

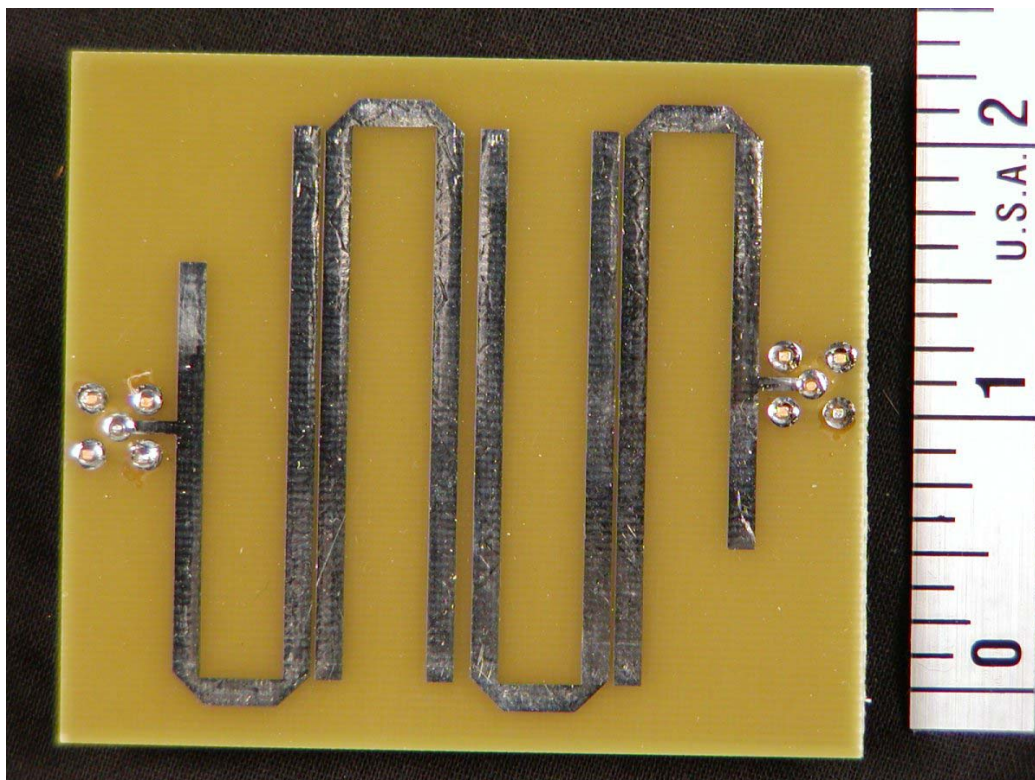
Design of Printed Hairpin Filters

with predictable performance

Paul Wade W1GHZ ©2003

w1ghz@arri.net

Printed filters like the one in Figure 1 are popular for the lower microwave bands — they are used in all the KK7B no-tune transverters¹. The filters consist of a series U-shaped $\frac{1}{2}\lambda$ resonators edge-coupled to their neighbors. Design of these filters has traditionally been somewhat empirical — cut-and-try. However, the filter shown in Figure 1 was designed using a free version (Student Version) of Ansoft² Serenade and Harmonica software; the performance is almost exactly as predicted by software, after correction for the actual dielectric constant of the printed-circuit material. Now that this factor is known, it should be possible to design these filters with predictable performance. The cut-and-try is done in software.



Filter Design

To many hams, a filter is a cavity resonator, like a $\frac{1}{4}\lambda$ “can” or a pipe-cap resonator. For a single resonator like these to have good selectivity, it must have very high Q; this usually requires physically large dimensions. Often, a better choice is a filter with several coupled resonators. Design of the filter is a bit more complicated, but a filter with almost any desired passband characteristic and selectivity skirts may be designed using modern techniques, usually employing a computer to do the math.

Computer programs are freely available to accurately design interdigital filters³ and waveguide filters⁴, but printed hairpin filters have been less predictable. I have used some in weak-signal sources⁵, but they were scaled by eye from the no-tune boards and don’t really have enough rejection of unwanted frequencies.

