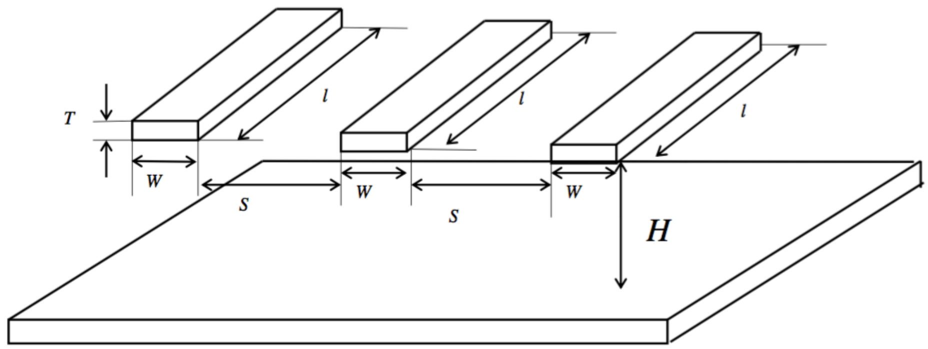
EE201C Homework1 Report

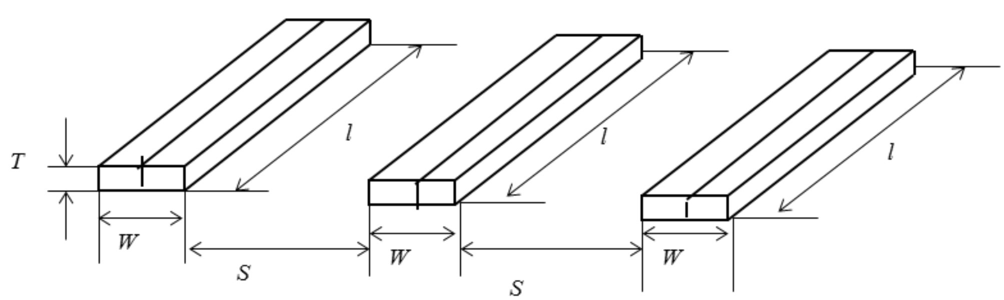
Name: Peidong Chen   UID: 204432674

[1] Given three wires, each modeled by at least 2 filaments, find the 3x3 matrix for (frequency-independent) inductance between the 3 wires, along with the capacitance and resistance. We assume that the ground plane has infinite size and is 10 um away for the purpose of capacitance calculation.



- *Wire width: W=9um, wire thickness: T=6um, wire length: l=9000um,*
- *Wire spacing: S = 15um, distance to ground: H=10um,*
- *Copper electrical resistivity 0.0175 $\Omega mm^2/m$ (room temperature),*
- *$\mu = 1.256\times10^{-6} H/m$,*
- *free space $\varepsilon_0 = 8.85\times10^{-12} F/m$*

Solution:



Discretize the 3 wires into 6 filaments.

For each filament, calculate its self-inductance with:

$$L_{self-L} = \frac{\mu l}{2\pi}\left[\ln\left(\frac{2l}{W'+T}\right)+\frac{1}{2}+\frac{(W'+T)}{4l}\right]$$
$$W' = W/2$$

For each pair of filament, calculate the mutual inductance with:

$$L_{mutual-L} = \frac{\mu l}{2\pi}\left[\ln\left(\frac{2l}{D}\right)-1+\frac{D}{l}\right]$$

With the equations above, I assigned the value of width, thickness and length to the three wires and six filaments respectively. And also assigned the x and y-coordinate to the endpoint of each wire and filament. The Matlab code can be coded like below:

```
%% - Step 2: Inductance Calculation [without ground] -
ind_filament = zeros(6,6);
% - (2.1) self-Inductance of filaments -
u = 1.256e-6;
for i = 1:6
    % TASK: use equation to compute self-inductance for each filament
    ind_filament(i,i) =
(u*filament{i}.length/(2*pi))*(log(filament{i}.length*2/(filament{i}.width+filament{i
}.thickness))+0.5+(filament{i}.width+filament{i}.thickness)/(4*filament{i}.length));
end

% - (2.2) mutual-Inductance of filaments -
for i=1:6
    for j=1:6
        % TASK: use equation to compute mutual-inductance between each pair of
filaments
        if(i~=j)
        ind_filament(i,j) =
(u*filament{i}.length/(2*pi))*(log(filament{i}.length*2/(abs(filament{i}.x-
filament{j}.x))) - 1 + (abs(filament{i}.x-filament{j}.x))/filament{i}.length);
        end
    end
end
```

