

Extending Design and Simulation with User Defined Equations and Variables

By Dale D. Henkes
ACS

This article describes how modern EDA tools that support user-defined equations allow functionality and flexibility beyond the software's built-in models and computational options

Though taken for granted now, the ability to routinely enter almost any kind of complex circuit schematic into a circuit simulator program and get circuit performance data accurately representing the physical hardware is truly remarkable. However, one of the greatest values of computer aided engineering software is provided, not by the simulation of a static circuit with unchanging components, but by the ability to change the value of any number of circuit components interactively (in real time) while immediately observing the changes in circuit performance. Most circuit simulators provide the ability to change a component value via some kind of variable so that the effect on circuit performance can be seen without altering the original circuit. The variable(s) can be changed manually or through the program's optimizer, where they are adjusted in an algorithm designed to automatically achieve a specified performance goal.

Another asset related to variables, that is available in more advanced circuit simulators, is the ability for the user to construct mathematical models (from variables) that uniquely characterize component parameters or circuit behavior in new ways. These mathematical models are entered by the user (and retained in the schematic file) as user defined equations. In the following paragraphs, the usage of variables and equations in circuit simulation will be examined, showing examples of ways to enhance the simulations and provide greater insight into the design and analysis of

circuits. The LINC2 Pro circuit simulator from Applied Computational Sciences (ACS) is used to work out the examples.

Type and Scope of Variables

There are at least two different kinds of variables through which the user (or optimizer) can modify circuit component parameters in real time. The difference is primarily in the scope of the variable. The scope can be local or global. A variable that is local to only a single component can only give its value to a single parameter of that component. Whereas a global variable can be assigned to transfer its value to any parameter within any component in the entire schematic. A global variable can also be assigned to any number of parameters in any number of components. The process whereby a variable is set up to control two or more circuit component parameters simultaneously is referred to as Gang Tuning. For example, in a tunable filter, capacitors (typically implemented by varactor diodes) in each filter section can be ganged or tuned together by a common control voltage so that the frequency of each section tracks correctly [1].

A global variable can be directly assigned to a component parameter or indirectly through an equation. When equations are used, the value of the variable(s) is modified according to a user defined and supplied function, and the resulting functional value is assigned to the component parameter. A LINC2 equation is made up of one or more global variables combined into a mathematical expression.

Yet another type of global variable is the Swept Variable. The most common type of circuit simulation output is some rendering of

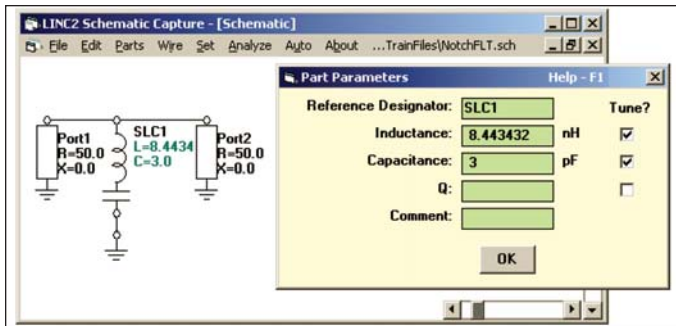


Figure 1 · Local variables L and C.

the frequency response of a circuit quantity such as gain or return loss etc. (i.e. as a function of frequency). However, a Swept Variable allows for a rendering of the circuit response as a function of the swept variable, at a specific frequency.

When the swept variable is assigned to a circuit component parameter, then the result is referred to in LINC2 as a Variable Parameter Sweep. In other words, the variable parameter sweep replaces the usual frequency sweep with a sweep of a parameter of an individual component in the circuit. For example, using tuning voltage as the swept variable, the frequency of oscillation of a voltage-controlled oscillator (VCO) can be plotted as a function of the varactor tuning voltage [2]. Executing a variable parameter sweep answers the question “what influence does this particular component or circuit parameter exert on circuit performance at a given frequency?”