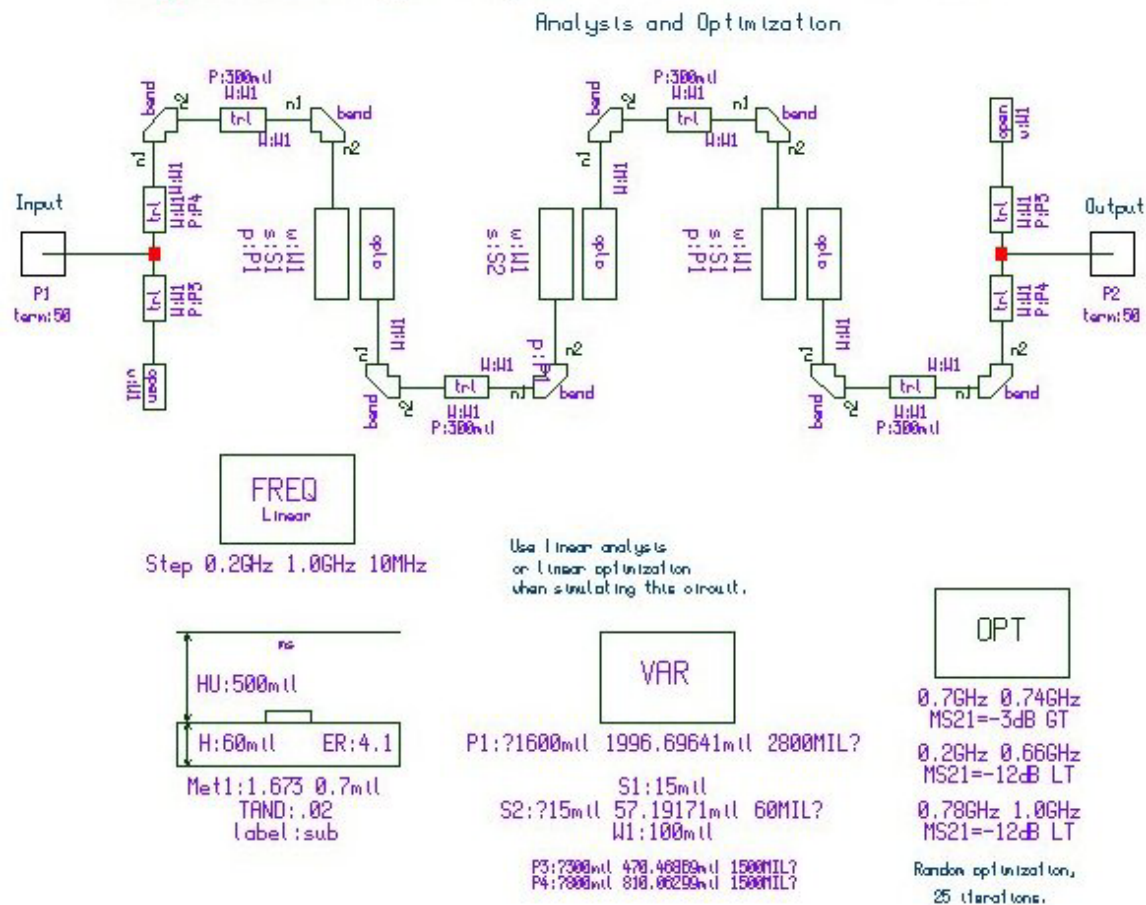
Interdigital filters and waveguide filters use well-defined structures and materials – metal and air – so the math, while difficult, is straightforward. Hairpin filters are microstrip lines, propagating partly in air and partly in a dielectric material, with bends. The calculations must include corrections for all these factors and their interactions; only recently has software included good approximations.

Recently, I needed a filter at about 720 MHz for a project. An interdigital filter would work fine, but is bulky and each copy requires machining. From my previous experiments with interdigital filters⁶, I estimated that a four-resonator filter (more resonators make the skirts sharper but have more loss) with about 5% bandwidth would have the desired characteristics and be relatively easy to make (filters with very narrow or wide bandwidths require extremely critical dimensions). A hairpin filter would be reasonably sized, and easily reproduced as part of a printed-circuit board. The problem was coming up with good dimensions without a large number of wasteful iterations.

There are several filter design software packages available that would probably do this job quite effectively, but I haven’t found one that is both good and free. The free versions are often too crippled to provide any usable results. What I did instead was traditional empirical design, but with the cut-and-try by software optimization using Ansoft Serenade and Harmonica SV; much quicker and less expensive.

A sketch of the filter is shown in Figure 2, a Serenade software schematic; it includes more information than a normal schematic. For accurate results, each detail must be included: a hairpin includes the edge-coupled sections, two bends, and a straight section between the bends. Each piece has a length and width, and the spacing between edges controls the coupling between resonators. Thus, a filter with several resonators has a number of variables to optimize.

