Aircraft Communication Transceiver

EE310 Spring 2020 Project
Professor Dorr

Ethan Nagelvoort	821234668
Karl Horcasitas	821630141
Giemer Lozares	820172243

Introduction

This project is based on an aircraft communication transceiver and gave us a look into the real world of electrical engineering. This is probably a simplified circuit, but we are glad to have seen the experience. As for LTSpice, since our group is also taking EE330, we have already had experience with LTSpice due to our previous project. However, we learned a few new tricks such as learning how to do mutual inductance using the K operative. In order to solve the current matrices in MATLAB, we had to use the math function linsolve() using the standard format AX = B. In our case, our X matrices hold our current values from i_1 to i_7 ($X = A^{-1}B$). We also made use of the abs() and sqrt() functions to calculate the real parts of our imaginary numbers, as well as the round() function to round our RMS values to two decimal places. Our Matlab script is written in a way such that some variables, such as $V_1$and V_{th}, are reused for additional calculations. The equations we used throughout the project were equations presented in lecture. These mainly included Thevenin, impedance, and power equations. There were also a lot of conversions between rectangular form and polar form, mainly to either get Vpeak, Ipeak, Vrms, or Irms.

Part 1: Matrix Equation and Raw Mesh Equations

The mesh equations we found to create a matrix are as follows.

- Mesh i_1: $V_1 + V_2 = i_1(Z_1 + Z_2 + Z_3) - Z_3 i_3$
- Mesh i_2: $0 = i_2(Z_3 + Z_4 + Z_5 + Z_6) - Z_3 i_1 - Z_6 i_3$
- Mesh i_3: $0 = V_{T2} - V_{T1} = Z_6(i_3 - i_2) + Z_7(i_3 - i_4)$, $V_{T2} - V_{T1} = 0$ since they equal each other, as stated in the project handout.
- Mesh i_4: $0 = i_4(Z_7 + Z_8 + Z_9) - Z_7 i_3 - Z_9 i_5$
- Mesh i_5: $0 = i_5(Z_9 + Z_{10} + Z_{11}) - Z_9 i_4 - Z_{11} i_6$
- Mesh i_6: $0 = i_6(Z_{11} + Z_{12} + Z_{13}) - Z_{11} i_5 - Z_{13} i_7$
- Mesh i_7: $-V_R = i_7(Z_{13} + Z_{14}) - Z_{13} i_6$

Now, using these mesh equations we were able to complete the required matrix.

$$\begin{bmatrix} (Z_1+Z_2+Z_3) & -Z_3 & 0 & 0 & 0 & 0 & 0 \\ -Z_3 & (Z_3+Z_4+Z_5+Z_6) & -Z_6 & 0 & 0 & 0 & 0 \\ 0 & -Z_6 & (Z_6+Z_7) & -Z_7 & 0 & 0 & 0 \\ 0 & 0 & -Z_7 & (Z_7+Z_8+Z_9) & -Z_9 & 0 & 0 \\ 0 & 0 & 0 & -Z_9 & (Z_9+Z_{10}+Z_{11}) & -Z_{11} & 0 \\ 0 & 0 & 0 & 0 & -Z_{11} & (Z_{11}+Z_{12}+Z_{13}) & -Z_{13} \\ 0 & 0 & 0 & 0 & 0 & -Z_{13} & (Z_{13}+Z_{14}) \end{bmatrix} * \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \\ i_6 \\ i_7 \end{bmatrix} = \begin{bmatrix} V_1+V_2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -V_R \end{bmatrix}$$

Figure 1.1 Matrix of Mesh Equations

Part 2 : Power Dissipated at Z_{14} with $V_1 = V_2 = 1$ V Peak

After putting the previous matrix into Matlab with V_1 and V_2 equal to 1 and V_R equal to 0, we got an i_7 of 0.0195483326158642 + 0.0157926254381107i A. After converting to polar, we got i_7 to equal **0.025131 ∠ 38.9339**. Then, we used this power equation to calculate the power at Z_{14}:

$$\text{Power dissipated at } Z_{14} = .5 \cdot 50 \cdot |i_{7m}|^2 = .5 \cdot 50 \cdot |0.025131|^2 = 0.015789\ W$$

Part 3: Required Input Voltage for 80 W Output

We first found what i_7 should be when Z_{14} is dissipating 80 W. We used the following equation:

$$80 = .5 \cdot 50 \cdot |i_{7m}|^2$$

Using that equation, we found i_{7m} should equal 1.78885 A so that Z_{14} dissipates 80 W. Then we increased V_1 and V_2 until i_{7m} had a close value of 1.78885 A. Hence, we would increase the V_1 and V_2 multiple times and for each time convert the i_7 we got into polar form and see if it was close to 1.78885 amps. We got i_{7m} equal to **1.78879 A** after increasing our V_1 and V_2 to **71.180 V**. When putting 1.78879 A into the power equation, $.5 \cdot 50 \cdot |1.78879|^2$, we got a power of **79.9942 W**.

