Exploring RF

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

Transistor Modeling, Part 3: Constraints in Optimization, Two-Port Data with Noise Parameters, and Introducing *ARRL Radio Designer 1.5!*

In July 1995 Exploring RF, we did a pretty good job of optimizing an *ARRL Radio Designer* (*ARD*) bipolar junction transistor (BIP) element to make it act like an *ARD* two-port element (TWO)—a TWO characterized by manufacturer-supplied *databank* data for an actual transistor. This gave us a transistor model usable outside the frequency limits of the databank data used to generate it—a neat thing to be able to do, especially considering that manufacturers usually don't characterize today's small-signal RF transistors at frequencies below a few hundred megahertz.

Notice, though, that I said *pretty good.* I said that because I neglected to do one very important thing in running that BIP optimization: Believing I had determined that the optimizer could not be kept from immediately driving the BIP's alpha (A) to a value equal to or greater than 1 (illegal for a real transistor) if A were made optimizable (`A=?.99?`), I hard-coded A to a realistic, yet unoptimizable, value (0.98) and went on to achieve the results shown in last column's Figures 1 and 2.

But it turns out that an A of 0.98 was my lucky guess. A transistor's gain is one of its most important characteristics, so what I *should* have done with A was make it optimizable and *constrain*—set hard limits for—the range of A values the optimizer could try in finding its solution. Doing so would let the optimizer adjust A for best results while keeping its value realistic.

We can constrain optimizable *ARD* values in several ways. An *unconstrained* value takes the form `?NominalValue?`—one value bracketed by question marks—as in `R=?50KOH?`, or `A=?.99?`. During gradient optimization, *ARD* can replace `NominalValue` with any value from 0.0 to infinity; during random optimization, *ARD* can replace `NominalValue` with any value from 0.0 to no more than twice `NominalValue`—a range that's considerably smaller than that available during gradient optimization. Either way, we now can see why specifying a BIP's optimizable alpha as `A=?.99?` could result in unrealistic values of A: During gradient optimization, A's upper limit would be infinity; during random optimization, A's upper limit would be 1.98. Since A must be less than 1 in any real transistor, both values are illegal. Luckily, *ARD*'s BIP modeling is too smart to let us get away with either!

A *constrained* value takes the form `?Value NominalValue Value?`, as in `C=?10PF 100PF 500PF?`. *ARRL Radio Designer* handles this spec the same way in random and gradient optimization: Any value from 10 to 500 pF, inclusive, is fair game, and the game starts with the nominal (100 pF). As a variation on this, *relatively constrained* values take the form `?Value% NominalValue Value%?`, as in `R=?50% 50 300%?`, which specifies a resistance from 25 to 150 Ω, inclusive, starting with 50 Ω. *ARD* handles relatively constrained values the same way in random and gradient optimization.

Unconstrained optimizable values may seem superficially simpler to use than constrained values, but they are actually trickier to use if we want to be able to switch smoothly back and forth between random and gradient optimization in solving a problem. (We may well want to do just this if gradient optimization bogs down in a local minimum, or if random optimization seems to be taking several lifetimes to zero in on its target.)

Just what do I mean by "trickier"? Random optimization with unconstrained values can fail to find a solution *if the solution happens to require that any of the optimizable values be increased to more than twice their nominal values.* If, for instance, we specify even one capacitor in an optimizable circuit as `C=?100PF?` and that capacitor's value for a successful solution must ultimately be 432 pF, *we'll never get there with random optimization* because *ARD* can't increase that unconstrained value to more than 200 pF. (We'd probably suspect something was up after noticing that *ARD* had stalled with this component's value right at 200 pF.) Switching to gradient operation could dig us out of the hole—if we didn't bog down in a local minimum, that is—because the gradient optimizer could increase the capacitor's value to infinity, as opposed to the 200 pF available to us in random optimization.