Example of Local, Component Level, Variables

Figure 1 is a schematic of a simple notch filter with L and C values set to resonate at the notch frequency of 1000 MHz. Checking the Tune check box next to the part’s inductance and capacitance values creates local variables L and C for tuning and/or optimization.

Different values of L and C can achieve the same notch center frequency while changing the notch filter’s bandwidth. If the objective is to observe the changes in filter bandwidth—at a fixed center frequency—for various ratios of L and C values, we would have to vary the value of L in one direction while varying the value of local variable C in the opposite direction to bring the center frequency back to 1000 MHz.

For example, selecting the local variable L and tapping the down arrow key decreases the inductance, sending the resonant frequency higher as shown in the light curves of Figure 2A. Then, selecting the local variable C and tapping the up arrow key increases the capacitance, bringing the resonant frequency back to 1000 MHz as shown by the light curve of Figure 2B. Notice that the light curve has greater bandwidth than the original baseline (heavy) curve.

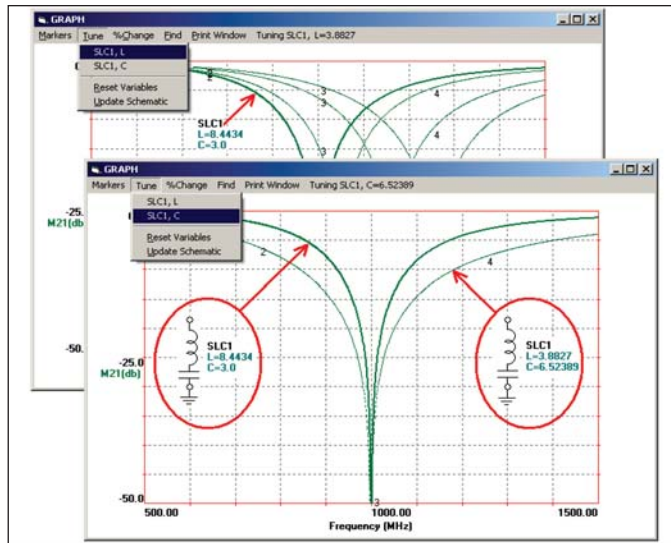


Figure 2 · A) Decreasing L increases the resonant frequency (top curves); B) Restoring resonance by increasing C (lower light curve).

Keeping the resonant frequency set at 1000 MHz while manipulating the local variables for different ratios of L and C can be a bit tedious. Doing this with global variables is much easier since one of the global variables can be used in an equation to compute the value that will maintain a fixed frequency, as will be demonstrated next.

Example of Global Variables

The schematic of Figure 3 adds two global variables and an equation to the simple notch filter schematic of Figure 1. Variable 1 (Var1 on the schematic) is named L_IND and is assigned to the inductance parameter for the series LC resonator component SLC1. Var2 is the frequency variable, Fre = 1000 MHz (in this example we treat the Fre variable as a constant and use it only to make the equation more readable).

For any value of the inductance variable L_IND, the capacitance (C_Cap) in pF required to resonate the series LC component at the specified frequency (Fre) is:

$$C_Cap = 10^{12} / ((2\pi \cdot Fre \cdot 10^6)^2 L_IND \cdot 10^{-9})$$

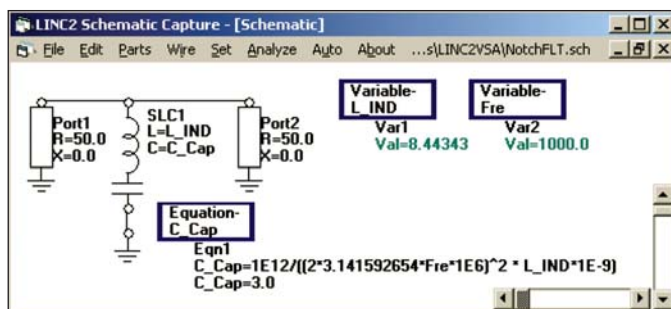


Figure 3 · Use of global variables in equation for C.

