Exploring RF

Conducted By David Newkirk, WJ1Z
Senior Assistant Technical Editor
e-mail: dnewkirk@arrl.org

Now that *ARRL Radio Designer* is on the street, we're hoping that ham interest in exploring radio-frequency (RF) technology through computer-based modeling will take off in a way that hasn't been possible before. This column is part of taking that show on the road—a place for suggesting *ARRL Radio Designer* applications we hope will be interesting to many hams.

But this column won't stop there. Many useful electronic and RF circuit design programs have been with us for years in various forms, and new ones keep appearing. For starters, I'm thinking of the various commercial products based on the University of California at Berkeley's *SPICE* (Simulation Program with Integrated Circuit Emphasis)—at least one of which, MicroSim Corp's *PSpice*, is available free in a limited-function evaluation version, and other of which, *MICRO-CAP*, is available jointly from Addison-Wesley Publishing and Benjamin/Cummings Publishing in a low-cost student edition. Just sketching the abilities of the many *SPICE* derivatives would take several pages. A diskful of antenna, transmission-line and propagation software now ships with every new *ARRL Antenna Book* sold, and software is also available with *The ARRL UHF/Microwave Experimenter's Manual*—what can *those* programs do? And what about the freeware and shareware that's almost everywhere? There's a lot to be excited about if you're interested in using computer-aided design (CAD) and computer-aided engineering (CAE) in Amateur Radio, and *QST* will share that excitement through Exploring RF.

Optimizing Circuit Performance with *ARRL Radio Designer*

One of the *ARRL Radio Designer* features we're most excited about is the program's ability to *optimize* circuits—to make circuits work better, or to modify circuit performance according to specified goals. I suggested in October *QST*'s *ARRL Radio Designer* article[1] that one use for this feature might be to tweak a 2-meter receiver preamp for a better noise figure, but that's a pretty far-out example for many of us. Then I went on to describe an antenna-tuner optimization that's a very nice example of optimizing a circuit almost, but not exactly, like the one actually shown.[2]

Elsewhere in this issue, Andy Griffith's "Getting the Most Out of Your **T**-Network Antenna Tuner" includes some eye-opening findings on just how lossy a **T** network can be when transforming some impedances to 50 Ω. *ARRL Radio Designer*, manipulated by yours truly, backed up Andy's findings and even provided a few graphics. Here's how I did it—and how *you* can use *ARRL Radio Designer*'s optimizer to simulate adjustable **T** networks, too.

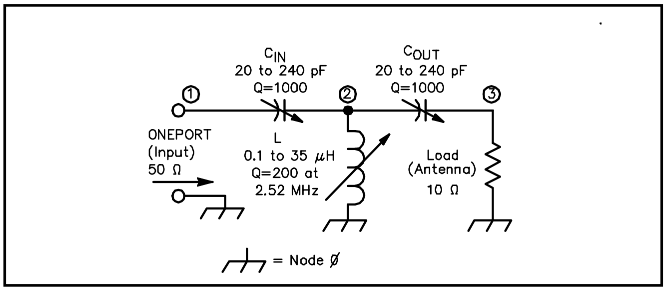


Figure 1—Many of us use a capacitor-inductor-capacitor **T** network similar to this one as an antenna tuner. With an SWR or reflected power meter acting as an indicator, we adjust its C and L values to transform the antenna's impedance (here, 10 Ω, resistive) to 50 Ω, resistive (50+*j*0 Ω when written in rectangular notation) at our frequencies of interest. (I specify L's Q at 2.52 MHz to tie in with lab Q-meter conventions, as pages 5-7 and 5-8 of *The ARRL Radio Designer Manual* explain.) The circled numbers indicate nodes for *ARRL Radio Designer* analysis, which treats the tuner as a one-port network imaginatively named `ONEPORT`.