Part 4: SPICE Schematic Printout

Spice schematic for when V_1 and V_2 equal 71.180 V:

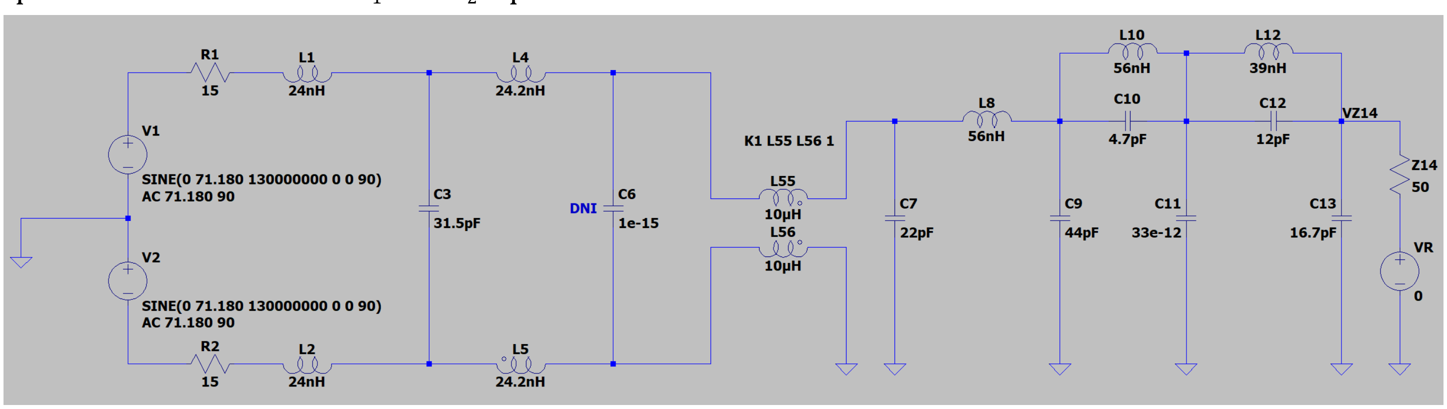

Figure 4.1 LTSpice Circuit with Computed V1 and V2

Part 5: Voltage Plots Across C_7 and Z_{14}

This is the voltage plot for C_7 after transient has died out, using the Spice schematic for when V_1 and V_2 equal **71.180 V**:

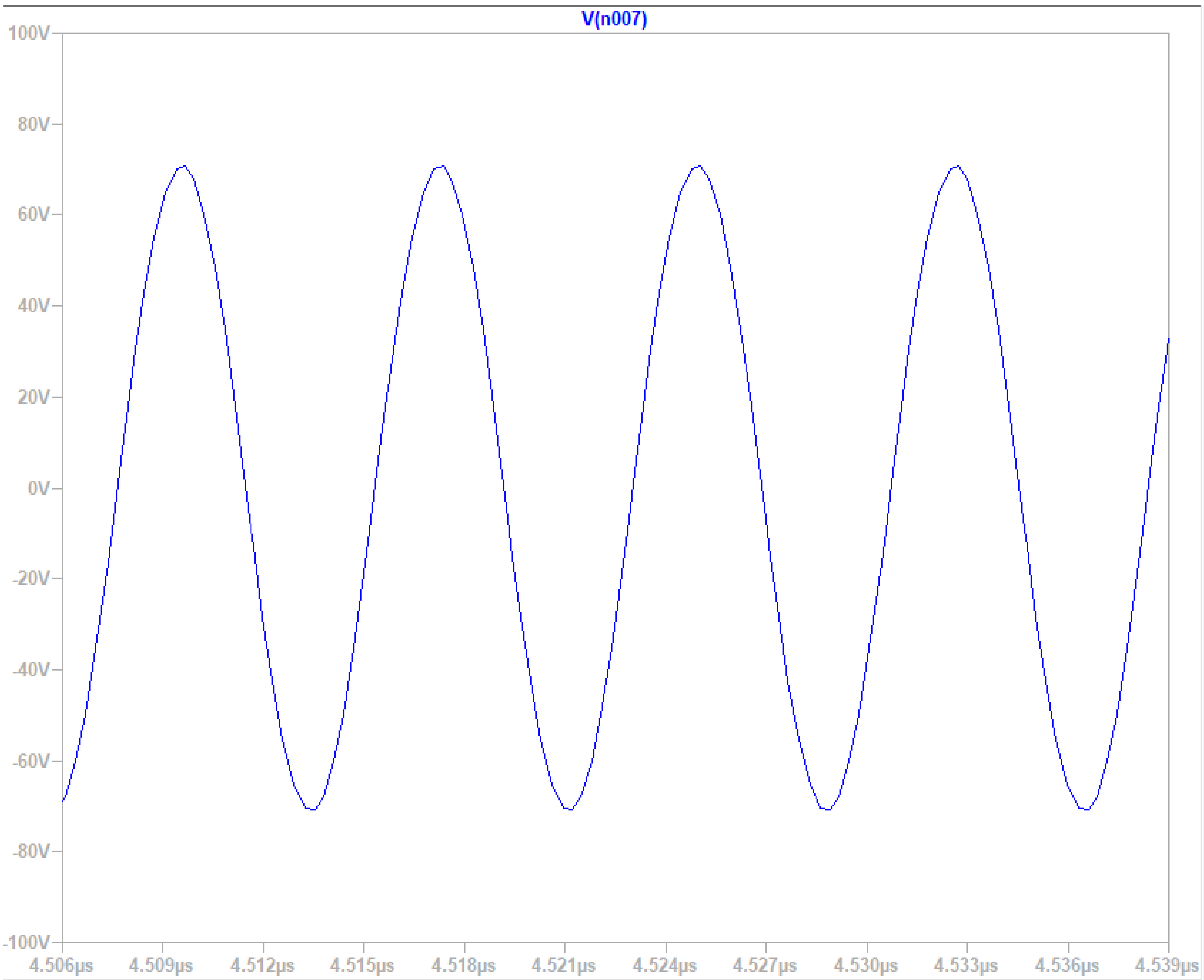

Figure 5.1 Voltage Across C7 Using Calculated V1 and V2

This is the voltage for Z_{14} after transient has died out, using the Spice schematic for when V_1 and V_2 equals **71.180 V**:

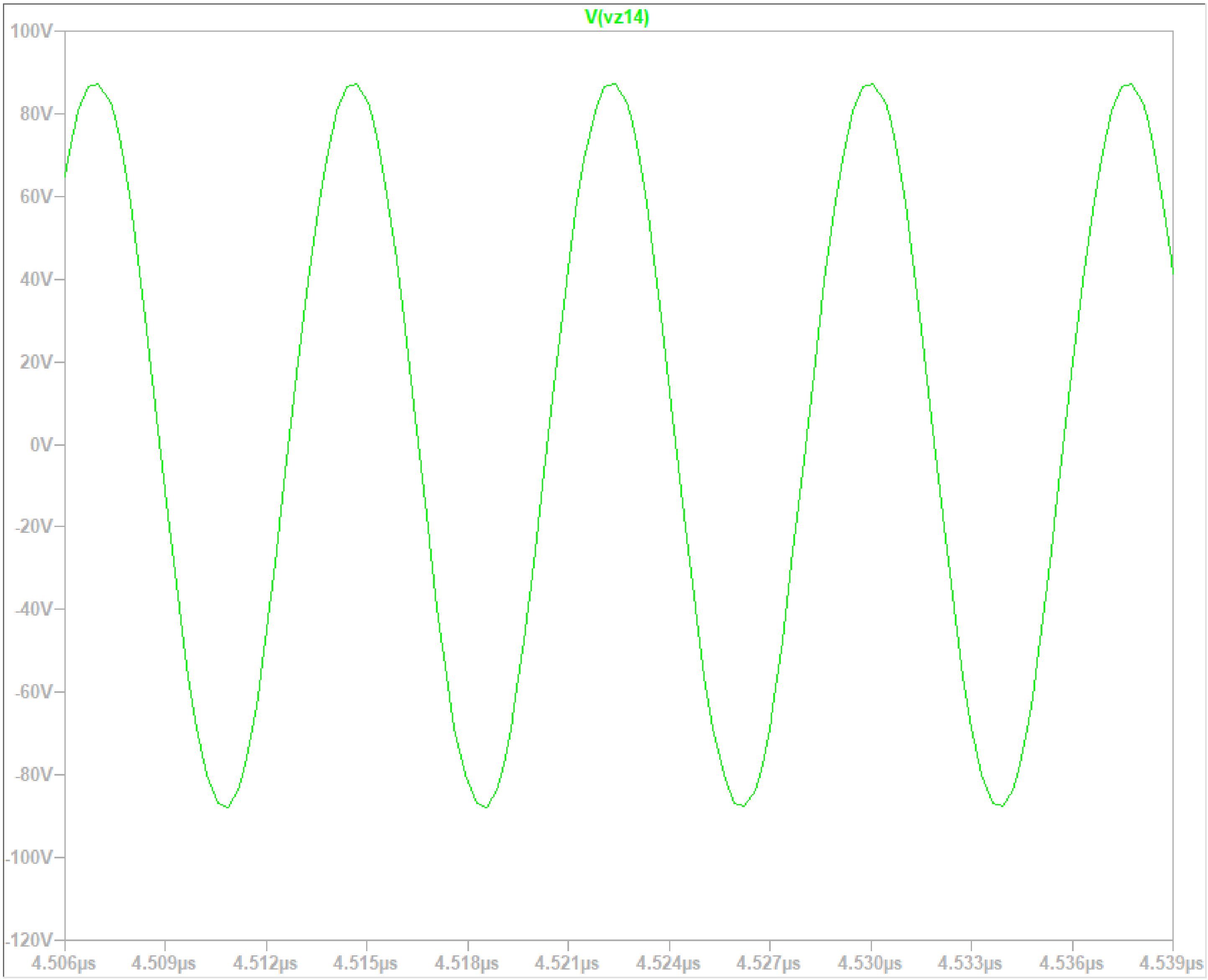

Figure 5.2 Voltage Across Z14 Using Calculated V1 and V2

Part 6: Table with Component Currents and Voltages (Irms and Vrms)

We calculated each component's Vrms and Irms by using Matlab to get each component’s voltage and current values and converting them into pola. We then took the real part of their polar values and divided them by the square root of two. We also rounded each value to two decimal places.

Component	Irms	Vrms
R1	1.31 A	19.59 V
L1	1.31 A	25.61 V
R2	1.31 A	19.59 V
L2	1.31 A	25.61 V
C3	2.17 A	84.42 V
L4	1.59 A	31.37 V
L5	1.59 A	31.37 V
C6	0.00 A	50.65 V
C7	0.91 A	50.65 V
L8	1.91 A	87.16 V
C9	2.60 A	72.26 V
L10	1.85 A	84.51 V
C10	0.32 A	84.51 V
C11	1.70 A	63.04 V
L12	2.23 A	70.92 V
C12	0.70 A	70.92 V
C13	0.86 A	63.24 V
Z14	1.26 A	63.24 V

