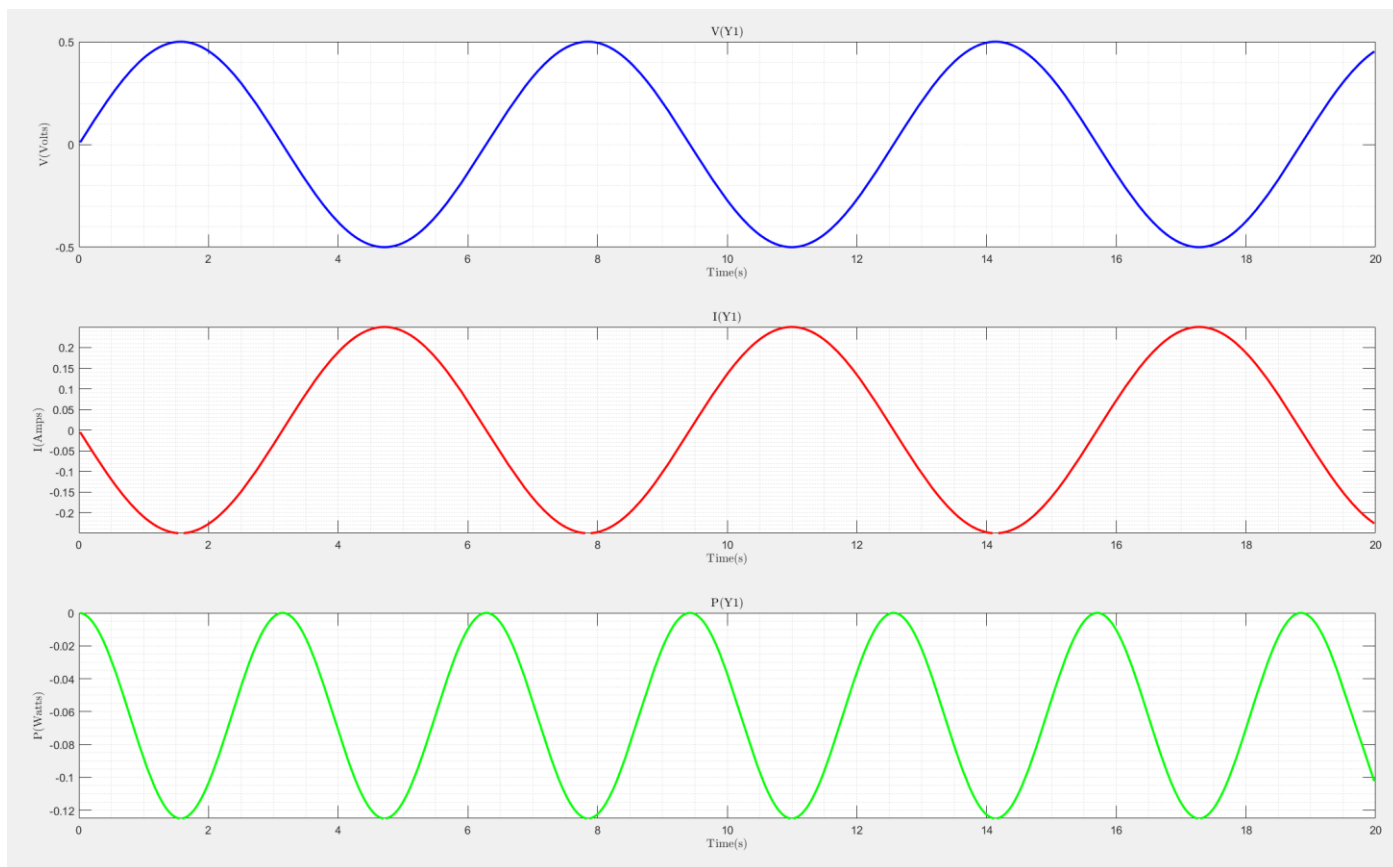
نت لیست:

```
V,V1,1,0,cos(t)
R,R1,1,2,1
R,R2,3,0,1
Y,Y1,4,0,2,3,1/s
R,R3,4,0,2
```