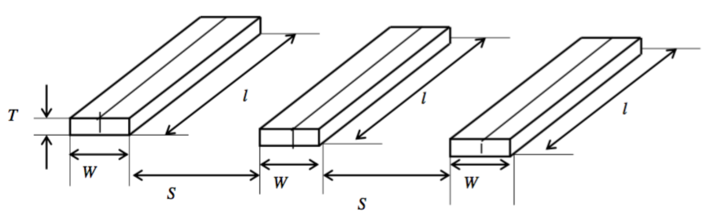
Here is the filament inductance matrix:

1	2	3	4	5	6
1.4297e-08	1.3124e-08	1.0116e-08	9.8075e-09	8.8736e-09	8.7132e-09
1.3124e-08	1.4297e-08	1.0488e-08	1.0116e-08	9.0498e-09	8.8736e-09
1.0116e-08	1.0488e-08	1.4297e-08	1.3124e-08	1.0116e-08	9.8075e-09
9.8075e-09	1.0116e-08	1.3124e-08	1.4297e-08	1.0488e-08	1.0116e-08
8.8736e-09	9.0498e-09	1.0116e-08	1.0488e-08	1.4297e-08	1.3124e-08
8.7132e-09	8.8736e-09	9.8075e-09	1.0116e-08	1.3124e-08	1.4297e-08

The results are shown in the table above. Notice that when i = j (11, 22, 33, etc.), it is self-inductance. And when i!= j, it is mutual-inductance.

Step 1.2

Calculate inductance matrix of three wires



Calculate inductance matrix of three wires with:

$$Lp_{km} = \sum_{i=1}^{P} \sum_{j=1}^{Q} Lp_{ij}$$

I calculated the self-inductance for each wire and the mutual inductance for each pair of wire using the above formulas, and stored the inductance in a three-by-three matrix. The Matlab code can be coded like below:

```
ind_wire = zeros(3,3);
for i=1:3
    for j=1:3
        if (i==j)
        %TASK: use equation to calculate self-inductance of each wire
        ind_wire(i,j) = ind_filament(2*i-1,2*j-
1)+ind_filament(2*i,2*j)+ind_filament(2*i-1,2*j)+ind_filament(2*i,2*j-1);%rref
        else
```

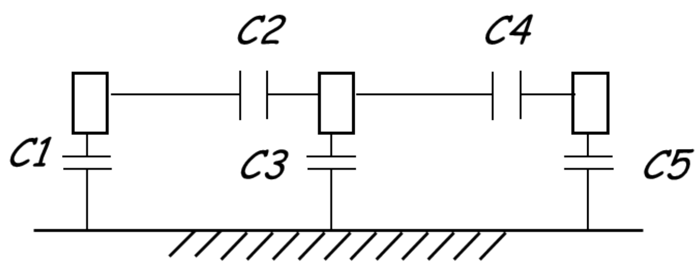
```
        %TASK: use equatoin to calculate mutual-inductance of each pair of wires
        ind_wire(i,j) = ind_filament((i-1)*2+1,(j-1)*2+1)+ind_filament((i-1)*2+1,(j-
1)*2+2)+ind_filament((i-1)*2+2,(j-1)*2+1)+ind_filament((i-1)*2+2,(j-1)*2+2);
        end
    end
end
```

Then the results are shown below:

1	2	3
5.4842e-08	4.0528e-08	3.5510e-08
4.0528e-08	5.4842e-08	4.0528e-08
3.5510e-08	4.0528e-08	5.4842e-08

Step 1.3

Capacitance calculation:

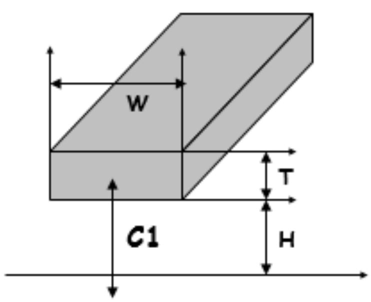


C1 and C5 equals to average of those for the following two cases:

• single wire over ground

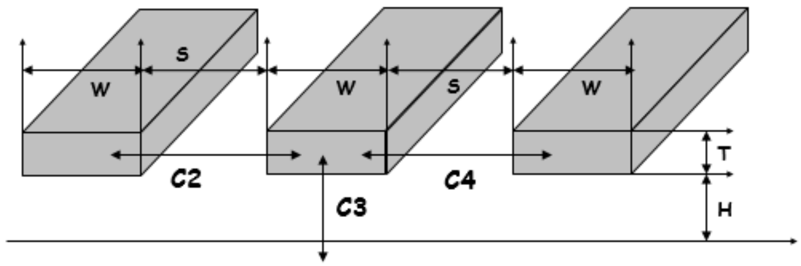
• three parallel wires over ground

According to the reference "Simple Formulas for Two- and Three-Dimensional Capacitances", the capacitance of single line on ground plane is



$$\frac{C_1}{E_{ox}} = 1.15\left(\frac{W}{H}\right) + 2.80\left(\frac{T}{H}\right)^{0.222} \quad (1)$$

where Eox is a dielectric constant of an insulator. And in case of three lines on ground plane, the total capacitance of the middle line includes the coupling capacitance between lines and ground capacitance between the line and the ground. So the total capacitance of the middle line per unit length is



$$\frac{C_2}{E_{ox}} = \frac{C_1}{E_{ox}} + 2\left[0.03\left(\frac{W}{H}\right) + 0.83\left(\frac{T}{H}\right) - 0.07\left(\frac{T}{H}\right)^{0.222}\right]\left(\frac{S}{H}\right)^{-1.34} \quad (2)$$

Since C1 and C5 equals to average of single wire over ground and there parallel wires over ground, C1 = C5 = [C3 + (C3 + 2*C2)]/2 = C2 + C3.