Table 6.1 RMS Current and Voltage of All Components

Part 7: Thevenin Source Model

We first set V_1 and V_2 to 0 and V_R to 1. We then found the voltage drop at C_7 since that will be the voltage connected to the “pickoff point”. That point is our load, so finding its voltage will give the V_{th}. Through Matlab, we were able to create another matrix with the new values for V_1, V_2, and V_R. We then solved for the new currents.

```
v1b=0; v2b=v1b;
%with vR=1(cos(2pi * 130MHz * t), vR=1
vRb=1;

Ath = [(z1+z2+z3) (-z3) 0 0 0 0 0; %Matrix for Thevin Voltage
     (-z3) (z3+z4+z5+z6) (-z6) 0 0 0 0;
     0 (-z6) (z6+z7) (-z7) 0 0 0;
     0 0 (-z7) (z7+z8+z9) (-z9) 0 0;
     0 0 0 (-z9) (z9+z10+z11) (-z11) 0;
     0 0 0 0 (-z11) (z11+z12+z13) (-z13);
     0 0 0 0 0 (-z13) (z13+z14)];
Bth = [(v1b+v2b);0;0;0;0;0;(-vRb)];
Xth = linsolve(Ath,Bth);

%Vth = i3-i4 * Z7
Vth = (Xth(3,1)-Xth(4,1))*z7;
```

Then to find V_{th}, we computed $Z_7(i_3 - i_4)$, which gave us a value of (**-0.311890115471633 + 0.375749596415572i**) V. We then found I_N, which is the current through the load. We did this by setting C_7 to a very big value (0.01 microfarad), then finding the current through that component. We then computed the new Z_7 and created a whole new matrix in Matlab with the purpose of finding the new currents with this altered Z_7.

```
%make C7 so huge so that it behaves like a short
z7th = 1/(j * w * 0.01*10^-6);
Ath2 = [(z1+z2+z3) (-z3) 0 0 0 0 0; %Matrix for Norton Current
    (-z3) (z3+z4+z5+z6) (-z6) 0 0 0 0;
    0 (-z6) (z6+z7th) (-z7th) 0 0 0;
    0 0 (-z7) (z7th+z8+z9) (-z9) 0 0;
    0 0 0 (-z9) (z9+z10+z11) (-z11) 0;
    0 0 0 0 (-z11) (z11+z12+z13) (-z13);
    0 0 0 0 0 (-z13) (z13+z14)];

Xth2 = linsolve(Ath2,Bth);

Inorton = (Xth2(3,1) - Xth2(4,1)); %Inorton = i3-i4 (the current of C7)

Zth = Vth/Inorton; %our Thevinin impedance Vth/Inorton
```

We then computed $i_3 - i_4$ which gave us the current going through C_7 . This current is going through the "pickoff point" as well, so this will be our I_N. For I_N we got a value of (**-0.014906433715352 + 0.019982858227671i**) A. Finally, to get the Z_{th}, we computed $\frac{V_{th}}{I_N}$ and got (**19.561383192915270 + 1.015853352636761i**) Ω. Hence, this is our thevenin circuit.