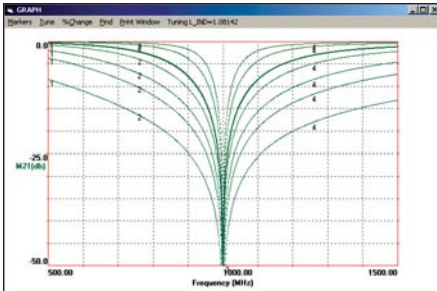


Figure 4 · Tuning resonator L while automatically recalculating resonator C for 1000 MHz resonance.

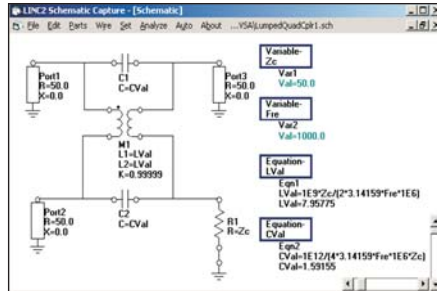


Figure 5 · LINC2 schematic setup for designing the lumped element 3 dB quadrature coupler.

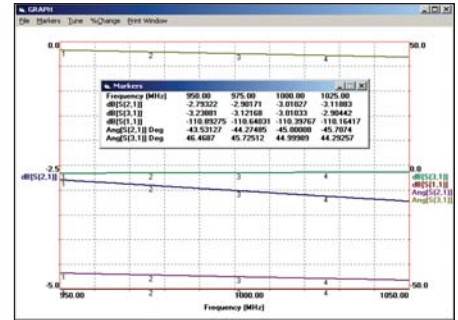


Figure 6 · Simulation of the 3 dB lumped element quadrature coupler.

Using scientific notation for numerical constants and $\pi = 3.141592654$, the equivalent LINC2 equation (Figure 3) is:

$$C_Cap = 1E12 / ((2 * 3.141592654 * Fre * 1E6)^2 * L_IND * 1E-9)$$

With the schematic set up with variables L_IND, Fre and equation C_Cap as in Figure 3, we can simply click Analyze and View > Plot to get the frequency response plots of Figure 4. Notice that as L_IND is tuned in Figure 4, all the corresponding notch response plots remain fixed about the 1000 MHz center frequency (Fre). This is a result of equation C_Cap recalculating the corresponding capacitance value required to maintain the frequency.

In this example we chose L as the independent variable for which to calculate C. We also could have made the reactance (X) the independent variable from which both L and C could be calculated according to the formulas:

$$L = X / (2\pi \cdot Fre) \quad \text{and} \quad C = 1 / (2\pi \cdot Fre \cdot X)$$

Reactance is a more natural choice for investigating matters of loaded Q and bandwidth but it requires an additional equation.

This analysis provides insight into how the loaded Q (and thus notch bandwidth) can be controlled by the inductance (which determines the reactance to load resistance ratio or $Q = X/R$). Moreover, for a specific application we may need to determine how much impact the notch has on the adjacent passband (from marker 1 through marker 2 in Figure 4). In this case we will want to pick an inductance value corresponding to one of the higher Q curves (higher and narrower plots in Figure 4).

Using Equations to Design Circuit Components Design Template for the 3 dB Quadrature Coupler

Figure 5 shows a design template for designing lumped element 3 dB quadrature couplers. The schematic is made up of two identical tightly coupled inductors (coupling coefficient K approaching 1) and two identical capacitors. Variables Zc and Fre allow the user to design

a 3 dB coupler for any practical port impedance and frequency respectively. In this example the variables are set to design the coupler for operation at 1000 MHz into 50 ohm ports.