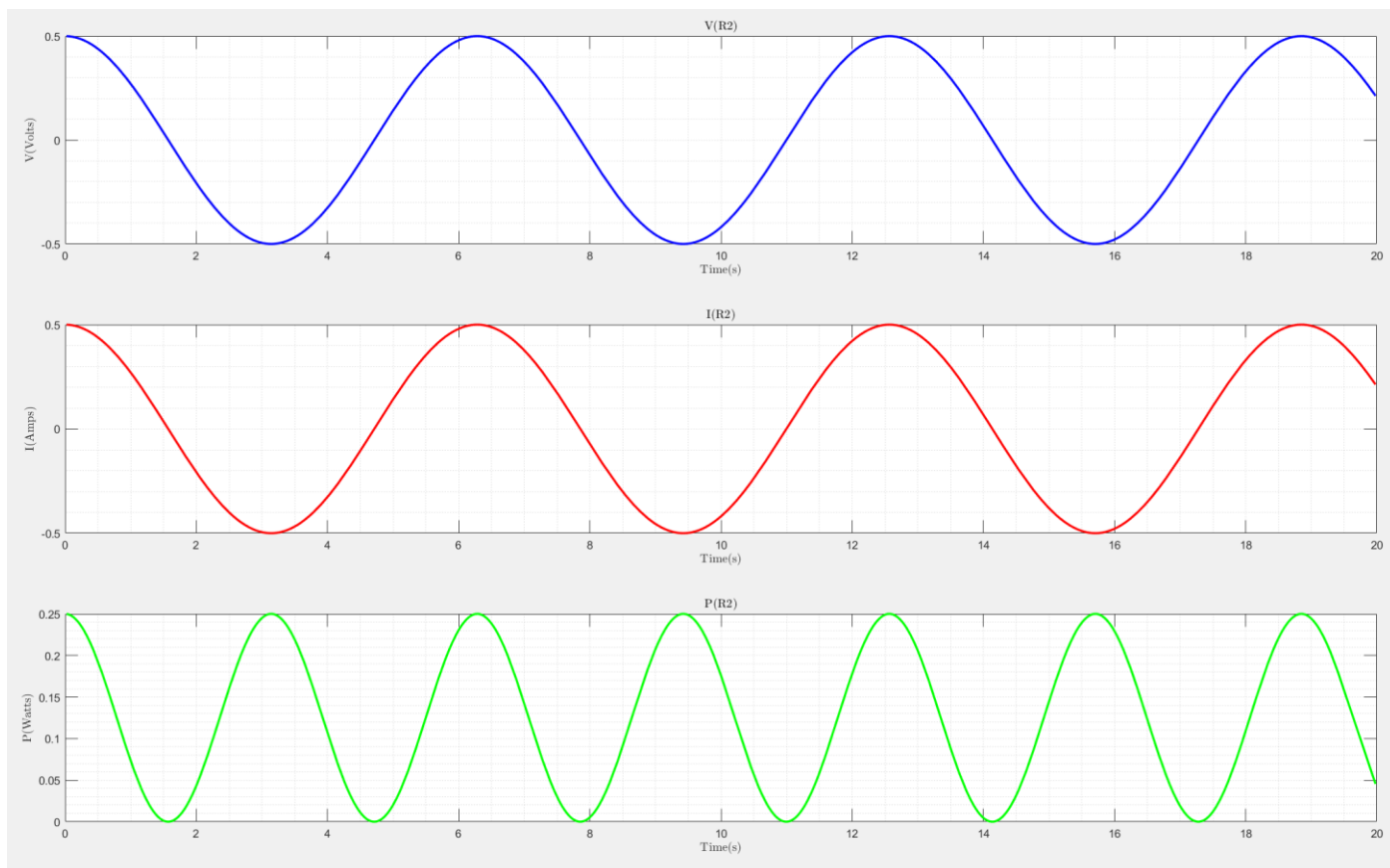
فایل خروجی:

```
<V1><cos(t)><-0.5*cos(t)><-0.5*cos(t)^2>
<R1><0.5*cos(t)><0.5*cos(t)><0.25*cos(t)^2>
<R2><0.5*cos(t)><0.5*cos(t)><0.25*cos(t)^2>
<Y1><0.5*sin(t)><-0.25*sin(t)><-0.12*sin(t)^2>
<R3><0.5*sin(t)><0.25*sin(t)><0.12*sin(t)^2>
```

## نمودار های Y1:



## نمودار های R2:



این کد نرم افزار است که حلقه های آن را collapse کرده ایم تا روند کلی با استفاده از کامنت ها معلوم باشد:

```
1 clc; clear; close all;
2
3 syms t s
4 assume(t, 'real'); % this assumes the t as real value as it is
5 prompt = 'Enter netList file name : ';
6 inputFileName = input(prompt, "s"); % get input file name from command window
7 clc;
8 prompt = 'Enter output file name : ';
9 outputFileName = input(prompt, "s"); % get output file name from command window
10 clc;
11 fileID = fopen(inputFileName);
12 compline = fgetl(fileID);
13 digits(2);
14 elements = {};
15
16 Nodes = [];
17
18 while ischar(compline) % this while loop read the elements and copies them to an array with element type which defined as class
19     fclose(fileID);
20     numOfNodes = max(Nodes, [], 'all');
21
22     A = sym(zeros(numOfNodes)); % create the left hand side matrix of equations
23     B = sym(zeros(numOfNodes, 1)); % create the right hand side matrix of equations
24
25     for n = 1:length(elements) % complete the A matrix regarding to elements which there is no need to know their current directly
26
27     for n = 1:length(elements) % complete the A matrix regarding to elements which we have to know their current directly to solve the node equations
28
29
30     X = linsolve(A, B); % solve the system of equations
31     X = simplify(simplifyFraction(X)); % simplify the expressions of results
32     X = [sym(0); X]; % enter the ground node to the matrix
33     IDN = 1; % number of current dependent elements
34     numOfNodes = numOfNodes + 1;
35
36     for n = 1:length(elements) % in this for we assign the voltage, current and power of each element in elements array
37
38     fileID = fopen(outputFileName, 'w');
39
40     for n = 1:length(elements) % in this for we generate the output text file
41     fclose(fileID);
42
43     disp('output file generated ...');
44     in = ' ';
45     while ~strcmp(in, 'end') % this is the while loop which is used to get input repeatedly from user to plot informations of elements
46
47     clc; close all;
```

این هم کلاس element که آن را تعریف کردیم تا با استفاده از شی گزایی برنامه را بنویسیم (کل کاربردش طبقه بندی و struct مانند است):

```
1 classdef element
2     properties
3         type % type of element
4         name % name of element
5         nodes % nodes of element
6         snodes % source nodes for VCVS, CCVS, VCCS, CCCS
7         value % values needed for element characterization
8         IC % initial condition
9         voltage % voltage of element
10        current % current of element
11        power % the power of element
12    end
13
14    methods
15        function obj = element(type, name, nodes, snodes, value, IC)
16            obj.type = type;
17            obj.name = name;
18            obj.nodes = nodes;
19            obj.snodes = snodes;
20            obj.value = value;
21            obj.IC = IC;
22        end
23    end
24 end
```