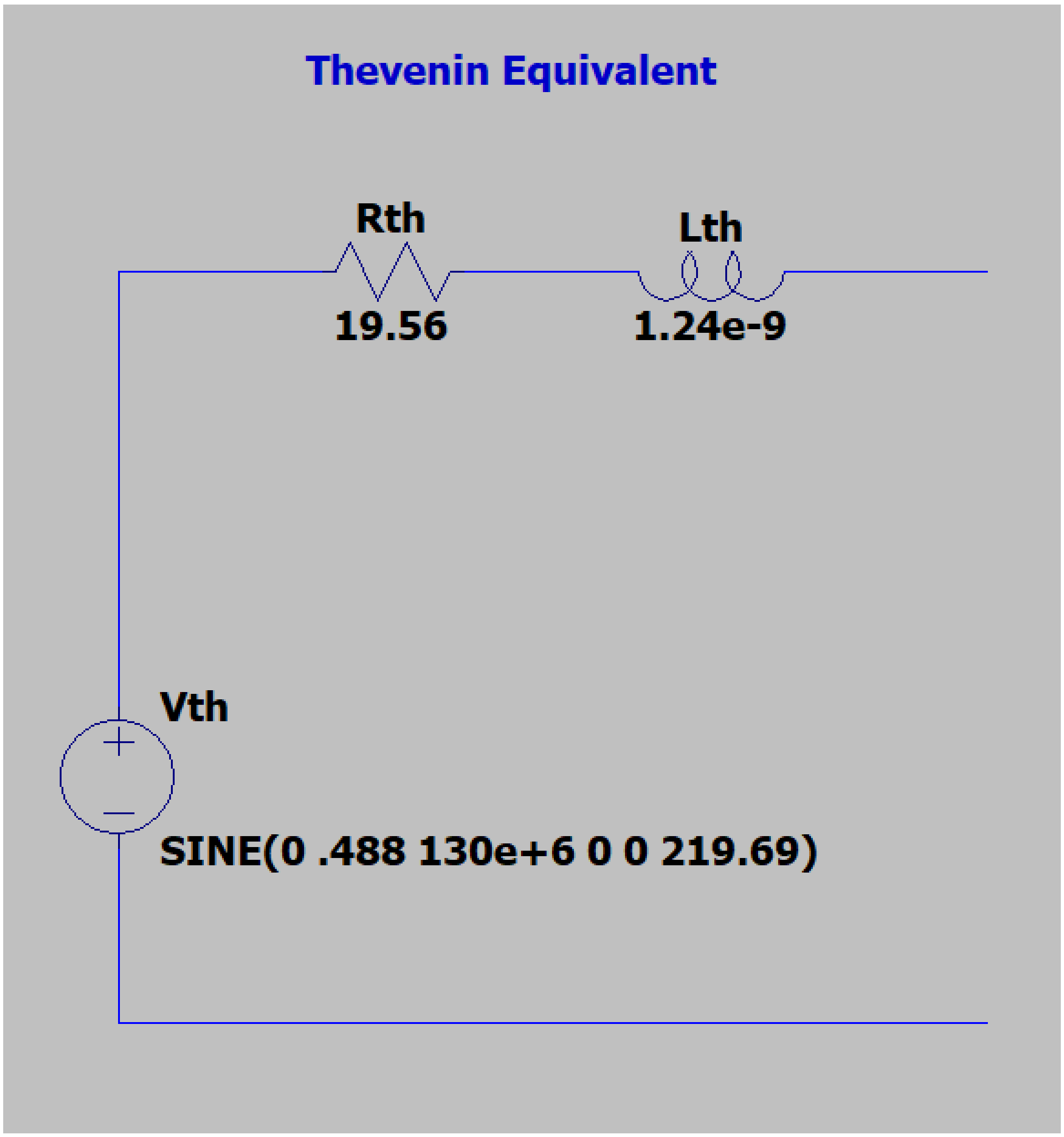

Figure 7.1 Thevenin Equivalent Circuit

Part 8: Max Avail. Source Power & Load for Max Power Transfer

To calculate the maximum available source power, we first take the complex conjugate of our thevenin impedance. With a Zth of (**19.561383192915270 + 1.015853352636761i**) Ω, our complex conjugate is (**19.561383192915270 - 1.015853352636761i**) Ω. We then used the resistive part of our impedance to calculate the max power using equation 11.20 from our textbook:

$$P_{max} = \frac{|\mathbf{V}_{Th}|^2}{8R_{Th}} \qquad \textbf{(11.20)}$$

Using this equation we got a maximum power of 0.001523812203745 or **1.523mW**.

```
%the conjugate of our thevenin impedance
ZthConj = 19.561400245825563 - 1.015825791894316i;
Rth = 19.561400245825563; %the resistive part of our thevivin impedance
%using eq. 11.20 in textbook
%Pmax = abs(Vth^2)/8Rth
PmaxTh = ((abs(Vth))^2)/(8*Rth);
```

The load for max power transfer is found with our conjugate impedance. Since the imaginary part of our impedance is negative, this will be a capacitor. We calculated this in Matlab by calculating 1 divided by the product of our imaginary part, j and w (130 MHz · 2π), which equated to 1.205195617972604e-09 or **1.205nF**.

```
%Complex part of Zth
LforZth = 1.015825791894316i/(i*w); %our thevinin impedance as an inductor
CforZth = 1/(-1.015825791894316i * i * w); %as a capacitor (for our conjugate model)
```

