1 Tline – A Transmission Line Calculator

tline is an open-source project licensed under GPLv3 (or later). This program was written to explore the mathematics of transmission lines in a manner that encourages learning and participation. One might ask "Why write a new transmission line calculator when there are existing ones?"

The problem with the existing transmission line calculators we reviewed is that they are invariably proprietary programs, and their source code is not available for study. While they may be good tools for solving specific problems, they are not helpful if one wishes to understand the way the solutions are computed.

Also, since we cannot tell what the programs are doing internally, we have no guarantee that they are actually computing the correct results in all cases. *tline* is intended to provide a more transparent approach.

1.1 Formulas

This section describes the formulas *tline* uses to perform its calculations.

1.1.1 Common Line Properties

In many of the formulas below, we will need several parameters, as defined in this section.

1.1.1.1 Units

Manufacturers can either specify transmission line parameters in the English system or in the Metric system. In all equations that follow, we assume that one measurement system will be chosen and used consistently.

1.1.1.2 Length

The length ℓ of a transmission line is generally expressed in either *meters* or *feet*.

1.1.1.3 Velocity Factor

In a vacuum, electrical waves travel at the speed of light (*c*), which is 299792458 meters per second. In the English system, this is appropriately 983571056 feet per second.

In a transmission line, electrical waves travel more slowly, which we describe by the velocity factor (*vf*):

$$vf=\frac{speed\ of\ electrical\ wave\ in\ transmission\ line}{speed\ of\ electrical\ wave\ in\ vacuum} \tag{1}$$

The velocity factor is a unitless quantity, and will be less than 1.0; typical values range from 0.6 to 0.9 or so.

1.1.1.4 Wavelength

Wavelength (λ) in a transmission line is defined in terms of the frequency of a signal and the velocity factor of the line:

$$\lambda=\frac{vf\times speed\ of\ light}{frequency} \tag{2}$$

The units of wavelength will typically either be *meters per cycle* or *feet per cycle*, although smaller units may be used at very high frequencies. Often, we just say "*the wavelength is 40 meters*", but we really mean "*40 meters per cycle*".

1.1.1.5 Phase Constant

The phase constant (β) is defined in terms of the wavelength in a line, expressed as *radians per unit length*, i.e. *radians per meter* or *radians per foot*:

$$\beta=\frac{2\pi}{\lambda} \tag{3}$$

1.1.1.6 Attenuation

Attenuation (α) is defined in terms of the cable loss, which is typically specified in *dB per 100 feet* or *dB per 100 meters*. However, for most equations, we need attenuation as *nepers per unit length*, i.e. *nepers per meter* or *nepers per foot*:

$$\alpha=\frac{attenuation(dB\ per\ 100\ feet)\times 0.11513(nepers\ per\ dB)}{100} \tag{4}$$

$$\alpha=\frac{attenuation(dB\ per\ 100\ meters)\times 0.11513(nepers\ per\ dB)}{100} \tag{5}$$

1.1.1.7 Complex Loss Coefficient

The complex loss coefficient is defined in terms of α and β:

$$\gamma=\alpha+j\beta \tag{6}$$

1.1.2 Manufacturer-Specified Line

Manufacturers typically specify matched line loss only at a few frequencies. *tline* needs to interpolate / extrapolate to find the attenuation at other frequencies. This is done using the AC6LA transmission line model, according to the equations found in cableTypes.cpp.

Source file cableTypes.cpp also contains a table of so-called "k" parameters, which are generated by a script from AC6LA. The script takes the manufacturer's data as input, and produces the k parameters.

The table also contains the nominal velocity factor, the nominal cable impedance (real part only), and the maximum allowed cable voltage. As part of the transmission line model, we can also find the corrected velocity factor and cable impedance.