Equations LVal and CVal implement the design formulas for automatically calculating all the component values [3]. In the simulation, the result of equation LVal is assigned to inductors L1 and L2 while the equation value for CVal is assigned to capacitors C1 and C2. Running the simulation produces the results shown in Figure 6. Clicking Analyze starts the simulation running and updates the equation values on the schematic page. Figure 6 verifies that exactly half of the input power (-3 dB) appears at ports 2 and 3 and that the phase difference between these ports is almost exactly 90 degrees at the design frequency of 1000 MHz. An excellent input match is indicated by dB[S(1,1)] less than -110 dB over the entire simulation band.

Design Template for the Lumped Matching BALUN

Figure 7A shows a design template for designing lumped element BALUNS. The schematic is made up of two identical inductors and two identical capacitors.

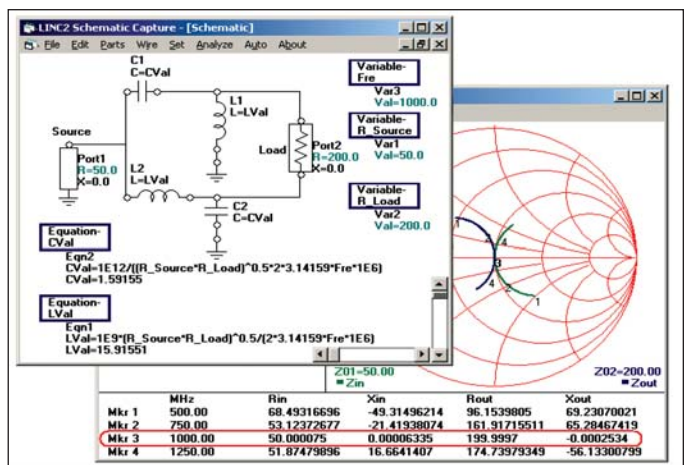


Figure 7 · A) LINC2 design template for matching BALUN circuit, B) Smith chart display of matching.

Variables R_{Source} and R_{Load} allow the BALUN to operate into any desired port impedances. The frequency variable, Fre , provides for the specification of any practical operating frequency. In Figure 7 the design has been set for matching a 50 ohm unbalanced source to a 200 ohm balanced load at 1000 MHz.

Equations L_{Val} and C_{Val} implement the design formulas for automatically calculating all the component values [3]. In the simulation, the result of equation L_{Val} is assigned to inductors $L1$ and $L2$ while the equation value for C_{Val} is assigned to capacitors $C1$ and $C2$. Running the simulation produces the results shown in Figure 7B..

Using Equations with the Optimizer to Design Broadband Distributed Matching Networks

A result of Fano's bandwidth limit formulation [4] is that an exact match into a complex load can only occur at precisely one frequency point. There are many circuit topologies derived from closed form equations that yield an exact match at exactly one frequency point. These are termed "narrowband matching" networks because, although they match at only one point, the quality of the match continues to be good (though not perfect) throughout a narrow band of frequencies about the match frequency. For simple lumped element networks, the loaded Q of the network generally determines the

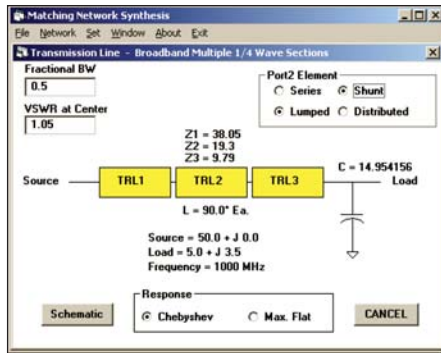


Figure 8 · Broadband matching network synthesis.

bandwidth of the match.

For wideband matching into a complex load, the Fano limit indicates that the quality of a uniformly constant match over a given bandwidth must necessarily be less than some maximum return loss value imposed by the constraint formula. Since the constraint formula can be expressed as an integral involving the reflection coefficient it can be understood that the match can exceed the uniform limit over some frequencies at the expense of a lesser match over other parts of the band. Unfortunately, for complex loads, closed form formulas for wideband matching networks that yield constant quality of match near the theoretical maximum do not exist. What does exist are methods of determining approximate solutions based on optimization or iterative calculations.