Part 9: Load Current Plot

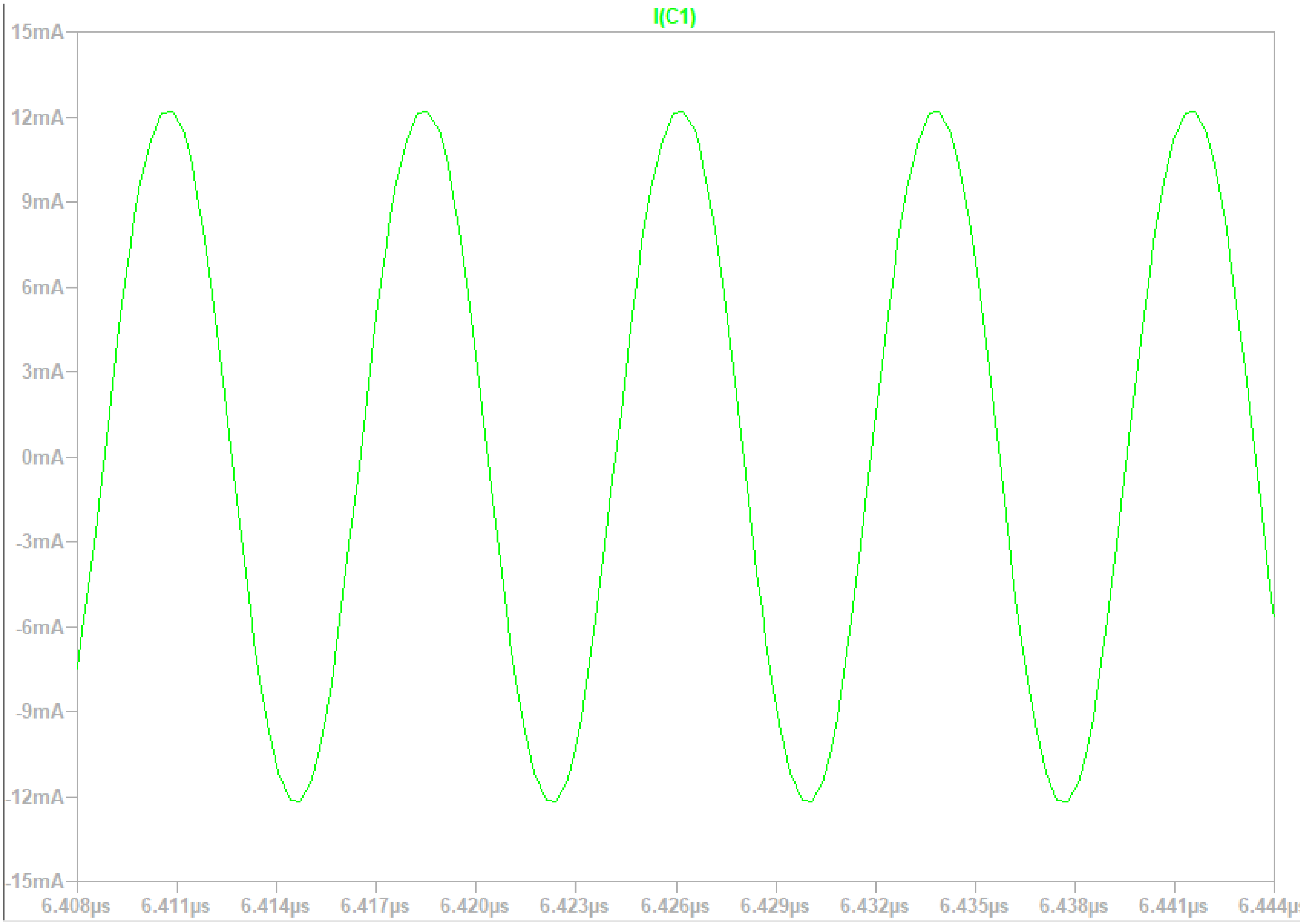

Figure 9.1 Load Current Through Thevenin Equivalent Circuit

The above graph shows the current through our complex conjugate that includes the 19.56 Ohm resistor and the 1.21 nanofarad capacitor. We see that we get 12mA through the capacitor and thus through the resistor. Performing Ohm's Law, we see that the voltage drop across the resistor is 234.72mV. Comparing this to our Thevenin voltage (polar form) we see that the voltage drop across the resistor is roughly half of the magnitude of the Thevenin voltage, 0.488327∠129.694. This shows that the max power we previously calculated is correct since we calculated max power using the Thevenin voltage. The load is only supposed to get half of the voltage of the Thevenin voltage, with the rest going to the Z_{th} or the rest of the circuit. Since this does happen with the load we calculated our predicted load/max power is correct.