Here are the steps to add a new cable type to *tline's* source code:

1) Download a copy of the AC6LA tool from the following URL:
   https://ac6la.com/adhoc/Set-Compare_k0k1k2.zip
2) Download a copy of SimSmith from the following URL:
   http://www.ae6ty.com/Smith_Charts.html
   (Scroll down on the web page to find the dropbox link for the current version)
3) Generate new "k" parameters from the manufacturer's data sheet using the instructions in the AC6LA "_Read Me.txt" file
4) Edit cableTypes.cpp to add the new data to the parameter table
5) Run the wxFormBuilder tool and add the exact same cable name string to the ui_cableType wxComboBox *choices* field (the order must be kept the same too)
6) Type "F8" in wxFormBuilder to regenerate the tlineUI source files
7) Rebuild *tline* from the updated source files

1.1.3 Relating Cable Impedance to Resistance and Reactance

While many of the parameters of a transmission line are easy to find in a typical datasheet, cable characteristic resistance R_0, and reactance X_0 are typically not specified by the cable manufacturers. Therefore, we must calculate R_0 and X_0 from $|Z_0|$, a parameter which *is* specified by cable manufacturers.

To perform this calculation, we need $\alpha, \beta, \text{and} |Z_0|$. We can find α and β from the velocity factor, wavelength, and attenuation, as described previously.

We know that transmission line impedance has a real and imaginary part, so $Z_0 = R_0 - jX_0$, where the negative sign implies that the reactance is capacitive (this is always the case for transmission lines). R_0 and X_0 are at right angles in the complex plane, so by the Pythagorean theorem, we have:

$$|Z_0| = \sqrt{R_0^2 + X_0^2} \quad (7)$$

We also know that R_0 and X_0 are related by:

$$X_0 = -R_0 \frac{\alpha}{\beta} \quad (8)$$

Substituting Eq. 2 into Eq. 1 and solving for R_0 we get:

$$R_0 = \frac{|Z_0|}{\sqrt{1 + \left(\frac{\alpha}{\beta}\right)^2}} \quad (9)$$

All the variables in Eq. (9) are available on the datasheet, so we can calculate R_0 . We can then find X_0 from Eq. (8).

1.1.4 User-Specified Line

While *tline* contains data for many different types of line, it may also be desirable to handle user-specified line. Selecting “User-Defined Transmission Line” in the pull-down on *tline’s* main page brings up a dialog where the user can enter the parameters from the manufacturer datasheet. If the user fills in the impedance value, *tline* will calculate the resistance and reactance using the method described in section 1.1.3 above. The user can also override the calculated resistance and/or reactance; *tline* will then calculate a new value for the impedance. See file userLine.cpp for the source code. In particular:

ResistanceReactanceFromImpedance(), as the name implies, calculates the resistance and reactance from the impedance.

ImpedanceFromResistanceReactance() calculates the impedance from the resistance and reactance.

Note that it is unlikely that the manufacturer will specify the attenuation at the exact frequency one wishes to analyze. Therefore, one will have to manually interpolate the datasheet data to find the attenuation at the desired frequency. It is best to do this on a log-log graph, because the attenuation roughly forms a straight line when plotted that way.

1.1.5 Impedance At The Input Of A Line

The *Transmission Line Equation* can be used to find the impedance at the input of a line. The variables in the transmission line equation are complex numbers, and the equation uses hyperbolic trigonometric functions of complex variables. This is easy to do by computer, but hard to do on paper.

The following variables are needed:

1) complex impedance of the load Z_L
2) complex characteristic impedance of the cable itself Z_0
3) complex loss coefficient γ
4) length of the line ℓ

Given those variables, *tline* solves for the input impedance Z_{in} :

$$Z_{\text{in}} = Z_0 \frac{Z_L \cosh(\gamma\ell) + Z_0 \sinh(\gamma\ell)}{Z_0 \cosh(\gamma\ell) + Z_L \sinh(\gamma\ell)} \tag{10}$$

This equation may be found in chapter 24 of *Reference Data For Radio Engineers*, *6th edition*, ISBN 0-672-21218-8.

See function *impedanceAtInput()* in file *tlineLogic.cpp*.

1.1.6 Impedance At The Far End Of A Line

If we plan to install a remote antenna tuner at an antenna, we may wish to find the impedance at the far end of the transmission line, given a known impedance at the input of the line. I.e. we wish to find Z_L given Z_{in} . Similar to the previous section, we can use the inverse form of the *Transmission Line Equation*:

$$Z_L = Z_0 \frac{Z_{\text{in}} \cosh(\gamma\ell) - Z_0 \sinh(\gamma\ell)}{Z_0 \cosh(\gamma\ell) - Z_{\text{in}} \sinh(\gamma\ell)} \tag{11}$$

This equation may be found in chapter 24 of *Reference Data For Radio Engineers*, *6th edition*, ISBN 0-672-21218-8.