On the other hand, there are closed form solutions for wideband

matching networks that can match between purely resistive terminations. These can also be used to approximate wideband matching networks operating into complex loads if the load reactance is cancelled at the network port with an appropriate reactive element. Figure 8 shows a LINC2 synthesis program for broadband matching with quarter-wave TRL sections using this technique.

The matching problem of Figure 8 is defined as matching a 50 ohm source to a complex load of $5 + j3.5$ ohms over a band centered at 1000 MHz. Practical loads do not present a constant reactive component over a significant band of frequencies. The fact that we are attempting a broadband match to a complex load implies that the load will not stay constant at $5 + j3.5$ ohms over the band.

Making the assumption that the load reactance is constant over the band is not realistic. There are two ways to address this problem. One way is to model the load with some combination of lumped RLC elements. Another method uses measured load data in the form of a one-port impedance (z1p) file assigned to the simulation load port. The schematic of Figure 9 employs the later method of making the load port (Port 2) a file based port. The LoadZFile.z1p file assigned to the load port for this example has impedance varying over frequency according to the following z-parameters:

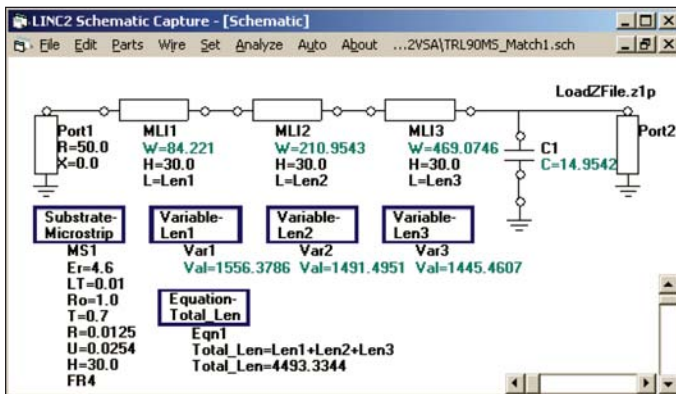


Figure 9 · Broadband matching network schematic.

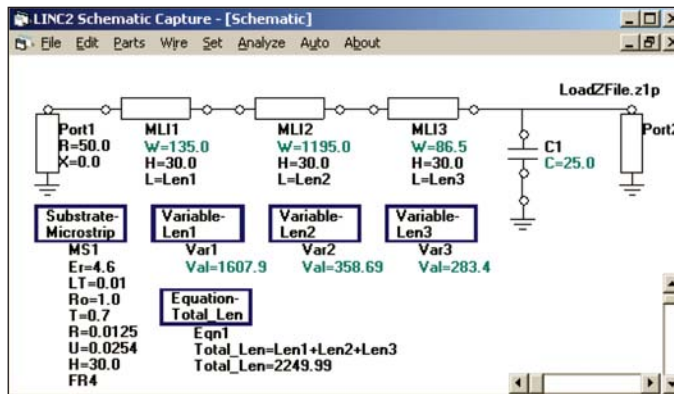


Figure 10 · Optimized broadband matching network.

#	MHz	Z	R	I	R	1
750	4.75	1				
900	4.9	2.5				
1000	5.0	3.5				
1100	5.1	4.5				
1250	5.25	6				

The load impedance data from this file is plotted on the left of each Smith chart (Figures 11 and 12) near the tail

of the red arrow. The arrow points to the matched impedances transformed by the matching network.

The schematic in Figure 9 was automatically created from the LINC2 Network Synthesis program (by clicking the Schematic button in Figure 10). Conversion of the quarter-wave (TRL) lines produced in the network synthesis program to physical

microstrip (lines MLI 1-3) was performed automatically using the LINC2 Auto > Convert To > Microstrip feature. Variables Len1, Len2 and Len3 were subsequently set up and assigned to the length parameter of each microstrip section. The equation Total_Len was added to sum up the total physical length of the matching network (Total_Len = Len1 + Len2 + Len3). Since each quarter-wave line is approximately 1.5 inches (1500 mils) on 30 mil FR4 board material at 1 GHz, the total length comes to about 4.5 inches.