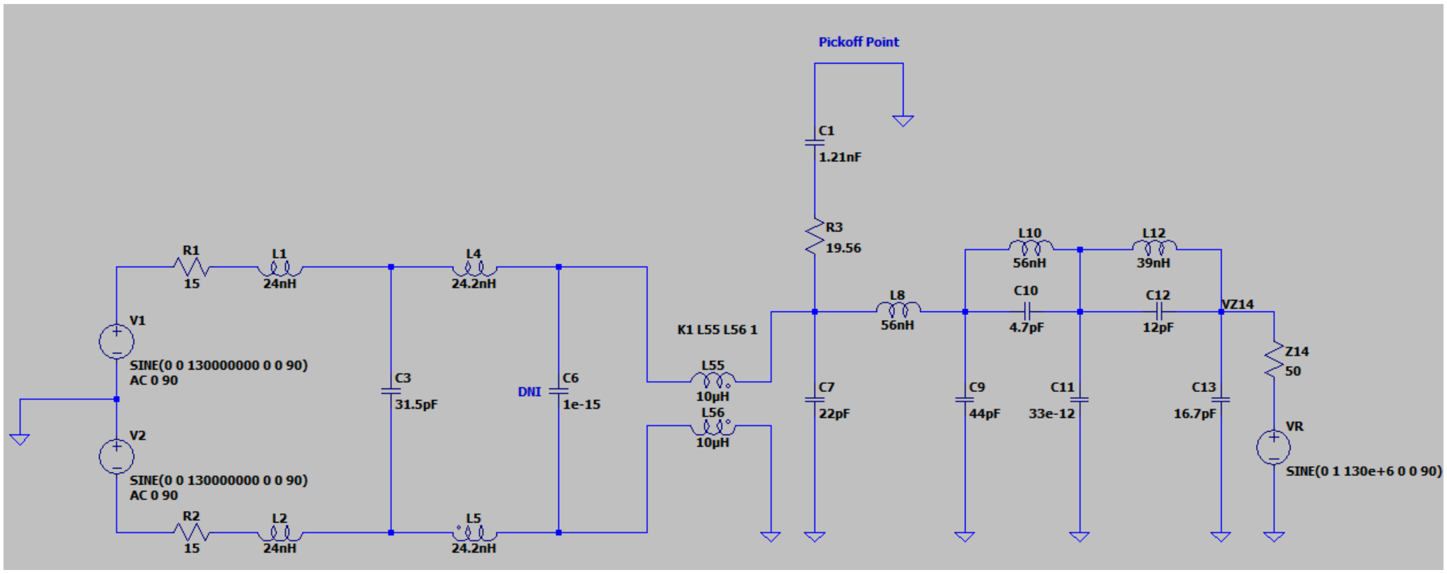

Figure 9.2 LTSpice Circuit with Load Components

Part 10: Matlab Print Out

```
% Filename:            EE310project.m
% Author:              Karl Horcasitas
%                      Ethan Nagelvoort
%                      Giemer Lozares
% Last Modified:       4/18/2020

% This script computes the frequency response for a s-domain transfer function.


clearvars
close all;
w = (130*10^6*2*pi); %our frequency is 130Mhz * 2pi


z1 = 15 + (j*w*24*10^(-9));  %initilize variables to hold our impedance values
z2 = 15 + (j*w*24*10^(-9));
z3 = 1/(j*w*31.5*10^(-12));
z4 = j*w*(24.2*10^(-9));
z5 = j*w*(24.2*10^(-9));
z6 = 1/(j*w*1*10^(-15));
z7 = 1/(j*w*22*10^(-12));
z8 = j*w*(56*10^(-9));
z9 = 1/(j*w*44*10^(-12));


%z10:
```

```
initL10 = j*w*(56*10^(-9));
initC10 = 1/(j*w*4.7*10^(-12));
z10 = (initL10*initC10)/(initL10+initC10);

z11 = 1/(j*w*33*10^(-12));

%z12
initL12 = j*w*(39*10^(-9));
initC12 = 1/(j*w*12*10^(-12));
z12 = (initL12*initC12)/(initL12+initC12);

z13 = 1/(j*w*16.7*10^(-12));
z14 = 50;

v1 = 1; v2 = v1; vR = 0; %using v1=v2=1cos(2pi*130mHz*t), vR = 0

A = [(z1+z2+z3) (-z3) 0 0 0 0 0; %Setting up matrix A using our mesh equation
       (-z3) (z3+z4+z5+z6) (-z6) 0 0 0 0;
       0 (-z6) (z6+z7) (-z7) 0 0 0;
       0 0 (-z7) (z7+z8+z9) (-z9) 0 0;
       0 0 0 (-z9) (z9+z10+z11) (-z11) 0;
       0 0 0 0 (-z11) (z11+z12+z13) (-z13);
       0 0 0 0 0 (-z13) (z13+z14)];
B = [(v1+v2);0;0;0;0;0;(-vR)]; %Setting up matrix B (note that vT2-VT1=0 in the third row)
X = linsolve(A,B); %linearly solve the matrices and store in X (X=A^-1*B)

%since v2=v1,
%now we raise v1 and v2 until power in z14 is 80 watts.
v1 = 71.180; v2=v1; %using these values of v1,v2, we get a power of 79.9942W
B = [(v1+v2);0;0;0;0;0;(-vR)];
X3 = linsolve(A,B); %using these new currents, we use eq 1/2 * Z14 * abs(i7)^2 to get power

ourI5 = X3(5,1); %convert i5 to polar to check answers
rhoI5=abs(ourI5);
thetaI5=angle(ourI5);
%-------------------
ourI7 = X3(7,1); %convert i7 to polar to check answers
rhoI7=abs(ourI7);
thetaI7=angle(ourI7);

%----------------------------------------Irms & Vrms Values
```

```
Power = 0.5 * (rhoI7^2) * (z14);

%RMS VALUES:
%R1:
R1Irms = (abs(X3(1,1)))/(sqrt(2)); %get the real part of i1 and divide by square root 2
R1V = X3(1,1) * 15; %R1 voltage drop = 15 ohm * i1
R1Vrms = abs(R1V)/(sqrt(2)); %get the real part of the R1 voltage drop and divide by square root 2
%R2:
R2Irms = R1Irms; %since R2 = R1, it will have the same Irms
R2Vrms = R1Vrms; %also the same Vrms
%L1:
L1Irms = R1Irms; %since L1 is in series with R1, they will have the same Irms
L1V = X3(1,1) * (j * w * 24*10^-9); %L1 voltage drop is i1 * (jw * 24nano Hz)
L1Vrms = abs(L1V)/(sqrt(2)); %get the real part of VL1 and divide by square root 2
%L2:
L2Irms = L1Irms; %since L2 is equal to L1 and in the same mesh, they have the same Irms
L2Vrms = L1Vrms; %also the same Vrms
%C3:
C3I = (X3(1,1)-X3(2,1)); %capacitor 3 current is (i1-i2)
C3Irms = abs(C3I)/sqrt(2); %get real part of C3I and divide by square root 2
C3V = C3I * (1/(j * w *31.5*10^-12)); %C3 voltage drop is C3I * ZC3
C3Vrms = abs(C3V)/sqrt(2); %get real part of C3 voltage drop and divide by square root 2
%L4:
L4Irms = abs(X3(2,1))/sqrt(2); %get real part of i2 and divide by square root 2
L4Vrms = abs(L4Irms * (j * w * 24.2*10^-9)); %get real part of L4 voltage drop (I * ZL4) and divide by square root 2
%L5:
L5Irms = L4Irms; %since L5 is equal to L4 and in the same mesh, they have the same Irms
L5Vrms = L4Vrms; %also the same Vrms
%C6:
C6Irms = abs(X3(2,1) - X3(3,1))/sqrt(2); %get real part of C6I (i2-i3) and divide by square root 2
C6Vrms = abs(C6Irms*(1/(j * w * 1*10^-15))); %get real part of C6 voltage drop (C6I * ZC6) using Irms
%C7:
C7Irms = abs(X3(3,1) - X3(4,1))/sqrt(2); %get real part of C7I (i3-i4) and divide by square root 2
C7Vrms = abs(C7Irms*(1/(j * w * 22*10^-12))); %get real part of C7 voltage drop (C7I * ZC7) using Irms
%L8:
L8Irms = abs(X3(4,1))/sqrt(2); %get real part of i4 and divide by square root 2
L8Vrms = abs(L8Irms*(j*w*56*10^-9)); %get real part of L8 voltage drop (L8I * ZL8) using Irms
%C9:
C9Irms = abs(X3(4,1) - X3(5,1))/sqrt(2); %get real part of C9I (i4-i5) and divide by square root 2
C9Vrms = abs(C9Irms*(1/(j * w * 44*10^-12))); %get real part of C9 voltage drop (C9I * ZC9) using Irms
%C10:
```

```
VZ10 = X3(5,1) * z10; %since we calculate impedance of L10 and C10 in parallel, we must split them up