All things considered, if you're new to optimization with *ARRL Radio Designer*, it's probably best to use constrained rather

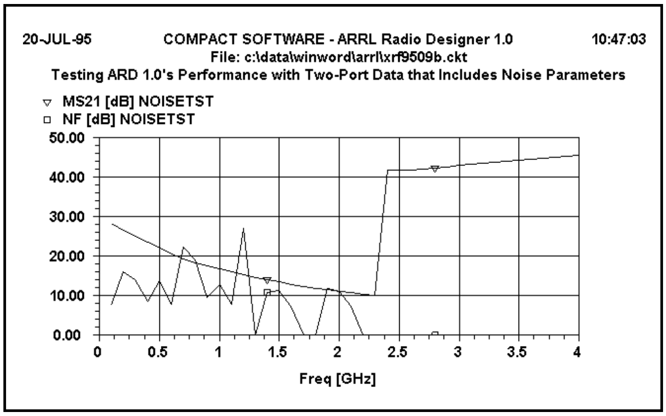


Figure 1—Graphing the performance of a two-port element (TWO) characterized by two-port data that includes noise parameters suggests that all is not well with active-TWO noise modeling in *ARRL Radio Designer 1.0*. No real, useful device would exhibit such extremes and fluctuations in forward transmission gain (MS_{21}) and noise figure (NF). Figure 2 shows how these traces *should* look.

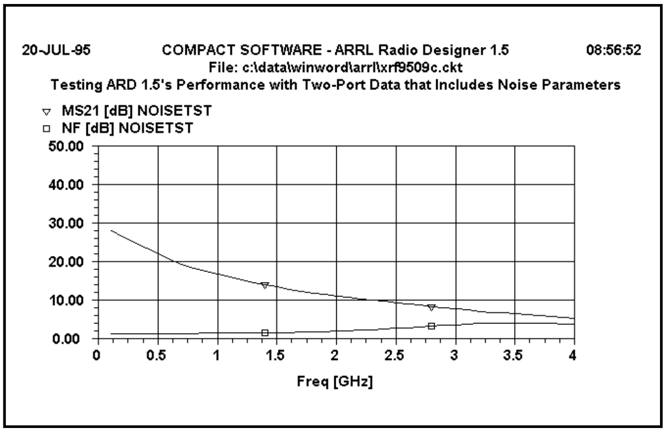


Figure 2—Now, *this* is more like it! Figure 1 shows unbelievable MS_{21} and NF values because of a bug in *ARRL Radio Designer 1.0*'s routine for using noise-parameter data to model active-TWO noise. *ARRL Radio Designer 1.5*, now shipping to new users and available as a maintenance release for those already using *ARD 1.0*, solves the problem.

Table 1

Constrained Values for a More Optimizable BIP

```
BLK
 BIP 1 2 0 A=?.85 .9 .999?  RE=?0.5 1 15?
 + F=?0.1GHZ 1GHZ 10GHZ? T=?0.005NS 0.01NS 0.1NS?
 + CE=?0.01PF 0.1PF 5PF? CI=?0.005PF 1PF 10PF?
 + RC=?100 1000 10000? RO=?0.001 0.1 5?
 + CO=?0.001PF 1PF 10PF? RB1=?0.005 1 10?
 + RC1=?1E-7 1E-6 1E-5? RB2=?0.001 0.1 5?
 + CBE=?0.005PF 5PF 10PF? CBC=?0.001PF 0.1PF 5PF?
 + CCE=?1E-5PF 1E-4PF 1E-3PF?
 + LB=?1E-7NH 1E-6NH 1E-5NH?
 + LC=?1E-5NH 1E-4NH 1E-3NH?
 + LE=?0.001NH 0.1NH 1NH?
 OPTBIP:2POR 1 2 0
END
```

than unconstrained optimizable values—constrained values specified to cover ranges that far exceed your reasonable expectations of what's necessary for success. Then, you'll be freer to switch between random and gradient optimization at will, and updating your circuit with the results of a run of either type won't hobble subsequent runs of the other. (Using large *unconstrained* values doesn't guarantee the same freedom, by the way. If, at some point during a series of optimizations, you tell the optimizer to update your circuit file with the latest values found by the optimizer, *ARD* may write at least one value that's less than half of the value necessary for success. If this occurs, any subsequent random optimizations will stall short of solution until you modify the culprit value[s] by hand or switch to gradient optimization.)

