
AC 2012-3243: MATLAB DEMONSTRATION OF TRANSMISSION LINE PHENOMENA IN ELECTROMAGNETICS

Dr. Stuart M. Wentworth, Auburn University

Stu Wentworth received his electrical engineering doctorate from the University of Texas, Austin, in 1990. Since then, he has been with Auburn University's Department of Electrical and Computer Engineering, specializing in electromagnetics and microelectronics. He has authored a pair of undergraduate electromagnetics texts and has won several awards related to teaching. He is the department's undergraduate Program Director and Chair of its Curriculum and Assessment Committee.

MATLAB¹ Demonstration of Transmission Line Phenomena in Electromagnetics

The university involved in this paper has a two semester sequence of electromagnetics. Coverage begins with the study of transmission lines, which constitute a very important application of electromagnetics. Signal reflection and impedance matching are critical concepts in wireless communications and radar. However, while students often can properly apply formulas to solve transmission line problems, they often lack a deep understanding of how these lines work.

MATLAB can serve as a useful tool for illuminating the operation of transmission lines, and its use has been incorporated in several textbooks such as Wentworth². MATLAB is used in variety of engineering classes and students have good familiarity with this important software package. In most schools MATLAB is widely available in engineering laboratories with access available to all faculty and students, mainly for classroom use. Many electrical/computer engineering leading industries use MATLAB and its toolboxes.

The first investigation in this paper demonstrates under what conditions a length of connecting wire must be treated as transmission line. Essentially, if the operating frequency is high enough that the wire length is a significant portion of a wavelength, then a transmission line model must be used. Here, a MATLAB demonstration clarifies where this occurs.

The second investigation concerns wave propagation in an impedance matching network realized using sections of transmission line (the so-called stub matching network). A MATLAB demonstration shows how the waves on each section of line behave and how they combine to achieve the impedance matching condition. An especially interesting aspect of this problem concerns how long after the wave is launched it takes for the matching condition to be achieved, and how the waves perform in the meantime.

When must wire be considered a T-Line?

It is commonly understood that wire interconnects must be treated as transmission lines when their lengths become comparable to a wavelength. A thorough discussion of this topic is provided by Inan & Inan,³ and Ulaby *et al.*⁴ considers that transmission line effects may be significant for wire length-to-wavelength ratio as low as 0.01. In this paper, a MATLAB demonstration⁵ illustrates when a distributed transmission line must be used to model wire interconnects.

Figure 1(a) shows a load R_L connected to a source (of voltage v_s and source resistance R_S) through a pair of connecting wires of length ℓ . The voltage across the load, v_L , is to be examined

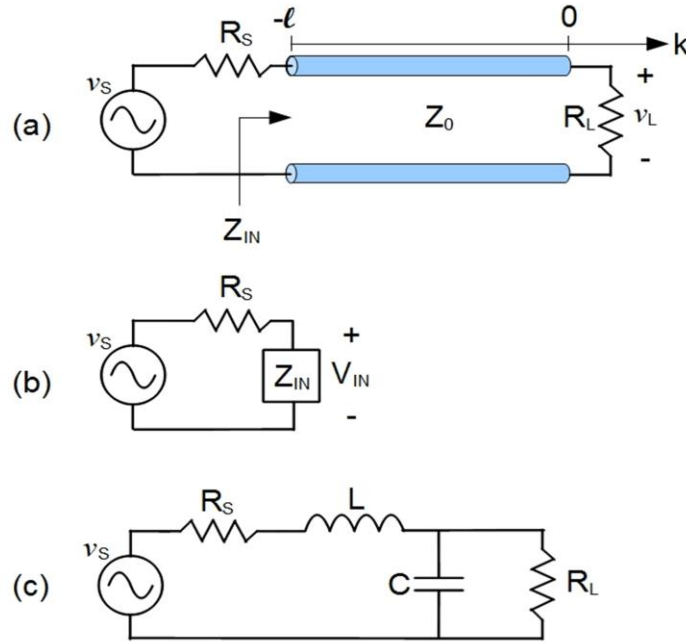


Figure 1: (a) Connecting wires between source and load are modeled by a length ℓ of transmission line. (b) Equivalent circuit with input impedance Z_{in} replacing the terminated line. (c) The circuit can also be modeled by a single section lumped element model.