With all the equations given above, the Matlab code can be like below:

```
%% - Step 3: Capcitance Calculation [with ground]  -

% - (3.1) capacitance of signal wire -

% TASK: use equation to calculate the capacitance between signal wire (middle one)
and
% ground

w=9e-6;
t=6e-6;
l=9000e-6;
s=15e-6;
h=10e-6;
e=8.85e-12;

% TASK: use equation to calculate the coupling capacitance
C3  = e*(1.15*(w/h)+2.8*(t/h)^0.222)*l; %single wire
C2  = e*(0.03*(w/h)+0.83*(t/h)-0.07*(t/h)^0.222)*(s/h)^-1.34*l; %coupling capacitance

C4  = C2;

C31 = C3/2;
C32 = C3/2;


% - (3.2) coupling capacitance of edge wires-
```

```
% TASK: use equation to calculate capcitance between edge wire and ground
C1  = C2+C3;
C5  = C1;
C11 = C1/2;
C12 = C1/2;
C51 = C5/2;
C52 = C5/2;
```

Here is the result:

(1) Ground capacitance of middle wire:

C3=e*(1.15*(width/h)+2.8*(thickness/h)^0.222)*length =2.815e-13 F;

(2) Coupling capacitance of middle wire:

C2=C4=e*(0.03*(width/h)+0.83*(thickness/h)-0.07*(thickness/h)^0.222)*(s/h)^-1.34*length = 2.140e-14 F;

(3) Ground capacitance of edge wires:

C1=C5=(single wire + three wire)/2 = 3.029e-13 F;

Step 1.4

We know that Copper electrical resistivity 0.0175 $\Omega mm^2/m$ (room temperature), also the equation:

$$R = \rho \frac{l}{A}$$

I calculated the resistance for each wire and stored them in a vector.

Here is my matlab code:

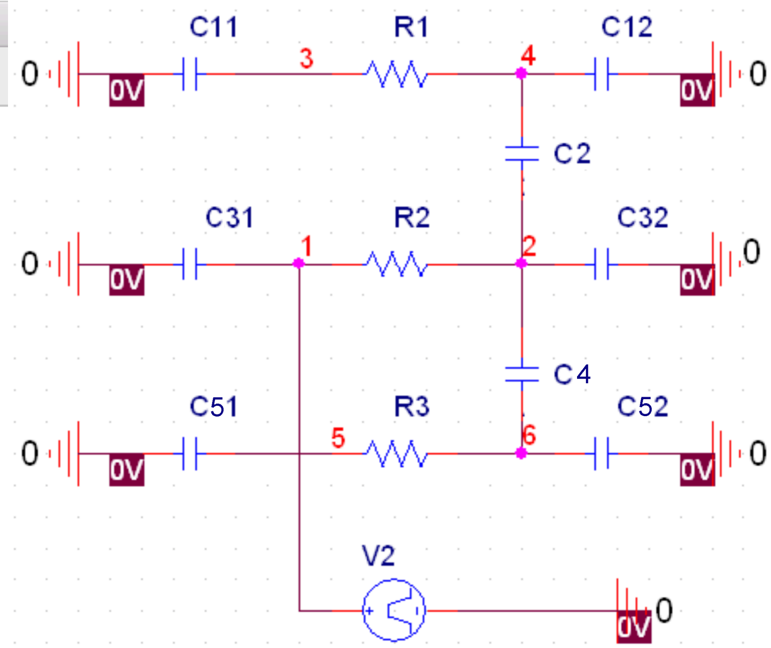
```
%% - Step 4: Resistance Calculation
r_wire = zeros(3,1);
for i=1:3
    % TASK: calculate R for each wire: r_wire(i)
    r_wire(i) = 0.0175*l/w/t/10e5;
end
```

The result is:

R1 = R2 = R3 = 2.9167 Ω

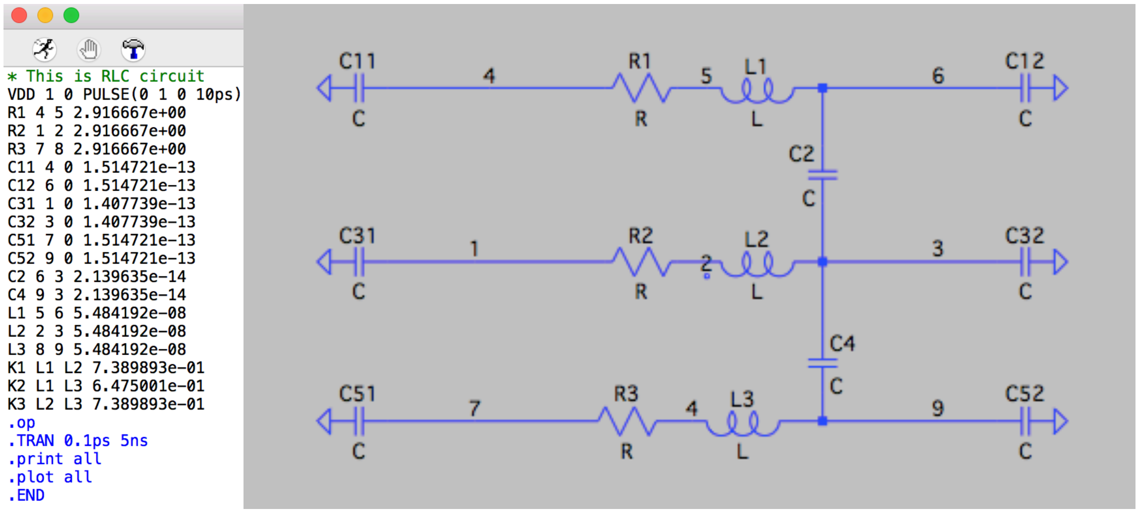
[2] Build the RC and RCL circuit models in SPICE netlist for the above wires. (suggest to use Matlab script to generate matrix and thus SPICE netlist)

```
*This is RC circuit
VDD 1 0 PULSE(0 1 0 20ns)
R1 3 4 2.916667e+00
R2 1 2 2.916667e+00
R3 5 6 2.916667e+00
C11 3 0 1.514721e-13
C12 4 0 1.514721e-13
C31 1 0 1.407739e-13
C32 2 0 1.407739e-13
C51 5 0 1.514721e-13
C52 6 0 1.514721e-13
C2 4 2 2.139635e-14
C4 6 2 2.139635e-14
.op
.TRAN 0.1ps 50ns
.print all
.plot all
.END
```



I wrote all R and C components into the Spice Input file and generated RC netlist for SPICE. Here is my matlab code for building the RC circuit model:

```
%% - Step 5: Generate RC and RCL netlist for SPICE -
fid1 = fopen('rc.sp', 'w');
fprintf(fid1, '*This is RC circuit\n');
%TASK: write all R and C component into the Spice Input file
% fprintf(fid1, 'VDD 1 0 PULSE(0 1 0 10ps)\n');
fprintf(fid1, 'VDD 1 0 PULSE(0 1 0 20ns)\n');
fprintf(fid1, 'R1 3 4 %e\n',r_wire(1));
fprintf(fid1, 'R2 1 2 %e\n',r_wire(2));
fprintf(fid1, 'R3 5 6 %e\n',r_wire(3));
fprintf(fid1, 'C11 3 0 %e\n',C11);
fprintf(fid1, 'C12 4 0 %e\n',C12);
fprintf(fid1, 'C31 1 0 %e\n',C31);
fprintf(fid1, 'C32 2 0 %e\n',C32);
fprintf(fid1, 'C51 5 0 %e\n',C51);
fprintf(fid1, 'C52 6 0 %e\n',C52);
fprintf(fid1, 'C2 4 2 %e\n',C2);
fprintf(fid1, 'C4 6 2 %e\n',C4);
fprintf(fid1, '.op\n');
% fprintf(fid1, '.TRAN 0.1ps 30ps\n');
fprintf(fid1, '.TRAN 0.1ps 50ns\n');
fprintf(fid1, '.print all\n');
fprintf(fid1, '.plot all\n');
fprintf(fid1, '.END\n');
fclose(fid1);
```