Figure 2 Edge-Coupled Filter for 720 MHz



While we could turn the computer loose to optimize all the variables, the time required is an exponential function of the number of variables, so reducing the number is prudent. We can start by noticing that the circuit is symmetrical about the gap in the center; if we maintain this symmetry by making the same changes to both halves, then the number of variables is cut in half. Another reasonable simplification, which also simplifies the PC board layout, is to make all the line widths the same. We can also fix the length of the straight section of the hairpin, between the bends. Then we are

left with six variables to optimize, shown as parameters in the “VAR” section of the schematic:

1. Line width (Parameter W1 on the schematic).
2. Width of center gap. (S2)
3. Width of the other two gaps. (S1)
4. Length of the coupled sections. (P1)
5. Length of the end hairpin between the bend and the input port. (P4)
6. Length of the open-ended section of the end hairpin. (P3)

The optimization target is shown in the “OPT” section of the schematic. I chose a 5% bandwidth with simple specifications: from 0.7 to 0.74 GHz, “MS21= -3dB GT” translates to a gain (**M**agnitude of **S21**) greater than (**GT**) -3 dB. Similar, I specified a gain less than -12 dB from 0.2 to 0.66 GHz and from 0.78 to 1.0 GHz. Just meeting this target wouldn’t make a very good filter, but making the target too strenuous usually just confuses the computer.

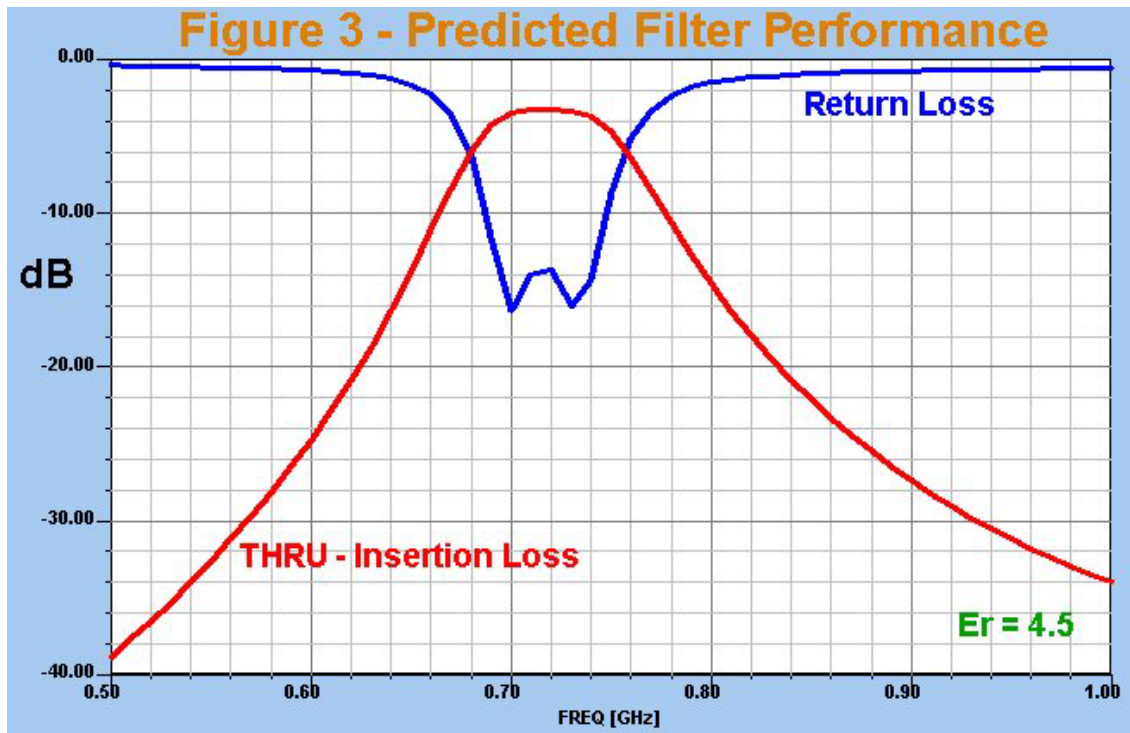
Finally, the cross section dimensions of the board are specified in the lower left, and the “FREQ” section calls for calculation every 10 MHz from 0.2 to 1.0 GHz.

Now we are ready to analyze and optimize, except for one limit of the free “Student Version” of the Harmonica software, which will only optimize a maximum of four parameters at a time. The workaround is to optimize four at a time, with some thought as to the proper combinations — it’s like having four hands for tuning, and six knobs to turn. My choice of order was roughly:

1. The line width and three lengths – this gets the filter close to the desired frequency.
2. The two gap parameters, which set the coupling between the resonators, plus the lengths of the end section above and below the input tap point, which control the return loss (VSWR).
3. Repeat until it looks good or improvement ceases.