as the length ℓ is increased. For ideal connecting wires, the length is assumed negligible, and the magnitude and the phase of v_L will therefore remain constant. This is compared with the actual v_L calculated assuming a transmission line of characteristic impedance Z_o .

Analysis of the transmission line circuits of Figure 1a and 1b provide the basis for the MATLAB code. First, for time harmonic excitation, the total voltage $v(k)$ at any point k on a lossless transmission line is the superposition of the incident and reflected waves. This can be written

$$v(k) = V_o^+ \left(e^{-j\beta k} + \Gamma_L e^{+j\beta k} \right) \quad (1)$$

where V_o^+ is the incident voltage wave amplitude, β is the phase constant, and Γ_L is the reflection coefficient at the load given by

$$\Gamma_L = \frac{R_L - Z_o}{R_L + Z_o} \quad (2)$$

We can therefore find the voltage at any point on the line using (1) if we know the value of V_o^+ .

At the input end of the line we have

$$V_{IN} = v(-\ell) = V_o^+ (e^{+j\beta\ell} + \Gamma_L e^{-j\beta\ell}) \quad (3)$$

We can also find an expression for V_{IN} by replacing the terminated line by its input impedance, Z_{IN} , given by

$$Z_{IN} = Z_o \frac{R_L + jZ_o \tan(\beta\ell)}{Z_o + jR_L \tan(\beta\ell)} \quad (4)$$

and we find by simple voltage division in Figure 1b:

$$V_{IN} = \frac{Z_{IN}}{Z_{IN} + R_S} V_S \quad (5)$$

where V_S is the phasor source voltage. Then we can equate (3) and (5) to find an expression for V_o^+ :

$$V_o^+ = V_S \frac{Z_{IN}}{Z_{IN} + R_S} \frac{1}{(e^{+j\beta\ell} + \Gamma_L e^{-j\beta\ell})} \quad (6)$$

The voltage at the load is then

$$V_L = V_o^+ (1 + \Gamma_L) \quad (7)$$

We can simplify our formulas for use in the MATLAB code by eliminating $\beta\ell$ in equations (4) and (6). Let us consider the line length ℓ to be d wavelengths long, or $\ell = d\lambda$, and we have

$$\beta\ell = \frac{2\pi}{\lambda} d\lambda = 2\pi d \quad (8)$$

Finally, we know that the formulas governing transmission line behavior are derived based on a distributed parameter model. Figure 1(c) shows a single-element model for the transmission line. The MATLAB demonstration will show when this single-element model must yield to the distributed model of the transmission line.

An interesting aspect of the analysis of Figure 1(c) involves calculation of L and C . In calculating total inductance L , starting with the distributed inductance L' (H/m), we can use the length $d\lambda$ from above and have $L = L' \ell = L' d\lambda$. Then,

$$j\omega L = j2\pi f L' d\lambda = j2\pi d L' (\lambda f) \quad (9)$$

Now, propagation velocity u_p along the line can be written

$$u_p = \lambda f = \frac{1}{\sqrt{L'C'}} \quad (10)$$

so we have

$$j\omega L = j2\pi dL' \left(\frac{1}{\sqrt{L'C'}} \right) = j2\pi d \sqrt{\frac{L'}{C'}} = j2\pi dZ_o \quad (11)$$

Similarly, it is easy to show

$$\frac{-j}{\omega C} = \frac{-jZ_o}{2\pi d} \quad (12)$$

Thus, we only need to know Z_o for the line rather than the particular values of L' and C' .