The **T** Network in Schematic and Netlist Form

Figure 1 shows Andy's **T** net's schematic with a few additions and changes. Here's the corresponding *ARD* netlist block:

```
BLK
 CAP 1 2 C=?20PF 82.1733PF 1000PF? Q=1000 ; C/IN\
 IND 2 0 L=?0.1UH 30.2528UH 35UH? Q1=200 F=2.52MHZ;L
 CAP 2 3 C=240PF Q=1000 ; C/OUT\
 RES 3 0 R=10 ; this is the load
 ONEPORT:1POR 1
END
```

Question marks bracket the component values we want to optimize. Note that instead of just specifying a single value, we've indicated *ranges* for C_{IN} and L. (In keeping with Andy Griffith's suggested tuning technique, we start out by setting C_{OUT} to a fixed value and leaving it alone.) In *ARD*-speak, this *constrains* their values to the ranges shown in Figure 1. Note also that the optimizer won't touch the Q and F values associated with C_{IN} and L. This is appropriate, considering that we can't adjust them at will in a real antenna tuner.

We're interested in seeing how the network plays at 160 meters, so we'll set up a frequency block that combines good resolution across the band with enough wide-range exponential steps for graphing the network's harmonic reduction:

```
FREQ
 STEP 1.8MHZ 2.0MHZ 1KHZ
 ESTP 0.5MHZ 30MHZ 300
END
```

Finally, we add an optimization (`OPT`) block that tells *ARD* which circuit to optimize—a netlist may contain more than one, after all—at what frequency or frequencies, in what way, and to what degree:

```
OPT
 ONEPORT
 F=1.9MHZ
 RZ11=50
 IZ11=0
 TERM=1E-5
END
```

This block tells *ARRL Radio Designer* to optimize the performance of the circuit block `ONEPORT` so that, at 1.9 MHz, the real (resistive) part of its impedance is 50 Ω and the imaginary (reactive) part of its impedance is *j*0 Ω; that is, 50+*j*0 Ω. The `TERM=1E-5` line tells *ARD* to terminate optimization—that is, to consider the optimization goals reached—when the optimizer's *error function* reaches or drops below 1×10^{-5}, or 0.00001. The `OPT` and `END` statements merely identify the block and delimit it so *ARRL Radio Designer* knows where it starts and stops. Table 1 shows the corresponding *ARD* netlist, all set to go.

This column isn't the place for excruciating detail about exactly how *ARD* calculates the optimizer's error function. (*The ARRL Radio Designer Manual* explains how optimization goals factor into the error function, and how you can specify their *weight*—the degree to which they contribute to the number.) To get started with optimization, all we need to know is that the greater the error function, the farther we are from a successful optimization.

Running It

Once we've got our optimizable file ready to go, we click on **Optim** (or press Shift+F10). *ARRL Radio Designer* analyzes the circuit first, and then pops its Optimization and Optimization Data dialogs (Figure 2).[3]

When the optimizer succeeds in pushing its error function to a number equal or less than the OPT block's `TERM` value—or to 0 if you didn't specify one—it halts, and the message "Optimization Completed" appears in the Optimization dialog.[4] Click the **Done**

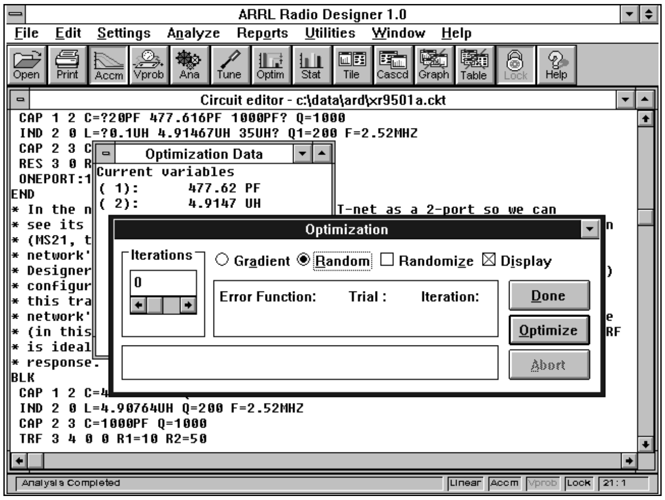


Figure 2—Its pre-optimization analysis complete, *ARRL Radio Designer* pops its Optimization and Optimization Data dialogs. All we need to do now is enter a number in the Iterations edit control (I always enter 1000), clear the **Display** box (*ARD*'s optimizer runs much faster when it doesn't stop to do display calculations between iterations), and click **Optimize**. Leaving the optimizer set to its **Random** default is best for now; we'll have to discuss random versus gradient optimization some other time.