Table 1 shows this constrained-value doctrine put to work. It's just the `OPTBIP` circuit from July 1995 Exploring RF's Table 1 modified to use constrained values.

It turns out that optimizing this constrained-values BIP with the same MRF581 data we used last time doesn't give results that significantly differ from those achieved with `A=.98`—that's how good my guess for alpha was! In general, though, you'll get better results in optimizing *ARD* transistor elements to behave like databank-data devices if you turn the optimizer loose on *every* parameter you use in your BIPs and FETs, suitably constraining them to keep your optimization on the rails.

Active-TWO Noise Modeling with *ARRL Radio Designer*

In the past two Exploring RFs, we've highlighted *ARRL Radio Designer*'s ability to use two-port S-parameter data in simulating transistor behavior. In the May column, we explored just plugging some basic MRF586 data into a netlist and calling the job done. In the July column, we learned how to optimize a BIP (or FET) on the basis of databank data—a principal value in this being that we gain a transistor model that's frequency-transportable.

Using an optimized BIP or FET instead of the raw two-port data we've used so far does something else for us, though: It lets us use these transistor models' ability to simulate noise.[1] Accurately simulating noise is important in radio design because successful radiocommunication requires that we keep winning the ongoing battle between signal and noise.

ARD can't use the basic parameters-versus-frequency databank data we've used so far for modeling transistor noise because it's too smart to. If *ARRL Radio Designer* senses that two-port data represents a *passive* (lossy) network, it can determine that network's noise contribution (on the basis of thermal modeling) usefully enough, but if the data represents a network with gain—an *active* network—*ARD* intentionally *does not* attempt to model that network's noise. What may be inside an *active* black box is just too unpredictable for simple thermal modeling to work.

Table 2

Databank Data with Noise Parameters

```
0.1GHZ .74   -40 25.80 156 .017    81  .93  -14
0.5GHZ .61 -126 12.46 108 .035    44  .57  -31
1.0GHZ .57 -161   6.80  87 .040    38  .46  -33
1.5GHZ .57 -180   4.67  75 .052    47  .43  -34
2.0GHZ .58  166   3.55  64 .060    54  .41  -38
2.5GHZ .59  160   2.92  59 .068    58  .40  -39
3.0GHZ .61  150   2.42  50 .080    63  .39  -46
3.5GHZ .62  142   2.09  41 .091    61  .41  -54
4.0GHZ .62  134   1.84  32 .106    59  .42  -62
NOI RN
0.1GHZ 1.3   .12      5 .17
0.5GHZ 1.3   .10     25 .17
1.0GHZ 1.4   .06     50 .16
2.0GHZ 1.7   .25    172 .16
4.0GHZ 3.0   .48   -131 .24
END
```

The device is an Avantek AT-41485N operating with a V_{CE} of 8 V and an I_C of 10 mA.

Many RF transistor manufacturers, however, especially those who produce parts designed for frequencies at which every fraction of a decibel of noise figure counts, also supply databank data that characterizes their devices' noise performance in addition to just listing the devices' S parameters by frequency. Table 2 shows two-port data of this type for an Avantek AT-41485N bipolar junction transistor operating with a V_{CE} of 8 V and an I_C of 10 mA.

The noise-parameter part of the data—

```
NOI RN
0.1GHZ   1.3  .12      5  .17
0.5GHZ   1.3  .10     25  .17
1.0GHZ   1.4  .06     50  .16
2.0GHZ   1.7  .25    172  .16
4.0GHZ   3.0  .48   -131  .24
```

—lists, by frequency, the AT-41485N's minimum noise figure in dB (F_{min}), the magnitude and phase of the AT-41485N's optimum noise input reflection coefficient (G_{opt}), and the AT-41485N's equivalent noise resistance (R_n). (For details on the form of this data, see page 17-12 of *The ARRL Radio Designer Manual.*) A simple netlist (Table 3) can show us how this data plays; all it does is characterize an *ARD* two-port element (labeled AVANTBIP) with the AT-41485N data.