See function *impedanceAtLoad()* in file *tlineLogic.cpp*.

1.1.7 Voltage And Current Along A Line

The voltage at any point ℓ along a transmission line may be found from:

$$V_\ell = V_{\text{in}} \left\{ \cosh(\gamma\ell) - \left(\frac{Z_0}{Z_{\text{in}}} \right) \sinh(\gamma\ell) \right\} \tag{12}$$

Similarly, the current at any point ℓ along a transmission line may be found from:

$$I_\ell = \frac{V_{\text{in}}}{Z_{\text{in}}} \left\{ \cosh(\gamma\ell) - \left(\frac{Z_{\text{in}}}{Z_0} \right) \sinh(\gamma\ell) \right\} \tag{13}$$

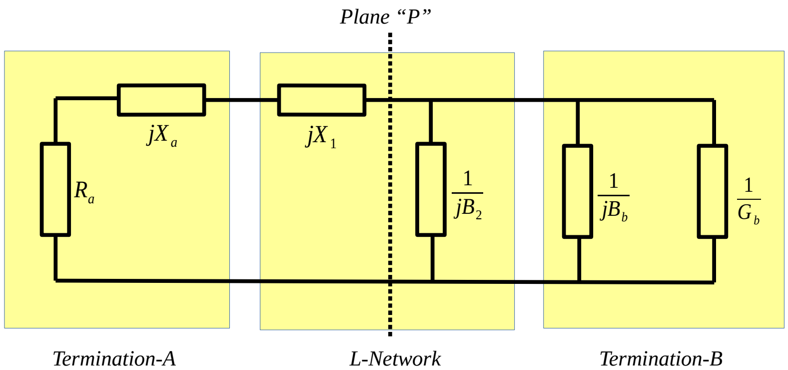
See functions *voltageOut()* and *currentOut()* in file *tlineLogic.cpp*.

1.1.8 L-Network Calculations

The following is based on Royal Aircraft Establishment Technical Report 78049, "Impedance-Matching Networks of the L Type", 4 May 1978, by P. B. Walkley.

We wish to design an *L-network* to match the impedance of two different devices for optimal power transfer. This will happen when the impedance of one device is the complex conjugate of the impedance of the other device. Call one device *Termination-A*, and the other device *Termination-B*.

Consider the following diagram, showing the two devices connected by an L-network:



Termination-A is presented as an arbitrary impedance, i.e. a real part R_a and an imaginary part jX_a in series. Similarly termination-B is presented as an arbitrary admittance, i.e. a real part $\frac{1}{G_b}$ and an imaginary part $\frac{1}{jB_b}$ in parallel. We use *admittance format* for termination B, because admittances in parallel add, which simplifies the math, as we will soon see. Similarly, we use *impedance format* for termination A, because impedances in series add.

Consider Plane "P", which cuts through the center of the "L" network. For maximum power transfer, the combined impedance of everything to the left of plane "P" must equal the complex conjugate of the combined admittance of everything to the right of plane "P".

It is required that $\frac{1}{G_b} \geq R_a$ because the parallel component of the L-network will lower the impedance looking from Plane "P" towards the right side of the diagram, while the series component of the L-network will raise the impedance looking from Plane "P" towards the left side of the diagram. This restriction is easily accommodated by choosing which device we call "A" and which device we call "B".

Since we are trying to find a complex conjugate match, we wish to solve for X_1 and B_2 in the following equation:

$$R_a + jX_a + jX_1 = \left\{ \frac{1}{G_b + jB_b + jB_2} \right\}^* \tag{14}$$

where the "*" represents the complex conjugate. The algebra required is not difficult, but it took a while to find a clean approach, thus we will present the steps in detail.