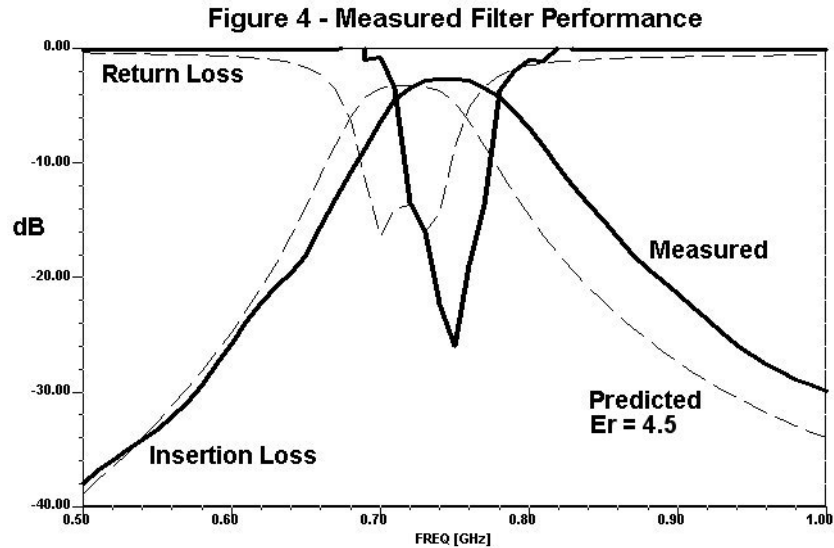
The dimensions in Figure 2 are after optimization; parameters with three values list the specified minimum, current value, and specified maximum. One interesting result is that the optimization has the central coupling gap larger than the outer ones, while cut-and-try designs usually have equal gaps.

The predicted filter response after optimization is shown in Figure 3. The passband is a bit wider than 0.7 to 0.74 GHz, with loss slightly more than 3 dB. This result looked pretty good, and I was really curious about how well it would really work, so I used the ExpressPCB⁷ software to layout the filter on a PC board. The next time I needed a small board and had some space left on a “Miniboard,” I added the filter and ordered the boards.