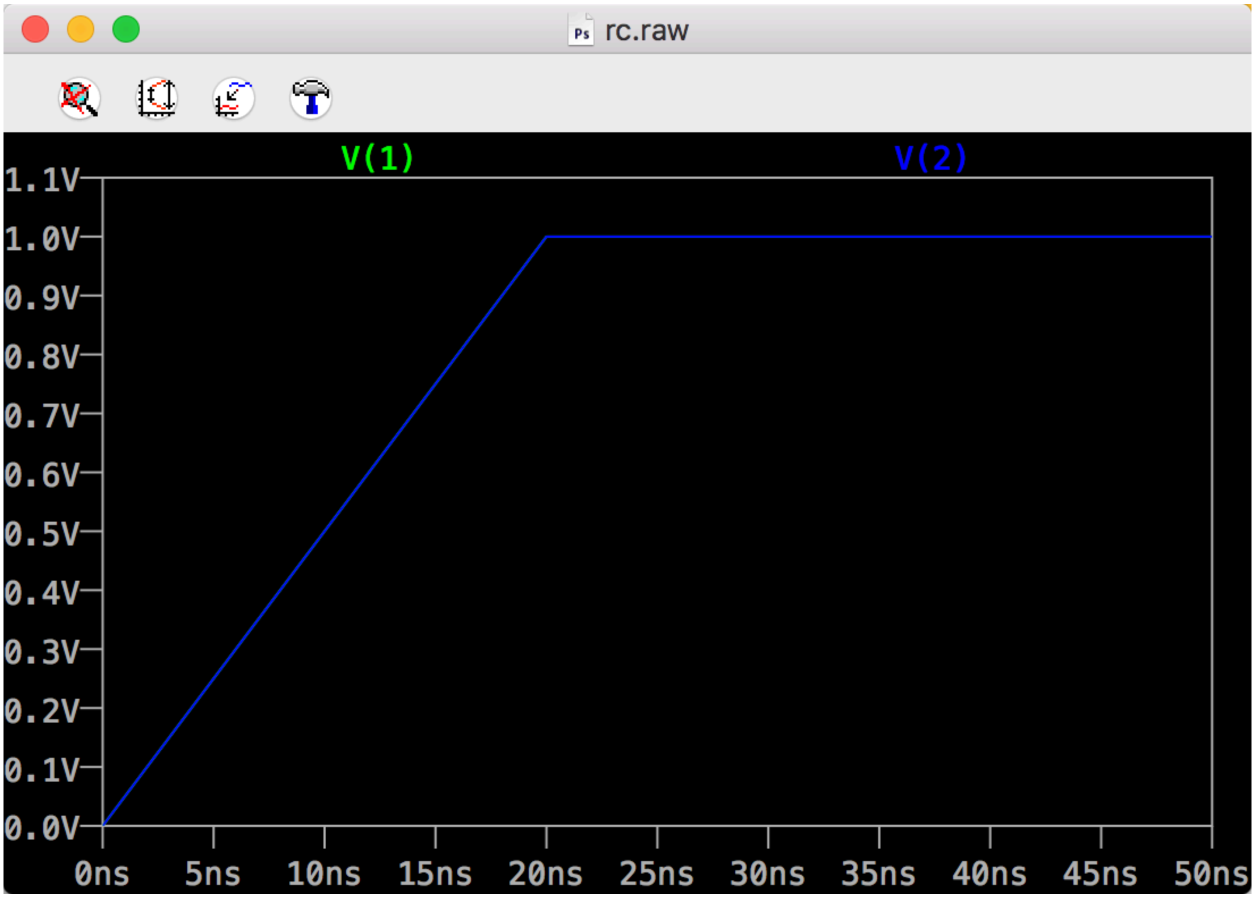
Then I calculated the transconductance between each wire and wrote all R, L, C components and the transconductance into the Spice Input file, then generated RLC netlist for SPICE. Here is my matlab code for building the RLC circuit model:

```
fid2 = fopen('rlc.sp', 'w');
fprintf(fid2, '* This is RLC circuit\n');
% TASK: write all R, L and C component into the Spice Input file
fprintf(fid2, 'VDD 1 0 PULSE(0 1 0 10ps)\n');
%fprintf(fid2, 'VDD 1 0 PULSE(0 1 0 20ps)\n');
fprintf(fid1, 'R1 4 5 %e\n',r_wire(1));
fprintf(fid1, 'R2 1 2 %e\n',r_wire(2));
fprintf(fid1, 'R3 7 8 %e\n',r_wire(3));
fprintf(fid1, 'C11 4 0 %e\n',C11);
fprintf(fid1, 'C12 6 0 %e\n',C12);
fprintf(fid1, 'C31 1 0 %e\n',C31);
fprintf(fid1, 'C32 3 0 %e\n',C32);
fprintf(fid1, 'C51 7 0 %e\n',C51);
fprintf(fid1, 'C52 9 0 %e\n',C52);
fprintf(fid1, 'C2 6 3 %e\n',C2);
fprintf(fid1, 'C4 9 3 %e\n',C4);
fprintf(fid1, 'L1 5 6 %e\n',ind_wire(1,1));
fprintf(fid1, 'L2 2 3 %e\n',ind_wire(1,1));
fprintf(fid1, 'L3 8 9 %e\n',ind_wire(1,1));
fprintf(fid1, 'K1 L1 L2 %e\n',K1);
fprintf(fid1, 'K2 L1 L3 %e\n',K2);
fprintf(fid1, 'K3 L2 L3 %e\n',K3);
fprintf(fid2, '.op\n');
fprintf(fid2, '.TRAN 0.1ps 5ns\n');
fprintf(fid2, '.print all\n');
fprintf(fid2, '.plot all\n');
fprintf(fid2, '.END\n');
fclose(fid2);
```