The MATLAB code “WireModel” developed using these equations is given in the appendix.

Figure 2 shows the results for the first simulation, where R_L and R_S are set to 200Ω and the line impedance is 50Ω . For ideal connecting wire, the magnitude of v_L would be constant at 1 V and the phase would be constant at 0° . Here we see that the length of the connecting wire begins to impact the load voltage (in particular the phase) at values of d less than 0.01λ . We also begin to see appreciable difference between the lumped and the distributed element model at about 0.10λ .

The second simulation is plotted in Figure 3, where R_L and R_S have each been reduced to 100Ω . With the lessening of impedance mismatch between connecting wire and system impedance, the deviation from ideal behavior occurs at longer length d . It is apparent that the degree of impedance mismatch is therefore an important consideration for determining when the wire must be treated as transmission line.

Development and understanding of the MATLAB code used in this demonstration requires a basic foundation in transmission line theory. Students can follow this theory to develop their own version of the code, or they can modify the existing code to study other situations. For instance, what happens when the line is matched to the system impedance? Can students determine how many lumped element sections are required to model a particular length of connection?

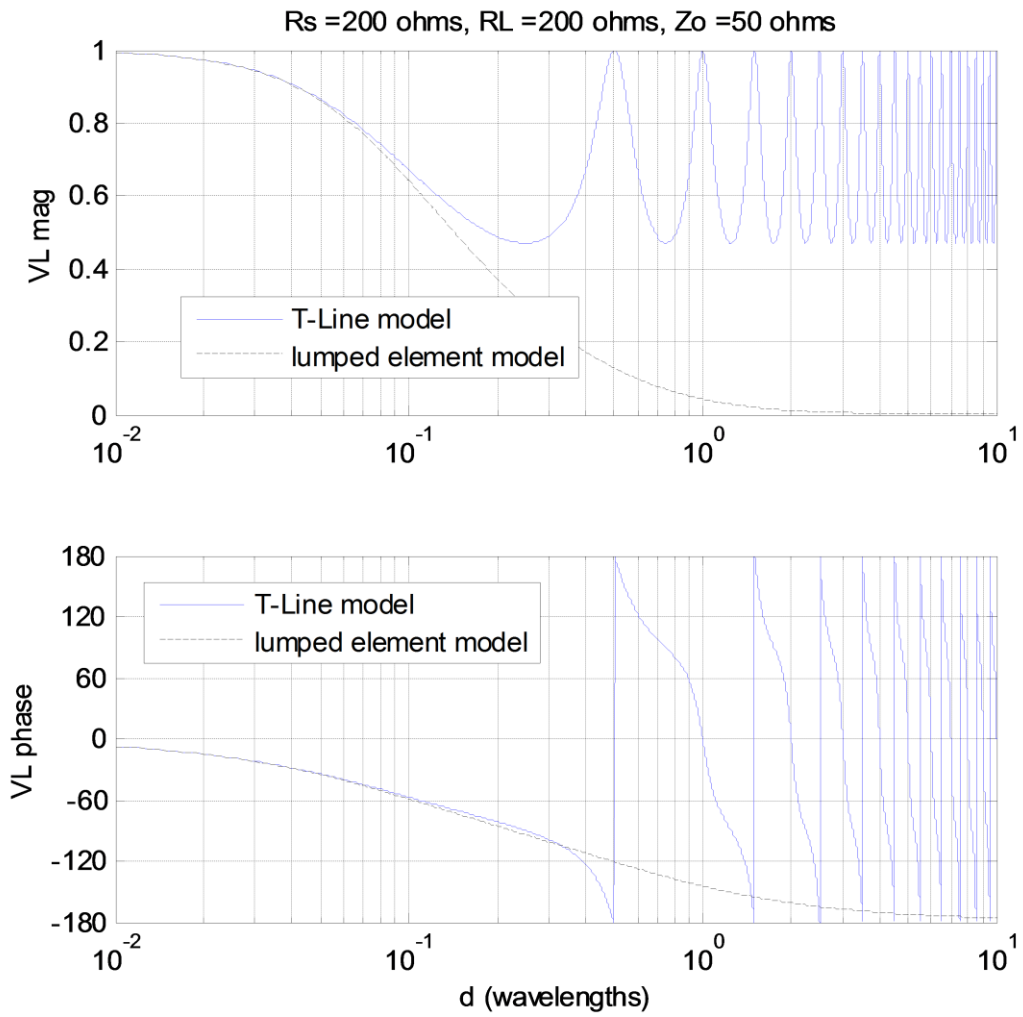


Figure 2: Magnitude and phase plots for v_L , (VL mag and VL phase, respectively) show the impact of transmission line length on load voltage for the case of 200 ohm system impedance and 50 ohm line impedance.