To solve Eq. 1 for X_1 we first clear the imaginary terms from the denominator of the right hand side by multiplying both the numerator and denominator by the complex conjugate:

$$R_a + jX_a + jX_1 = \left\{ \frac{1}{G_b + jB_b + jB_2} \right\}^* \cdot \left\{ \frac{G_b + jB_b + jB_2}{G_b + jB_b + jB_2} \right\} \tag{15}$$

Simplifying, we get:

$$R_a + jX_a + jX_1 = \left\{ \frac{G_b + jB_b + jB_2}{G_b^2 + (B_b + B_2)^2} \right\} \tag{16}$$

Next, separate the real and imaginary parts to get two simultaneous equations:

$$R_a = \frac{G_b}{G_b^2 + (B_b + B_2)^2} \tag{17}$$

$$X_a + X_1 = \frac{B_b + B_2}{G_b^2 + (B_b + B_2)^2} \tag{18}$$

We now take Eq. (17) through a series of rearrangements, because we want to isolate the $B_b + B_2$ term:

$$R_a G_b^2 + R_a (B_b + B_2)^2 = G_b \tag{19}$$

$$R_a (B_b + B_2)^2 = G_b - R_a G_b^2 \tag{20}$$

$$(B_b + B_2)^2 = \frac{G_b}{R_a} - G_b^2 \tag{21}$$

$$B_b + B_2 = \pm \sqrt{\frac{G_b}{R_a} - G_b^2} \tag{22}$$

Now we substitute Eq. (21) and Eq. (22) into Eq. (18) to produce Eq. (23):

$$X_a + X_1 = \frac{\pm\sqrt{\frac{G_b}{R_a} - G_b^2}}{G_b^2 + \frac{G_b}{R_a} - G_b^2} \tag{23}$$

Now simplify Eq. (23):

$$X_a + X_1 = \frac{\pm\sqrt{\frac{G_b}{R_a} - G_b^2}}{\frac{G_b}{R_a}} \tag{24}$$

$$X_a + X_1 = \pm\frac{R_a}{G_b}\sqrt{\frac{G_b}{R_a} - G_b^2} \tag{25}$$

$$X_a + X_1 = \pm R_a\sqrt{\frac{G_b}{R_a G_b^2} - \frac{G_b^2}{G_b^2}} \tag{26}$$

$$X_a + X_1 = \pm R_a\sqrt{\frac{1}{R_a G_b} - 1} \tag{27}$$

We now have an equation for the X_1 component of the L-network:

$$\boxed{X_1 = -X_a \pm R_a\sqrt{\frac{1}{R_a G_b} - 1}} \tag{28}$$

We will now go through the same process to solve for B_2 but we start by flipping Eq. (14) over and reversing the left hand and right hand sides:

$$(G_b + jB_b + jB_2)^* = \frac{1}{R_a + jX_a + jX_1} \tag{29}$$

$$(G_b + jB_b + jB_2)^* = \frac{1}{R_a + jX_a + jX_1} \cdot \left\{\frac{R_a + jX_a + jX_1}{R_a + jX_a + jX_1}\right\}^* \tag{30}$$

$$(G_b + jB_b + jB_2)^* = \frac{\{R_a + jX_a + jX_1\}^*}{R_a^2 + (X_a + X_1)^2} \tag{31}$$

This time, we apply the conjugate operator by flipping the signs of all the imaginary terms:

$$(G_b - jB_b - jB_2) = \frac{\{R_a - jX_a - jX_1\}}{R_a^2 + (X_a + X_1)^2} \tag{32}$$

Again separate the real and imaginary parts to get two simultaneous equations:

$$G_b=\frac{R_a}{R_a^2+(X_a+X_1)^2} \tag{33}$$

$$B_b+B_2=\frac{(X_a+X_1)}{R_a^2+(X_a+X_1)^2} \tag{34}$$

Take Eq. (33) through similar transformations to what we did previously:

$$G_b R_a^2+G_b(X_a+X_1)^2=R_a \tag{35}$$

$$G_b(X_a+X_1)^2=R_a-G_b R_a^2 \tag{36}$$

$$(X_a+X_1)^2=\frac{R_a}{G_b}-R_a^2 \tag{37}$$

$$X_a+X_1=\pm\sqrt{\frac{R_a}{G_b}-R_a^2} \tag{38}$$

Substitute Eq. (37) and Eq. (38) into Eq. (34):

$$B_b+B_2=\frac{\pm\sqrt{\frac{R_a}{G_b}-R_a^2}}{R_a^2+\frac{R_a}{G_b}-R_a^2} \tag{39}$$