Table 1

Simulating an Adjustable T-Network Antenna Tuner with *ARRL Radio Designer*

```
BLK
 CAP 1 2 C=?20PF 82.1733PF 1000PF? Q=1000 ; C/IN\
 IND 2 0 L=?0.1UH 30.2528UH 35UH? Q1=200 F=2.52MHZ;L
 CAP 2 3 C=240PF Q=1000 ; C/OUT\
 RES 3 0 R=10 ; this is the tuner's load
 ONEPORT:1POR 1
END
FREQ
 STEP 1.8MHZ 2.0MHZ 1KHZ
 ESTP 0.5MHZ 30MHZ 300
END
OPT
 ONEPORT
 F=1.9MHZ
 RZ11=50
 IZ11=0
 TERM=1E-5
END
```

button. "Optimization complete," responds *ARRL Radio Designer*. "Do you want to update circuit?" Click **Yes** (if you don't want to keep and use the updated values, click **No**), and *ARD* updates each optimizable component's initial nominal value with the new value found by the optimizer. A cheerful "WARNING - REANALYSIS IS REQUIRED AFTER OPTIMIZATION" message appears in the screen's lower left corner. When you see this, just click the **Ana** button or press F10.

If we want to determine the tuner's network's frequency response, we need to reconfigure it as a two-port network to see how its amplitude response varies with frequency. Table 2 shows how to do that, and Figure 3 graphs the analysis results based on this approach.

Specifying Reactive Antenna Loads

So far, we've used optimization only in transforming purely resistive loads to 50 Ω resistive—50+*j*0 Ω. In the ONEPORT block for our **T**-net circuit file, we specified the load with the netlist line RES 3 0 R=10—a 10+*j*0-Ω load. Often, however, the antenna-system load we want to transform to 50+*j*0 Ω will be reactive. Its

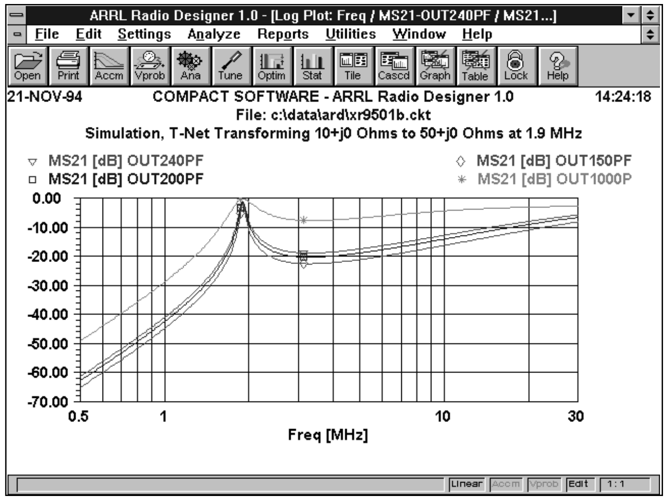


Figure 3—The 160-meter tuner's frequency response for 10+*j*0 to 50+*j*0 Ω transformations based on various C_{OUT} settings. I used netlist blocks like that shown in Table 3 to do these analyses. If harmonic rejection is what you're after, the capacitor-inductor-capacitor **T** network isn't the way to get it.

Table 2

The T Network Reconfigured as a Two-Port for Frequency-Response Simulation (Values for a 240-pF C_{OUT} Shown)

```
BLK
 CAP 1 2 C=122.811PF Q=1000
 IND 2 0 L=19.3847UH Q=200 F=2.52MHZ
 CAP 2 3 C=240PF Q=1000
 TRF 3 4 0 0 R1=10 R2=50
 OUT240PF:2POR 1 4
END
```

This block replaces ONEPORT's RES line with an ideal transformer (TRF) connected between nodes 3 and 4, with node 0 common for both windings. The phrase R1=10 R2=50 defines the transformer as presenting 10 Ω to the network when its secondary is terminated by 50 Ω—which it will be when we plot OUT240PF's response as the S-parameter MS_{21} using *ARD*'s default 50+*j*0 Ω terminations.Note: Use this block in addition to, not in place of, the ONEPORT block in Table 1.