Figure 1 graphs the results in terms of forward transmission gain (MS_{21}) and noise figure (NF)—and it's obvious that something's seriously wrong. The noise figure trace oscillates between wacky, numerical-noise-ridden values for half of the graph and then, just as unrealistically, plummets to zero for the remainder. MS_{21} looks believable until the noise figure drops to zero and stays there, and then takes off for the stratosphere.

Figure 2 shows how our AT-41485N model is *supposed* to work. The difference is that *ARRL Radio Designer 1.0*, the noise modeling ability of which is *buggy* when two-port data with noise parameters comes into play, produced Figure 1, and *ARRL Radio Designer 1.5*, which we're already shipping in response to *ARRL Radio Designer 1.0* orders, produced Figure 2!

ARRL Radio Designer 1.5 and How to Get It

We're calling *ARD 1.5* a *maintenance release* because it's mainly intended to fix this darned "two-port data with noise parameters" bug.[2] It does include a few enhancements, though, such as 3-D dialogs and controls (Figure 3).

---

[1] *ARRL Radio Designer* models noise on the basis of principles described in G. Vendelin, A. M. Pavio, and U. L. Rohde, *Microwave Circuit Design Using Linear and Nonlinear Techniques* (New York: John Wiley and Sons, 1990), and R. A. Pucel and U. L. Rohde, "An Accurate Expression for the Noise Resistance R_n of a Bipolar Transistor for Use with the Hawkins Noise Model," *IEEE Microwave and Guided-Wave Letters*, Vol 3, No. 2, Feb 1993, pp 35-37.

Table 3

Testing *ARD 1.0*'s Performance with Two-Port Data that Includes Noise Parameters

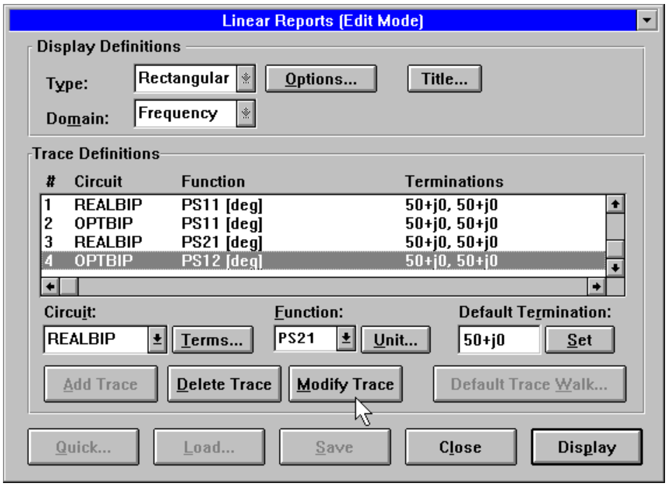
```
BLK
  TWO 1 2 0 AVANTBIP
  NOISETST:2POR 1 2
END
FREQ
STEP 0.1GHZ 4.0GHZ 0.1GHZ
END
DATA
  AVANTBIP:S
  0.1GHZ .74   -40  25.80  156 .017   81  .93  -14
  0.5GHZ .61  -126  12.46  108 .035   44  .57  -31
  1.0GHZ .57  -161   6.80   87 .040   38  .46  -33
  1.5GHZ .57  -180   4.67   75 .052   47  .43  -34
  2.0GHZ .58   166   3.55   64 .060   54  .41  -38
  2.5GHZ .59   160   2.92   59 .068   58  .40  -39
  3.0GHZ .61   150   2.42   50 .080   63  .39  -46
  3.5GHZ .62   142   2.09   41 .091   61  .41  -54
  4.0GHZ .62   134   1.84   32 .106   59  .42  -62
  NOI RN
  0.1GHZ 1.3   .12      5  .17
  0.5GHZ 1.3   .10     25  .17
  1.0GHZ 1.4   .06     50  .16
  2.0GHZ 1.7   .25    172  .16
  4.0GHZ 3.0   .48   -131  .24
END
```

How Much Does it Cost?