Most circuit simulators have optimizers that will take selected circuit component parameters and optimize their values towards one or more user specified performance goals. An optimization goal is usually an electrical circuit response such as gain, return loss or electrical isolation. The user specifies the profile of the selected response over frequency or at some number of specific frequencies. A new capability of the LINC2 optimizer is the ability to optimize the value of equations (via user specified goals) in the same way that electrical circuit responses are optimized. The optimized equation may represent something other than an electrical property of the circuit. In fact, the equation can represent a physical property such as overall physical size, as will be shown next.

Optimizing Dimensions with LINC2 Equations to Reduce the Size of the Physical Circuit

The total length of the matching network (Figure 9) at 4,493 mils almost exactly equals the length of the author's personal cell phone. This matching network is not a compact design. In order to reduce the physical size while improving the electrical performance, an optimizer will allow the user to constrain the range (upper and lower bounds) of the microstrip dimensions during optimization.

Optimizer constraints, though

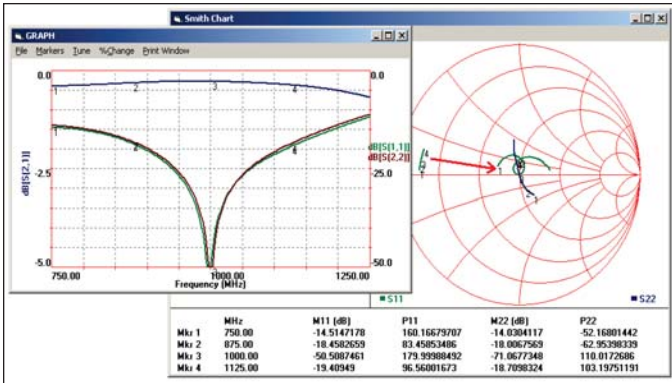


Figure 11 · Broadband matching from direct (closed-form) synthesis (before optimization).

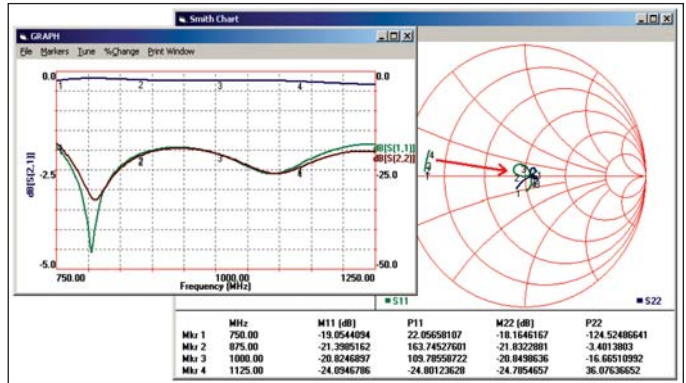


Figure 12 · Optimized broadband matching results.

necessary to enforce practical values, will almost always result in making it more difficult for the optimizer to achieve the desired goals. Unconstrained variables obviously give the optimizer the greatest freedom to find the best solution at the expense of possibly generating some impractical component values. However, optimizing this circuit without constraints on the microstrip dimensions could possibly lead to a design that is electrically superior but even longer than the initial 4.5 inches!

The ability of the LINC2 simulator to optimize equations, as well as electrical circuit responses, yields an uncommon power over the optimization process. For example, the original circuit in Figure 9 was optimized in Figure 10 without placing any constraints on the lengths of the individual microstrip sections. Yet, through the use of the Total_Len equation, it

resulted in a design exactly half the size of the original! This result was accomplished using the LINC2 optimizer by constraining only the value of the equation and not its constituent variables (Len1, Len2 and Len3). Thus, the optimizer had complete freedom to make each section any length necessary to achieve optimum performance as long as the overall length of the three sections combined didn't exceed 2,250 mils.