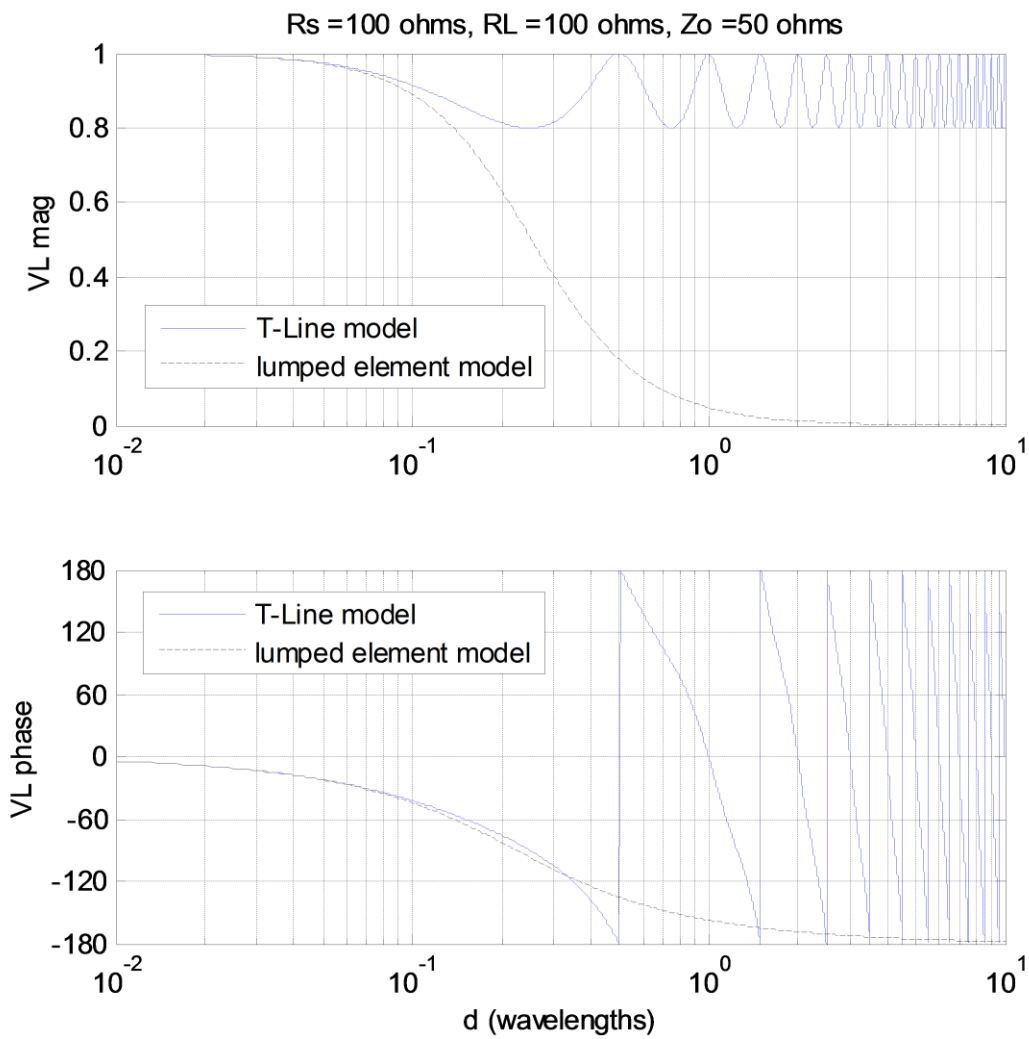


Figure 3: Magnitude and phase plots show the impact of transmission line length on load voltage for the case of 100 ohm system impedance and 50 ohm line impedance.

Waves launched on a matching network

After learning how transmission lines work, students are often introduced to the Smith Chart, a transmission line calculator tool that is handy for designing impedance matching networks using sections of transmission line. In the time harmonic case, the matching networks prevent waves from reflecting back to the source. If the lines are considered lossless, then no power is dissipated in the sections of transmission line and all of it must therefore reach the load, at least at the design frequency. One of the most popular of such networks is the shunt stub matching network, as shown in Figure 4(a). Here, the shunt stub is terminated in a short circuit although an open-ended termination could also be used. For the example shown in this paper, the system impedance is 50Ω and the load to be matched is $R_L = 170\Omega$. Appropriate lengths of through line and stub line are selected using the Smith Chart (see Figure 4(b)) to provide an impedance match between the Z_0 system impedance and the load impedance, R_L . A through line length of 0.170λ wavelengths and a stub line length of 0.104λ wavelengths accomplishes the impedance match.