$$B_b+B_2=\frac{\pm\sqrt{\frac{R_a}{G_b}-R_a^2}}{\frac{R_a}{G_b}} \tag{40}$$

$$B_b+B_2=\pm\frac{G_b}{R_a}\sqrt{\frac{R_a}{G_b}-R_a^2} \tag{41}$$

$$B_b+B_2=\pm G_b\sqrt{\frac{R_a}{R_a^2 G_b}-\frac{R_a^2}{R_a^2}} \tag{42}$$

$$B_b+B_2=\pm G_b\sqrt{\frac{1}{R_a G_b}-1} \tag{43}$$

We now have an equation for the B_2 component of the L-network:

$$\boxed{B_2=-B_b\pm G_b\sqrt{\frac{1}{R_a G_b}-1}} \tag{44}$$

Both Eq. (28) and Eq. (44) contain $\pm$*radicals* which might make us think that we have found four solutions to Eq. (14). However, if we substitute Eq. (28) and Eq. (44) back into Eq. (14), we find that the signs of the radicals must match; i.e. either we take both the Eq. (28) and Eq. (44) radicals to be positive, or we take both to be negative. The combinations where one radical is positive and the other radical is negative are not solutions to Eq. (14).

Summarizing, here are the two valid solutions:

$$X_1 = -X_a + R_a\sqrt{\frac{1}{R_a G_b} - 1} \text{ and } B_2 = -B_b + G_b\sqrt{\frac{1}{R_a G_b} - 1} \text{ (solution 1)}$$

$$X_1 = -X_a - R_a\sqrt{\frac{1}{R_a G_b} - 1} \text{ and } B_2 = -B_b - G_b\sqrt{\frac{1}{R_a G_b} - 1} \text{ (solution 2)}$$

Numerically, a positive value for either X_1 or B_2 means that the component is an inductor, while a negative value means that the component is a capacitor. When we apply the equations, we may find an L-network consisting of one inductor and one capacitor, or we may find an L-network consisting of two capacitors or two inductors.

See the various *lnet* functions in *tuner.cpp*.

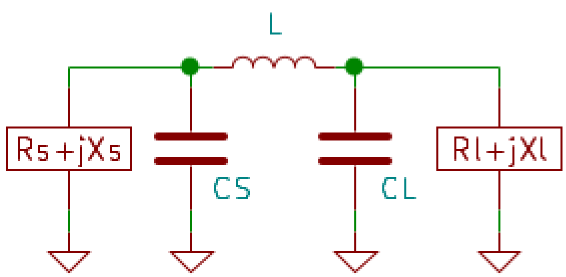
1.1.9 PI-Network and T-Network Calculations

PI and T networks have an extra degree of freedom compared to L networks. As a result, one must specify some extra information to constrain the solution. We specify a “network Q” parameter for that. The network Q is an upper bound; if one were to plot the resulting network on a Smith Chart, the breakpoints caused by the reactive components would have a Q less than or equal to the specified parameter.

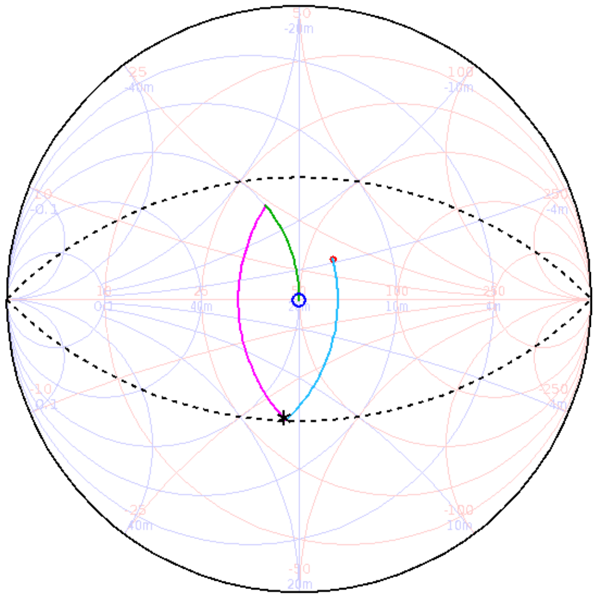
For example, suppose the following parameters:

- Frequency = 7 MHz
- Source Impedance = 50 + j0
- Load Impedance = 60.51 + j17.13
- Desired Q = 1

If we ask for a low-pass PI network, we will get the following schematic. CS will be 331.05 pF, L will be 1283.95 nH, and CL will be 446.35 pF.