The *ARRL Radio Designer 1.5* upgrade will be available September 1. It's free if you bought your copy of *ARRL Radio Designer 1.0* on or after August 1, 1995, and $10 if you bought your copy of *ARRL Radio Designer 1.0* on or before July 31, 1995.

How Do I Get It?

Contact ARRL Publication Sales at ARRL HQ (voice 203-594-0200 and fax 203-594-0303, or e-mail **ltardette@arrl.org**) for details. If you didn't purchase *ARD* directly from ARRL, please be sure to mention your point of purchase.



#	Circuit	Function	Terminations
1	REALBIP	PS11 [deg]	50+j0, 50+j0
2	OPTBIP	PS11 [deg]	50+j0, 50+j0
3	REALBIP	PS21 [deg]	50+j0, 50+j0
4	OPTBIP	PS12 [deg]	50+j0, 50+j0

Circuit: REALBIP
Function: PS21
Default Termination: 50+j0

Figure 3—In addition to fixing the active-TWO noise modeling bug, *ARD 1.5* also introduces three-dimensional controls and dialog boxes.

Free Electronic Goodies

The Table 2 netlist is available (as the file EXRF9509.CKT) from ARRL HQ's Internet info server (send e-mail to **info@arrl.org** with a message text consisting of just the word *HELP*). A ZIP file containing EXRF9509.CKT and its accompanying *ARRL Radio Designer* report file (EXRF9509.RP2) is available from the ARRL HQ BBS (203-594-0306) as EXRF9509.ZIP, and via FTP (as exrf9509.zip) from **ftp://arrl.org/pub/ard** and **ftp://oak.oakland.edu/pub/hamradio/arrl/infoserver/ard**. And *ARRL Radio Designer*'s World Wide Web home page resides at the URL **http://arrl.org/ard/ardpage.html**.

ARRLCAD, an e-mail reflector (mailing list) through which you can now share your questions, answers, ideas and views about *ARRL Radio Designer* and other Amateur-Radio-related circuit and antenna design and simulation tools with other users, is up, running and growing. To subscribe to ARRLCAD, send an e-mail message to

`listproc@tapr.org`

with text that reads

`subscribe arrlcad FirstName LastName`

in which *FirstName* and *LastName* mean exactly that. (I subscribed with the message `subscribe arrlcad david newkirk`, for example.) The reflector software will confirm your subscription with a informational welcome message. Subscribing to and participating in ARRLCAD cost *nothing at all*. QST

---

[2] As readers who run the Table 3 netlist in one of *ARRL Radio Designer*'s superset programs—*Super-Compact*, *Microwave Harmonica*, and *SCOPE*—will discover, only *ARRL Radio Designer 1.0* suffers from this active-TWO noise-modeling bug. *ARD*'s professional superset programs model active-TWO noise just fine!

New Products

PAR ELECTRONICS 2-METER INTERMOD FILTERS

◊ Take a drive through any urban setting and you'll quickly realize that *intermod* is a serious problem for 2-meter operators. Emissions from high-power commercial transmitters (paging systems, in particular) mix with other signals to create interference that can effectively block the 2-meter signals you want to hear.

PAR Electronics, a long-time manufacturer of filters for the CATV and MATV industry, now offers a line of filters designed to tackle this persistent problem. Their VHFDN152 filter is designed to eliminate pager signals from 152 to 153 MHz. Insertion loss is virtually nil and the filter causes no attenuation of 70-cm signals. This makes it ideal for use with 2-meter/70-cm dual-band transceivers. A smaller model (rated at 20 W rather than 50 W) features male and female BNC connectors, allowing it to be used easily with H-Ts. Both models include rugged brass enclosures and are priced at $62 and $68, respectively.

PAR also offers a filter designed for 2-meter packet and satellite users. The VHFDN152Q can eliminate offending signals that appear as close as 700 kHz from the desired passband.

For more information, contact: PAR Electronics, 6869 Bayshore Dr, Lantana, FL 33462, tel 407-586-8278, fax 407-582-1234. QST