Of course, it takes some amount of time after a wave is launched for a steady state time harmonic condition to be reached. The initially launched wave will impact the junction connecting the input line to the stub and through lines, and will be partially reflected. Eventually, superposition of the various wave components will accomplish the desired impedance match.

The MATLAB demonstration shows an animation of the wave launched on the input section of line. Progress of the wave on the through and stub lines can also be viewed. The programming becomes somewhat challenging (and the run time can become significant) because as a wave is incident on the junction, it spawns a reflected wave along with transmitted waves on the other

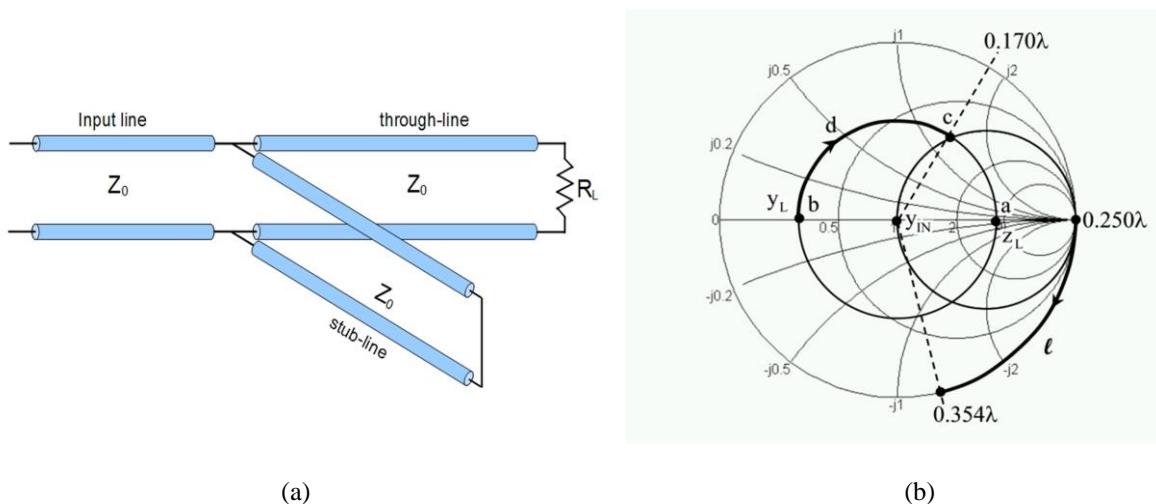


Figure 4: (a) A shorted shunt stub matching network. The lengths of the through and stub lines are typically found using the Smith Chart (b), where through and stub lengths are indicated as d and l , respectively. Here, the load is a 170Ω resistor, and the system impedance is 50Ω .