VZ10rms = abs(VZ10)/sqrt(2); %get real part of that Z10 voltage drop and divide by square root 2
ZC10 = 1/(j*w*4.7*10^-12); %the impedance of C10
C10I = VZ10/ZC10; %since we have VZ10, divide by ZC10 (ohms law) to get current
C10Irms = abs(C10I)/sqrt(2); %get real part of C10I and divide by square root 2
C10Vrms = VZ10rms; %C10 parallel to L10, so they will have the same voltage drop and thus Vrms
%L10:
ZL10 = j * w * 56*10^-9; %impedance of L10
L10I = VZ10/ZL10; %since we have VZ10, divide by ZL10 to get current
L10Irms = abs(L10I)/sqrt(2); %get real part of L10I and divide by square root 2
L10Vrms = VZ10rms; %C10 parallel to L10, so they will have the same voltage drop and thus Vrms
%C11:
C11Irms = abs(X3(5,1) - X3(6,1))/sqrt(2); %get real part of C11 current (i5-i6) and divide by square root 2
C11Vrms = abs(C11Irms*(1/(j * w * 33*10^-12))); %get real part of C11 voltage drop (C11I * ZC11) using Irms
%C12:
VZ12 = X3(6,1) * z12; %like for VZ10, calculate the voltage drop of the parallel components
VZ12rms = abs(VZ12)/sqrt(2); %get real part of VZ12 and divide by square root 2
ZC12 = 1/(j*w*12*10^-12); %the impedance of C12
C12I = VZ12/ZC12; %since we have VZ12, divide by ZC12 (ohms law) to get current
C12Irms = abs(C12I)/sqrt(2); %get real part of C12I and divide by square root 2
C12Vrms = VZ12rms; %C12 parallel to L12, so they will have the same voltage drop and thus Vrms
%L12:
ZL12 = j * w * 39*10^-9; %impedance of L12
L12I = VZ12/ZL12; %since we have VZ12, divide by ZL12 to get current
L12Irms = abs(L12I)/sqrt(2); %get real part of L12I and divide by square root 2
L12Vrms = VZ12rms; %C12 parallel to L12, so they will have the same voltage drop and thus Vrms
%C13:
C13Irms = abs(X3(6,1) - X3(7,1))/sqrt(2); %get real value of C13I (i6-i7) and divide by square root 2
C13Vrms = abs(C13Irms*(1/(j * w * 16.7*10^-12))); %get real part of C13 voltage drop (C13I * ZC13) using Irms
%Z14:
Z14Irms = abs(X3(7,1))/sqrt(2); %get real value of Z14I and divide by square root 2
Z14Vrms = abs(Z14Irms*50); %using ohms law and Irms get Vrms

RMSvaluesUnrounded = [R1Vrms R1Irms; %store RMS values in a matrix
        L1Vrms L1Irms;
        R2Vrms R2Irms;
        L2Vrms L2Irms;
        C3Vrms C3Irms;
        L4Vrms L4Irms;
        L5Vrms L5Irms;
        C6Vrms C6Irms;
```

```
        C7Vrms C7Irms;
        L8Vrms L8Irms;
        C9Vrms C9Irms;
        L10Vrms L10Irms;
        C10Vrms C10Irms;
        C11Vrms C11Irms;
        L12Vrms L12Irms;
        C12Vrms C12Irms;
        C13Vrms C13Irms;
        Z14Vrms Z14Irms];

RMSvalues = round(RMSvaluesUnrounded*100)/100; %round all RMS values to 2 decimal places

%------------------------Thevinin Model
v1b=0; v2b=v1b;
%with vR=1(cos(2pi * 130MHz * t), vR=1
vRb=1;

Ath = [(z1+z2+z3) (-z3) 0 0 0 0 0; %Matrix for Thevin Voltage
        (-z3) (z3+z4+z5+z6) (-z6) 0 0 0 0;
        0 (-z6) (z6+z7) (-z7) 0 0 0;
        0 0 (-z7) (z7+z8+z9) (-z9) 0 0;
        0 0 0 (-z9) (z9+z10+z11) (-z11) 0;
        0 0 0 0 (-z11) (z11+z12+z13) (-z13);
        0 0 0 0 0 (-z13) (z13+z14)];
Bth = [(v1b+v2b);0;0;0;0;0;(-vRb)];
Xth = linsolve(Ath,Bth);

%Vth = i3-i4 * Z7
Vth = (Xth(3,1)-Xth(4,1))*z7;

%make C7 so huge so that it behaves like a short
z7th = 1/(j * w * 0.01*10^-6);
Ath2 = [(z1+z2+z3) (-z3) 0 0 0 0 0; %Matrix for Norton Current
        (-z3) (z3+z4+z5+z6) (-z6) 0 0 0 0;
        0 (-z6) (z6+z7th) (-z7th) 0 0 0;
        0 0 (-z7) (z7th+z8+z9) (-z9) 0 0;
        0 0 0 (-z9) (z9+z10+z11) (-z11) 0;
        0 0 0 0 (-z11) (z11+z12+z13) (-z13);
        0 0 0 0 0 (-z13) (z13+z14)];
```

```
Xth2 = linsolve(Ath2,Bth);

Inorton = (Xth2(3,1) - Xth2(4,1)); %Inorton = i3-i4 (the current of C7)

Zth = Vth/Inorton; %our Thevinin impedance Vth/Inorton

%the conjugate of our thevenin impedance
ZthConj = 19.561400245825563 - 1.015825791894316i;
Rth = 19.561400245825563; %the resistive part of our thevivin impedance
%using eq. 11.20 in textbook
%Pmax = abs(Vth^2)/8Rth
PmaxTh = ((abs(Vth))^2)/(8*Rth);

%Complex part of Zth
LforZth = 1.015825791894316i/(i*w); %our thevinin impedance as an inductor
CforZth = 1/(-1.015825791894316i * i * w); %as a capacitor (for our conjugate model)

HalfOfVth = Vth/2; %calculate half of Vth to check our work

%Vth as sinusoid
VthToSin = -0.311889913008039 + 0.375749839024849i;
rhoVthToSin=abs(VthToSin); %real part of the sinusoid
thetaVthToSin=angle(VthToSin);
VthSin=rhoVthToSin*exp(1i*thetaVthToSin);
```