When plotted with a tool like "SimSmith", we will see the following plot, where the green line represents CS, the magenta line represents L, and the cyan line represents CL :



The dotted lines represent a Q = 1 boundary. We see that the point between CS and L (where the cyan and magenta lines meet) just touches the lower Q boundary at the asterisk symbol. The point between L and CL (where the magenta and green lines meet) is well within the Q = 1 area.

Given the Q constraint, we can solve for either the CS or CL component by using the following equation:

$$X_C = \frac{1}{(B_A - Q \times G_A)} \tag{45}$$

where B_A is the imaginary component of the user-specified source or load impedance, expressed as susceptance, and G_A is the real component of the impedance, expressed as admittance. Similarly, if we were solving a high-pass PI network, we would use:

$$X_L = \frac{1}{(Q \times G_A + B_A)} \tag{46}$$

Similar formulas are used for the T-network cases. Please see functions *tryPI* and *tryT* in file *tuner.cpp*.

Once we know the reactance of one of the components, we can absorb it into either the source impedance or the load impedance. Initially we don't know which one to choose, so we try them both. One choice will result in both meeting points being within the Q boundary; the other choice will generally result in one of the meeting points being outside the boundary; naturally we reject that choice.

Once we've selected CS or CL, we are left with an L-network, which can be solved using the algorithm described in chapter 1.1.8.

1.1.10 Voltage, Current, and Power Lost in Tuners

We wish to solve for the voltages, currents, and power lost in the various components of an antenna tuner. While there are several ways to do that, we have chosen to first solve for the impedance and admittance at each node in the circuit.

We work backwards from the Load finding the impedances and admittances. The impedance of the load is simply $R+jX$, where R, the resistance, is the real part, and X, the reactance, is the imaginary part.

Each component also has a resistance and reactance, where the reactance comes from the equations in sections 1.1.8 and 1.1.9 and the resistance comes from the non-ideal nature of the component. We calculate the resistance from the reactance divided by the "Q" or quality factor:

$$R=\left|\frac{X}{Q}\right| \tag{47}$$

In a loop, we look at each preceding component in turn. If the component is in series, we add its impedance to that of the following node; if it is in parallel, we add its admittance to that of the following node. In either case, we also find the reciprocal, since components alternate as series and parallel. Thus we wind up with the impedance and admittance at each node.

Once we find the impedances and admittances, we next find the voltages and currents. The user has specified the desired power, and we use that plus the source impedance of the antenna tuner to calculate the voltage and current that would produce the desired power. Note that if the source impedance has a reactive component, some of the power will also be reactive. Typically however, the source impedance of an antenna tuner will have a very minimal reactive component, because the whole point of an antenna tuner is to match the impedance of a transceiver or feed line.

$$V_{SOURCE}=\sqrt{P_{DESIRED}\times Z_{SOURCE}} \tag{48}$$

$$I_{SOURCE}=\sqrt{\frac{P_{DESIRED}}{Z_{SOURCE}}} \tag{49}$$

We now iterate over the components. If the first component is in series with the source, then we know the current through that component, and using the impedance of the following components we can find the voltage across those components:

$$V_{n+1}=I_n\times Z_{n+1} \tag{50}$$

Conversely, if the first component is in parallel with the source, then we know the voltage across that component, and we can find the current through the following components:

$$I_{n+1} = \frac{V_n}{Z_{n+1}} \tag{51}$$

For a purely resistive circuit, power can be calculated as $V \times I$, V^2/R, or $I^2 \times R$. However, as we've already noted, the components of an antenna tuner have both a resistance and a reactance. We can find the power lost in such a component by using formula (52) when we know the voltage across the component, and we can use formula (53) when we know the current through the component:

$$P_{LOST} = \Re\left(V_n \times V_n^* \times Y_n\right) \tag{52}$$

$$P_{LOST} = \Re\left(I_n \times I_n^* \times Z_n\right) \tag{53}$$

In the above formulas, Z_n is the impedance of the component, and Y_n is the admittance of the component. The "*" superscript denotes the complex conjugate operator, and $\Re(f)$ indicates the real part of f.

The implementation of these equations can be found in *tuner::show3Part()* in file *tuner.c*.