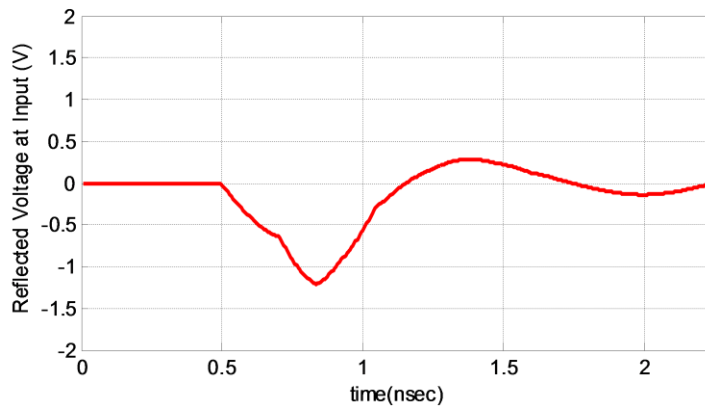


Figure 5: The value of the reflected voltage is displayed at the input end of the input line.

two sections of line. The spawned waves will reflect off of the discontinuity at the end of their lines (except for the input line, where a matched source impedance is used), and will travel back to the junction where three new waves will spawn. The approach taken in the MATLAB program is to prepare arrays for the input line, the through line and the stub line. Each array is of two dimensions, with columns representing position along the line, and rows representing time. Each row of data therefore represents a time snapshot of the total voltage on the line. The contributions of all waves (reflected, re-reflected, and so on) are folded into this array. Stepping through the snapshots therefore provides an animation of wave behavior on a line.

The program is kept fairly manageable by making several simplifications. First, the load is all real (no reactive components). All transmission line used is lossless. Finally, the source is matched to the system impedance so that we don't have to worry about re-reflection off of the source.

Figure 5 displays only the reflected voltage at the input end of the input line. At steady state, there will be no reflected waves if the line is properly impedance matched. Here we see that it takes 0.5 ns for the initial wave to travel to the junction and reflect back to the input end of the line. At this point we see significant reflection, but the matching network gradually manifests itself and the total reflected wave tends toward zero.

Figure 6 displays several snapshots from the animations for waves on the through and stub lines.

In this simulation, a half wavelength of transmission line is added to the design lengths for both through and stub lines, making their lengths 0.670 and 0.604 guide wavelengths, respectively. It doesn't change the matching condition, and gives us a chance to better view the waves on the line. It does significantly extend the simulation time, so this added length was not included for