Table 3

Simulating a T-Network Tuner Feeding a Reactive Antenna at 7.2 MHz

```
BLK
 CAP 1 2 C=?20PF 100PF 240PF? Q=1000 ; C/IN\
 IND 2 0 L=?0.1UH 30.2528UH 35UH? Q1=200 F=2.52MHZ;L
 CAP 2 3 C=240PF Q=1000 ; C/OUT\
 ONE 3 0 ANTENNA
 ONEPORT:1POR 1
END
FREQ
 STEP 7.1MHZ 7.3MHZ 1KHZ
 ESTP 1MHZ 30MHZ 300
END
OPT
 ONEPORT
 F=7.2MHZ
 RZ11=50
 IZ11=0
 TERM=1E-5
END
DATA
 ANTENNA: Z RI
 60 +60
END
```

j (imaginary) part will be a nonzero positive or negative number. How do we specify that?

If we happen to know what series or parallel combination of resistance and capacitance (or inductance) will exhibit that load at our frequency of interest, we can connect them between nodes 3 and 0 of `ONEPORT` and optimize away. If we don't, *ARRL Radio Designer*'s ONE element, a programmable one-port "black box," can serve.

ONE is a neat solution because we can specify its characteristics in terms of S, Y or Z parameters. Using ONE is a bit more involved than just coding in a few components, however, because we need to specify it *and its impedance data* in the netlist. That means our netlist will have to include a `DATA` block.

Table 3 puts it all together. The netlist line `ONE 3 0 ANTENNA` connects a ONE element between nodes 3 and 0 and labels it `ANTENNA`. The label is important because *ARRL Radio Designer* needs it to link this particular ONE—a more complex netlist might contain several different ONEs, after all—with its data in the `DATA` block:

```
DATA
 ANTENNA: Z RI
 60 +60
END
```

This `DATA` block tells *ARD* to provide impedance (Z) data in real-imaginary (RI) form for the element labeled `ANTENNA`—in this case, an inductively reactive load of 60+*j*60 Ω at all frequencies.[5] The modeling's yours to do, but here's a hint: The optimizer won't reach its goal with C_{OUT} fixed at 240 pF. You'll have to try a lower value, or make C_{OUT} optimizable.

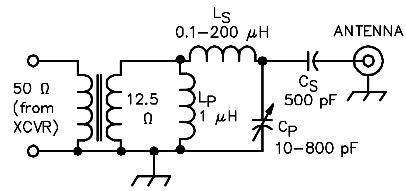
Free Electronic Goodies

You can get expanded versions of the Table 1 and Table 3 netlists, as well as the one I used to generate Figure 3, in the file EXRF9501.TXT, available from the ARRL HQ BBS (203-666-0578) and Internet info server (*info@arrl.org*). As always, the latest *ARRL Radio Designer* news is also available from those sources in the file ARD.TXT.

Notes

[1]David Newkirk, WJ1Z, "Introducing ARRL Radio Designer: New Software for RF Circuit Simulation and Analysis," *QST*, Oct 1994, pp 21-26.

[2]The "oops" is that I misspecified one of the network's capacitors in the netlist. *ARRL Radio Designer* found a match with C_P, shown in the article's Figure 9 as connected to common on the left side of L_S, actually connected to common from the junction of L_S and C_S. So the network I actually optimized was:



and not *exactly* the tuner described by Rohde. Having reoptimized the circuit after finding and fixing that error, I can now report that the good news is that Rohde's network can indeed transform a 1.6–*j*2246 Ω load to 50+*j*0 Ω at 1.83 MHz if modified as shown above, but that the bad news is that it *can't* transform that load to 50+*j*0 Ω at 1.83 MHz as it was originally specified by Rohde. Live and learn. The corrected circuit file is now part of the information file ARD.TXT, available from the ARRL BBS (203-666-0578) and the ARRL Technical Information Service's Internet-accessible information server (*info@arrl.org*).