Figures 11 and 12 show the network's matching performance before and after optimization. The network's insertion loss and return losses are plotted over a 50% bandwidth centered at 1000 MHz. Note in Figure 13 that the optimization goals include the network's total physical length in addition to network insertion loss (M21) and return loss (M22). The aggressive goal of cutting the total length in half was enforced by

applying an optimization weight several times larger than the other goals. All of the network's component values were allowed to vary except elements of the substrate definition.

The LINC2 Network Synthesis program provided an excellent seed circuit for the optimizer. Even before optimizing, Figure 11 indicates that direct circuit synthesis produced a capable matching network with return loss greater than 20 dB over 40% of the band and 14 dB or more over 90% of the band. Yet the power of the optimizer is demonstrated by the improved match displayed in Figure 12. Return losses are maintained between approximately 20 and 25 dB across the band with additional flattening of the insertion loss response—all while at the same time meeting the aggressive goal of reducing the overall physical length by 50%.

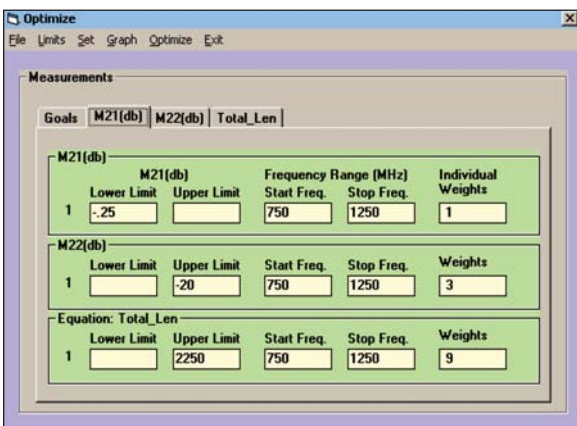


Figure 13 · LINC2 optimization goals.

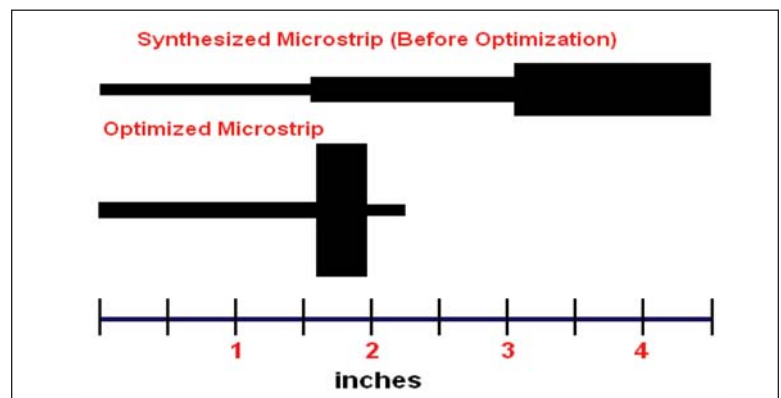


Figure 14 · Microstrip length comparison before and after optimization.

Swept Variables

Figure 1 shows the schematic for a simple fixed frequency notch filter, while Figure 2 displays the frequency response as discussed earlier. Now, consider the similar schematic in Figure 15 where we have replaced the fixed value capacitor with a variable capacitor in the form of a varactor tuning diode. We have thus changed the fixed frequency notch filter into a voltage variable tuned notch filter. The equation CVar models the voltage-to-capacitance characteristics of the varactor diode. The capacitance, CVar, is thus calculated as a function of the varactor tuning voltage (variable Varactor_V) and assigned to the capacitance element in the series LC (SLC1) component.

For the variable Varactor_V, the Variable Parameter Sweep check box is checked in the variable's Part Parameters dialog box. This action changes the variable from an ordinary variable to a swept variable. The range of values over which the variable sweeps is defined by start and stop values (ST and SP respectively). The resolution of the value steps is controlled by the number of points (PT). In this case, the varactor voltage is swept from 0.5 to 5.5 volts at a resolution of about 0.1 volts per step (PT = 51).