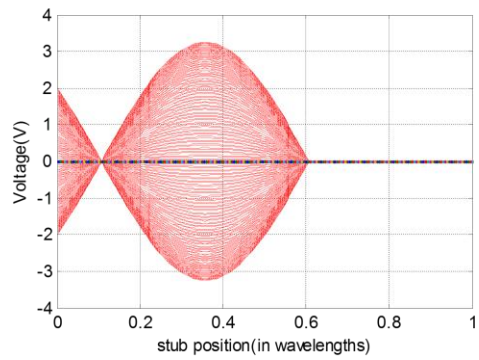
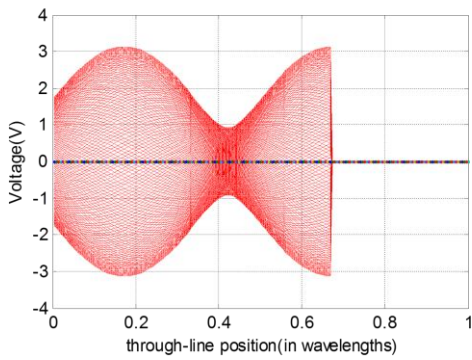
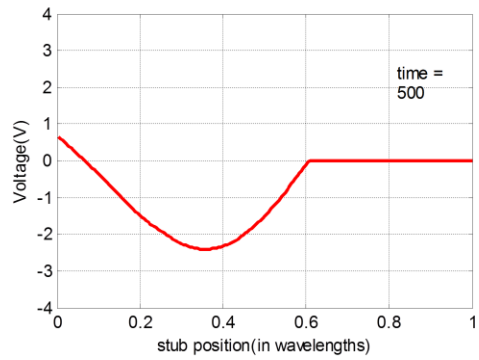
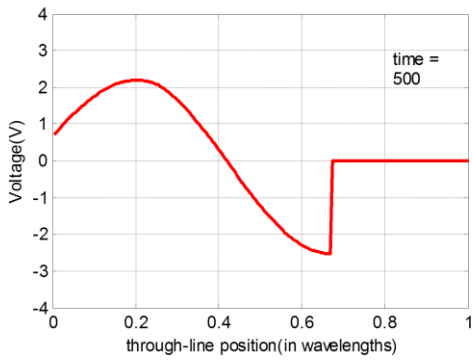
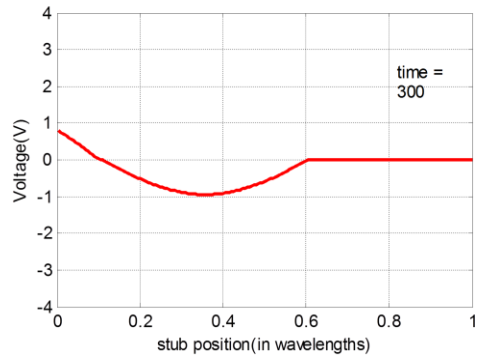
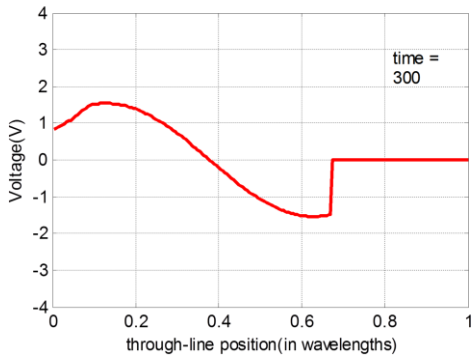
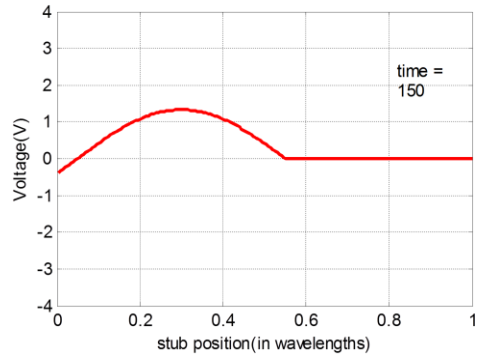
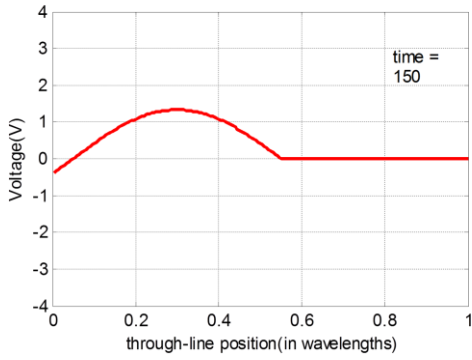


Figure 6: Snapshots of voltage on the through and stub lines. The last plots display standing wave patterns.

simulation of Figure 5. After a steady state is reached, when the additional reflected components are negligible, the “hold on” feature of MATLAB allows us to see the standing wave pattern on the through and stub lines. The animations of course are far more enlightening than the snapshots.