[3]If you've set *ARD*'s Auto Reporter to pop Report Editor or Quick Reporter after every analysis, the one you've selected will pop before the Optimization dialogs appear, and you may even have to close it or move it to work with the Optimization dialogs. Frankly, Auto Reporter irks me—I can't stand that big Linear Reports dialog popping when all I want to do is load predefined reports, which pressing F7 after analysis handily facilitates—so I keep Auto Reporter set to None.

[4]Well, okay—*ARD*'s optimizer also stops if it completes the specified number of iterations before finding a solution. The thing to do then—depending on whether you think a solution is possible—is to specify more iterations. If the error function hasn't been dropping through many iterations, a successful solution may not be possible within the value limits set by the netlist.

[5]If we don't specify a frequency for the ONE data, *ARRL Radio Designer* assumes that we mean "at all frequencies"—and this is important. Just as the purely resistive tuner load we've specified so far is unrealistic because no antenna exhibits a constant, pure resistance across even a small frequency range, a reactive load that doesn't vary with frequency is also unrealistic. *ARRL Radio Designer* lets us specify frequencies for black-box-element data, so if we happened to know our antenna's actual impedance across the 1.8 to 2.0-MHz range, we could exactly specify it in the DATA block with multiple frequency-specific lines. Note, though, that it's perfectly okay to use a frequency-nonspecific impedance value in adjusting/optimizing a tuner at any one frequency.

New Books

SHORTWAVE RADIO LISTENING FOR BEGINNERS

and

THE SHORTWAVE LISTENER'S Q & A BOOK

By Anita Louise McCormick, KA8KGI

Published by TAB Books, Division of McGraw-Hill Inc, Blue Ridge Summit, PA 17294-0850; 9¼×7¼ inches, black-and-white photos and drawings. Shortwave Radio Listening for Beginners, First edition 1993, 176 pp, $10.95; The Shortwave Listener's Q & A Book, First edition 1994, 144 pp, retail price $12.95.

Reviewed By Brian Battles, WS1O
QST *Features Editor*

With the publication of these two books, it appears that Anita McCormick is establishing herself as America's "First Lady of Shortwave Listening." She certainly logs an enormous amount of shortwave radio listening (SWLing) time from her home in Huntington, West Virginia, and she's enthusiastic and knowledgeable in her work to bring the hobby into the home of anyone who reads her books.

Shortwave Radio Listening for Beginners is a clear, nontechnical introduction to SWLing for anyone who has yet to venture into the airwaves outside the amateur bands. Anita briefly covers the beginnings of radio and a little bit of broadcasting history, and then goes on to discuss domestic AM broadcast DXing, VHF/UHF scanning, ham radio and more. She lists popular frequencies and modes for worldwide reception of broadcasters, utility, relay, government, pirate and clandestine stations. There are tips on the kinds of receivers, antennas and accessories to consider for your home listening station, and on collecting QSL cards and other mementos and souvenirs from SW broadcasters. Amateurs who operate HF probably already know most of the necessary theory regarding HF antennas, reception and equipment, although you may find a few pointers about the slightly different requirements of setting up an effective SW monitoring station.

It's also gratifying to see her mention the growing use of ham radio in schools and through the US space program, such as shuttle communications. The educational aspects of Amateur Radio are important, especially in a book for listeners, because stumbling across ham communication while tuning for other radio services is often a young person's first exposure to Amateur Radio.

The final chapter covers radio waves, propagation and other such information that may help a listener better understand how and why he can hear different signals at different times, and help determine the best antenna(s) and listening times to hear the stations of interest.

Three adequate appendices list SW stations, radio listening clubs and radio sources. There's a handy glossary and reasonably comprehensive index.

Although it's complete as a standalone work, *The Shortwave Listener's Q & A Book* is a handy companion volume. It depends on whether you prefer a straight text or question-and-answer format. It covers essentially the same topics, but organized in a style that might be more appealing to some readers.

Both books are illustrated with many black-and-white photographs and helpful diagrams, and the writing style is friendly and nontechnical. So if you know a new ham or nonham who has yet to be introduced to listening to shortwave radio and scanners, one or both of these books could make a fine gift…or as a loan from your personal radio library. At just a little more than 10 bucks each, I rate these books a good value for someone starting to get curious about the signals waiting all across the busy RF spectrum.