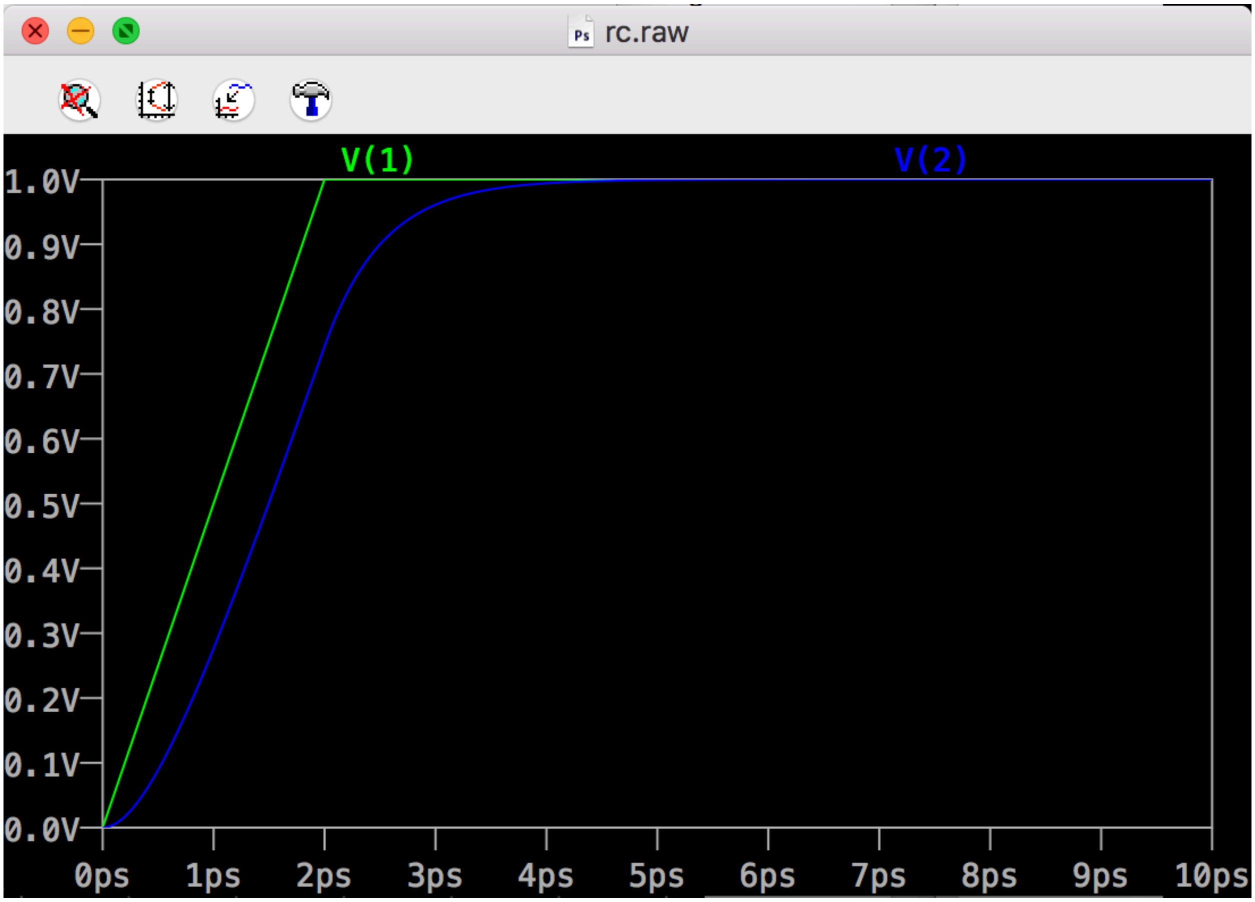
[3] Assume a step function applied at end-end, compare the four waveforms at the far-end for the central wire using SPICE transient analysis for (a) RC and RLC models and (b) rising time is 20ns, or try to use longer rising time.

I generated four waveforms at V1 and V2 for RC model and V1 and V3 for RLC model using SPICE transient analysis with different rising time. Here is the result:

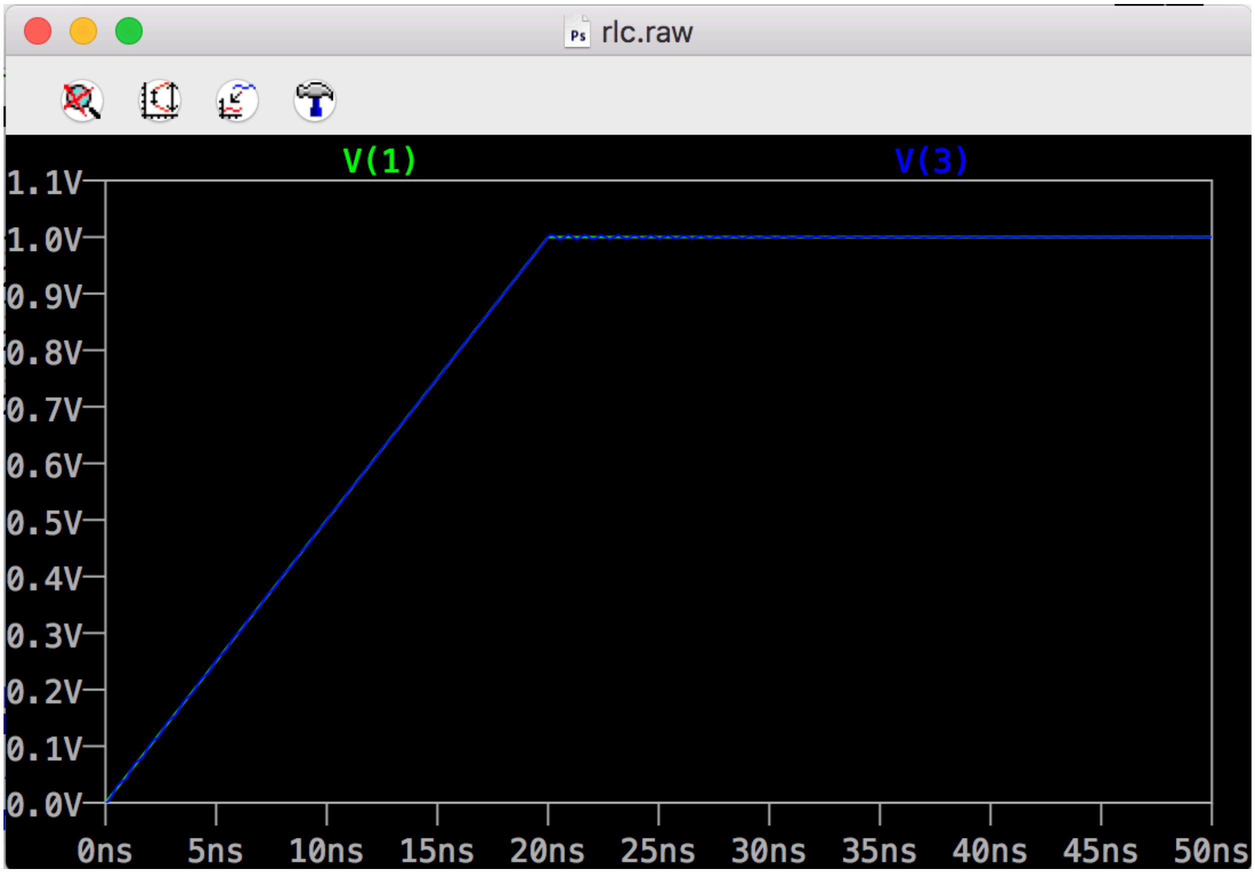
(1) Result of RC model with rising time of 20ns:



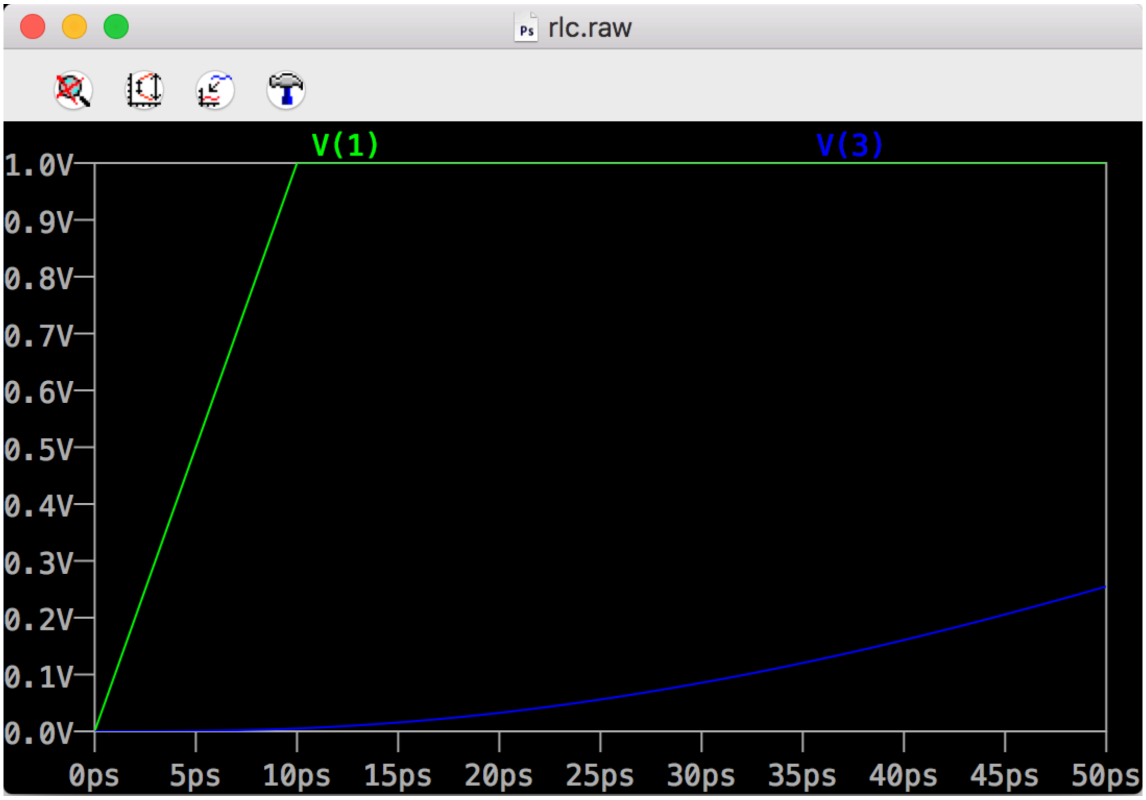
(2) Result of RC model with rising time of 2ps:

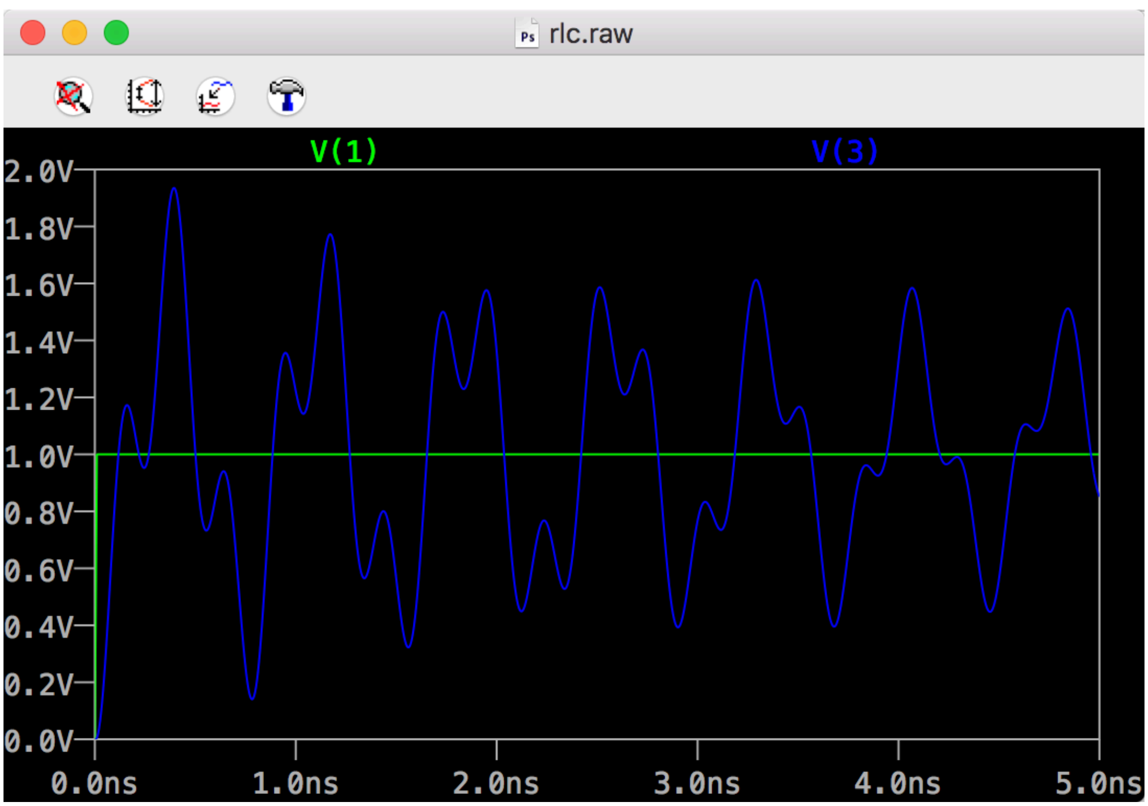


(3) Result of RLC model with rising time of 20ns:



(4) Result of RLC model with rising time of 10ps:


rlc.raw
V(1)
V(3)
1.0V
0.9V
0.8V
0.7V
0.6V
0.5V
0.4V
0.3V
0.2V
0.1V
0.0V
0ps
5ps
10ps
15ps
20ps
25ps
30ps
35ps
40ps
45ps
50ps


rlc.raw
V(1)
V(3)
2.0V
1.8V
1.6V
1.4V
1.2V
1.0V
0.8V
0.6V
0.4V
0.2V
0.0V
0.0ns
1.0ns
2.0ns
3.0ns
4.0ns
5.0ns