Figure 15 plots the tuning performance of the filter. Note that the varactor voltage (Varactor_V) is the independent swept variable here (as displayed on the horizontal axis in Figure 15), rather than the usual frequency sweep with frequency on the horizontal axis as in Figure 2. This allows us to focus on the filter's tuning performance (left axis in Figure 15); the span of frequencies tuned relative to the varactor's tuning range. The varactor's voltage-to-capacitance transfer function is also displayed in Figure 15 (plotted against the right axis). It is immediately evident from the plots that the filter can be tuned from 450

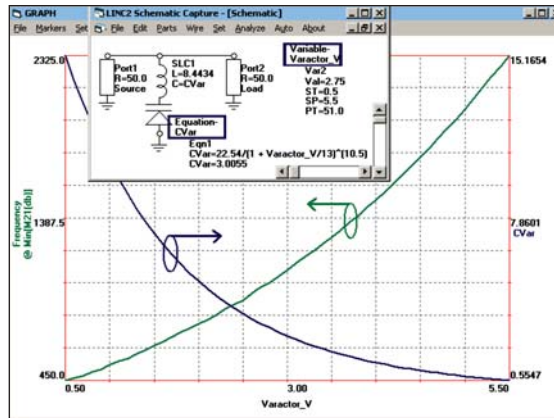


Figure 15 · Varactor tuned notch filter.

MHz to 2325 MHz for a tuning voltage ranging between 0.5 and 5.5 volts (corresponding to the varactor capacitance ranging between 0.55 and 15.2 pF). References [1] and [2] describe applications of swept variables.

Summary

Variables and equations can be placed on the schematic page along with the actual circuit schematic. When these equations are formulated to calculate component values from variables whose values are input by the user to capture the design specifications or operational aspects of the circuit (frequency of operation, port impedances, circuit Q etc.), the result is a complete design template that can be used repeatedly. Simply changing the input variables instantly creates a circuit uniquely adapted to the new application. The equations and variables help to document the design and become a record of how the circuit's component values are calculated. This kind of self documenting schematic will be better understood when the schematic file is later reloaded.

This article also demonstrated the power of a new optimizer capability, having the ability to control the physical size of the circuit. With this new capability the same optimizer that optimizes RF circuit performance can also be simultaneously directed to constrain or reduce physical size.

The LINC2 Pro software used in these examples is a high performance RF and microwave design and simulation program from ACS (Applied Computational Sciences, Escondido, CA). In addition to circuit simulation, optimization and statistical yield analysis, LINC2 Pro includes features for automating design tasks, including circuit synthesis and equation based modeling and optimization. The same techniques presented here for modeling discrete circuit component parameters through the use of user defined equations are also available in the ACS Visual System Architect (VSA) for modeling system components at the system level. For more information visit the ACS website at www.appliedmicrowave.com.

References

1. Dale D. Henkes, "Software Enhances the Design and Analysis of Tunable Circuits", *High Frequency Electronics*, July 2008.
2. Dale D. Henkes, "LINC2 Software Enhances the Design and Analysis of VCOs and Other Tunable RF and Microwave Circuits", *Microwave Product Digest*, July 2008.
3. Peter Vizmuller, *RF Design Guide: Systems, Circuits and Equations*, Artech House, 1995.
4. R. M. Fano, "Theoretical limitations on the broadband matching of arbitrary impedances". *J. Franklin Inst.*, Vol. 249, Jan./Feb. 1950: pp. 57-83, 139-154.

Author Information

Dale D. Henkes is the owner of Applied Computational Sciences (ACS), LLC, with more than 25 years of professional experience in RF design/electrical engineering. He earned his B.S. degree in engineering at Walla Walla College and is a member of the IEEE Microwave Theory and Techniques Society. He may be contacted by email at: LINC2@appliedmicrowave.com.