Students may modify the code to examine a number of conditions. First, it is easy to observe what happens on the lines when the matching network is not ideal; Simply change the line lengths or change the input signal frequency. A more challenging (but manageable) project would be to examine what happens when a pulse is launched. That is, after a certain amount of time, the input wave is turned off. This is an extremely useful case to examine as it is fundamental to radar systems. We would expect reflections to occur at both the leading and the trailing edges of the pulse. A much more significant programming challenge would be to allow the load impedance to be complex. While it is easy to handle complex loads in a steady state time-harmonic situation, it is much more difficult to handle them for non-steady state conditions.

Code described in this paper and other examples presented at the conference will be made available on the author’s website: <http://www.eng.auburn.edu/users/wentwsm/>

References:

1. MATLAB is a registered trademark of The MathWorks, Inc.
2. S. Wentworth, Applied Electromagnetics: Early Transmission Lines Approach, John Wiley & Sons, Inc., 2007.
3. U. S. Inan and A. S. Inan, Engineering Electromagnetics, Addison Wesley Longman, Inc., 1999, pp. 5-12.
4. F. T. Ulaby, E. Michielssen, and U. Ravaioli, Fundamentals of Applied Electromagnetics, 6th Ed., Prentice Hall, 2010, pp. 48-49.
5. Wentworth, pp. 58-61.

Appendix

```
% WireModel
% When does wire need to be treated as T-Line? And when
% does T-Line behavior deviate from simple lumped element?
% 1/3/12 SMW
clc;clear;
Zo=50;           % line characteristic impedance, ohms
RS=100;         % source resistance, ohms
RL=100;         % load resistance, ohms
Vs=2;          % source phasor voltage, V
d=.01:.001:10; % wire length, in wavelengths

%T-Line calculations
GL=(RL-Zo)/(RL+Zo); % load reflection coefficient
num=RL+j*Zo.*tan(2*pi.*d);
den=Zo+j*RL.*tan(2*pi.*d);
Zin=Zo.*num./den; % line input impedance (Eqn (4))
Vin=Vs.*Zin./(Zin+RS); % input voltage (Eqn (5))
Vop=Vin./(exp(j*2*pi.*d)+GL.*exp(-j*2*pi.*d)); % (Eqn (6))
vLTL=Vop.*(1+GL); % T-Line model phasor load voltage
VLTL=abs(vLTL); % magnitude
phLTL=180.*angle(vLTL)./pi; % phase

% Lumped element calculations
A=-j.*Zo./(2*pi.*d); % capacitor impedance
Z1=(RL.*A)./(RL+A); % impedance of parallel R and C
Z2=RS+j*2*pi*Zo.*d; % ...add series inductance
vLle=Vs.*Z1./(Z1+Z2); % lumped model phasor load voltage
VLle=abs(vLle); % magnitude
phle=180.*angle(vLle)./pi; % phase

%generate first plot
subplot(2,1,1)
semilogx(d,VLTL,d,VLle,'--k')
ylabel('VL mag')
grid on
legend('T-Line model','lumped element model')

%assemble string data for title
RSstr=strcat('Rs = ',num2str(RS),' ohms');
RLstr=strcat(' , RL = ',num2str(RL),' ohms');
Zostr=strcat(' , Zo = ',num2str(Zo),' ohms');
Tstr=strcat(RSstr,RLstr,Zostr);
title(Tstr)

%generate second plot
subplot(2,1,2);
semilogx(d,phLTL,d,phle,'--k');
axis([.01 10 -180 180])
set(gca,'YTick',[-180:60:180])
ylabel('VL phase')
legend('T-Line model','lumped element model')
xlabel('d (wavelengths)')
grid on
```