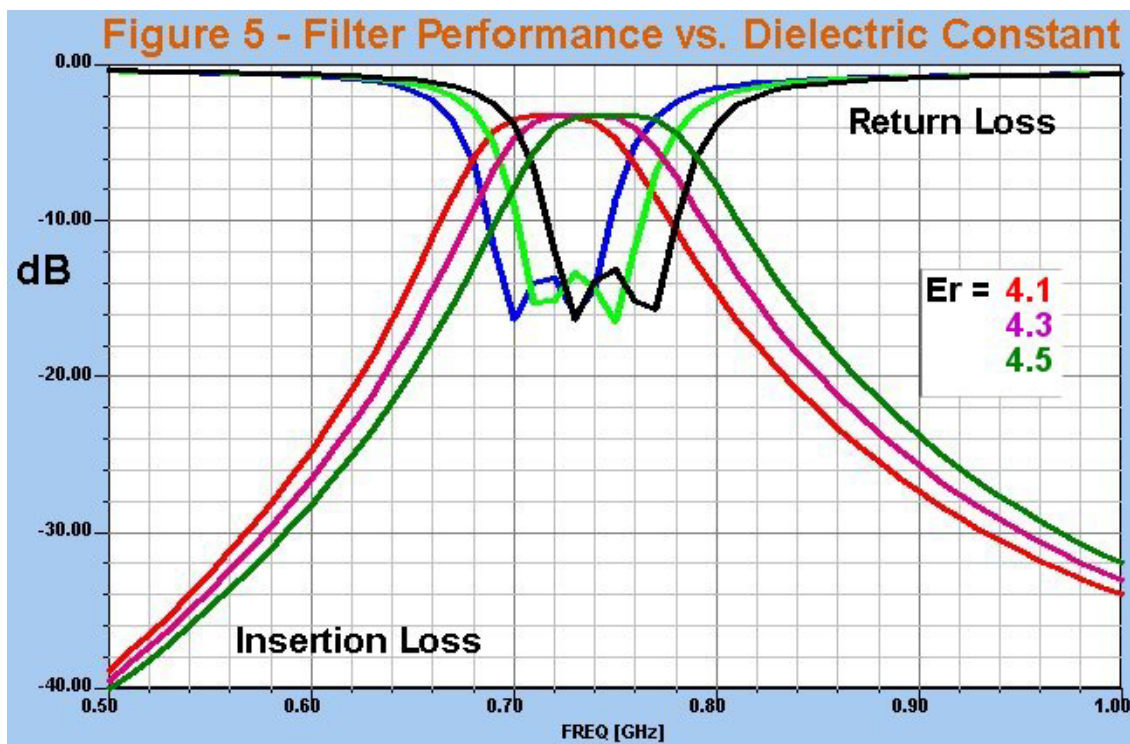


Closing the loop

Four days later, three boards showed up. Not bad for \$63. I cut off the filter with a tinsnips and soldered on the SMA connectors. The measured filter response is shown in Figure 4, with the expected response shown as dashed lines. Close, but not quite: the passband is about 730 to 760 MHz, but the loss is slightly less than expected. Other than the frequency difference, it looks like a good filter.

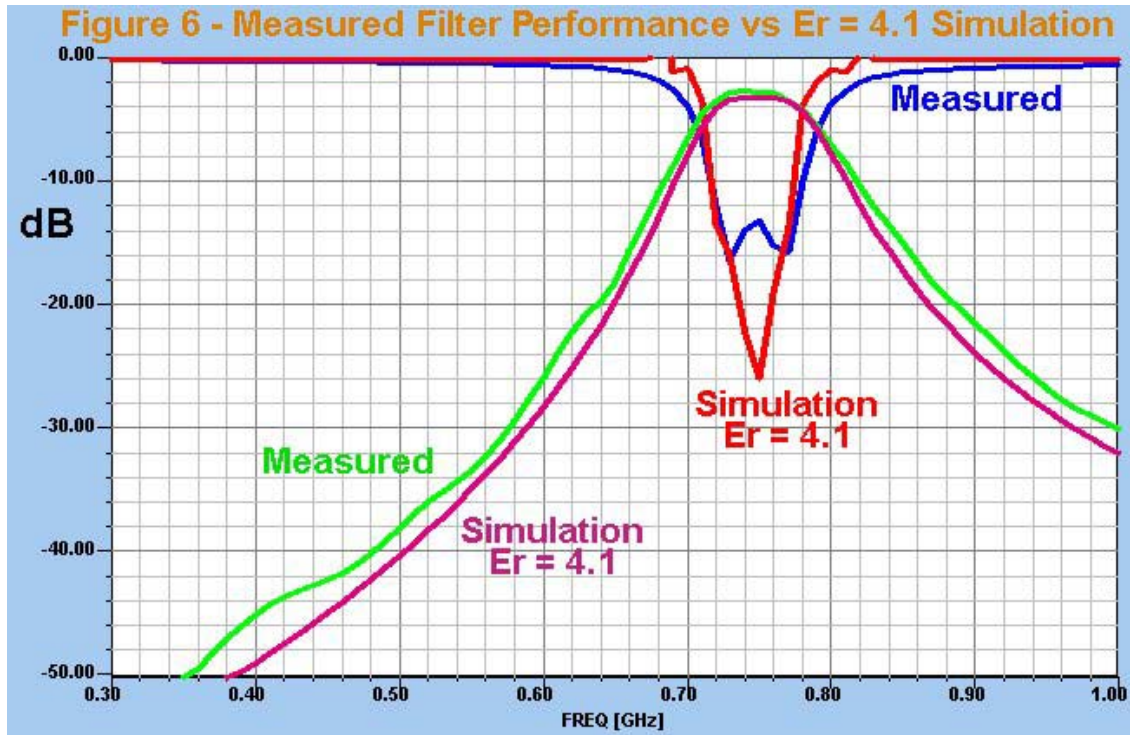


One of the unknowns is the exact dielectric constant of the green PC board material. The material is epoxy-fiberglass, layers of woven fiberglass cloth impregnated with an epoxy resin. The dielectric constant depends on the proportions of the two materials in the mix. The value usually given is 4.5, shown in the schematic as “ER:4.5” along with the loss tangent, shown as “TAND: .02”, which is used to calculate loss.



I suspected the dielectric constant might be the problem, so I ran the filter analysis with different values. Figure 5 shows the filter response with

dielectric constants of 4.1, 4.3, and 4.5. The curve for 4.1 looks a lot closer to the measured response, so I plotted this curve and the measured curve in Figure 6. Looks like a good fit! Now we have a known value for the dielectric constant, $\epsilon_r = 4.1$, for ExpressPCB boards (a manufacturer is usually pretty consistent). Since the measured loss is slightly lower than calculated, we can also infer that the loss tangent is probably slightly less than 0.02, but not enough to recalculate.



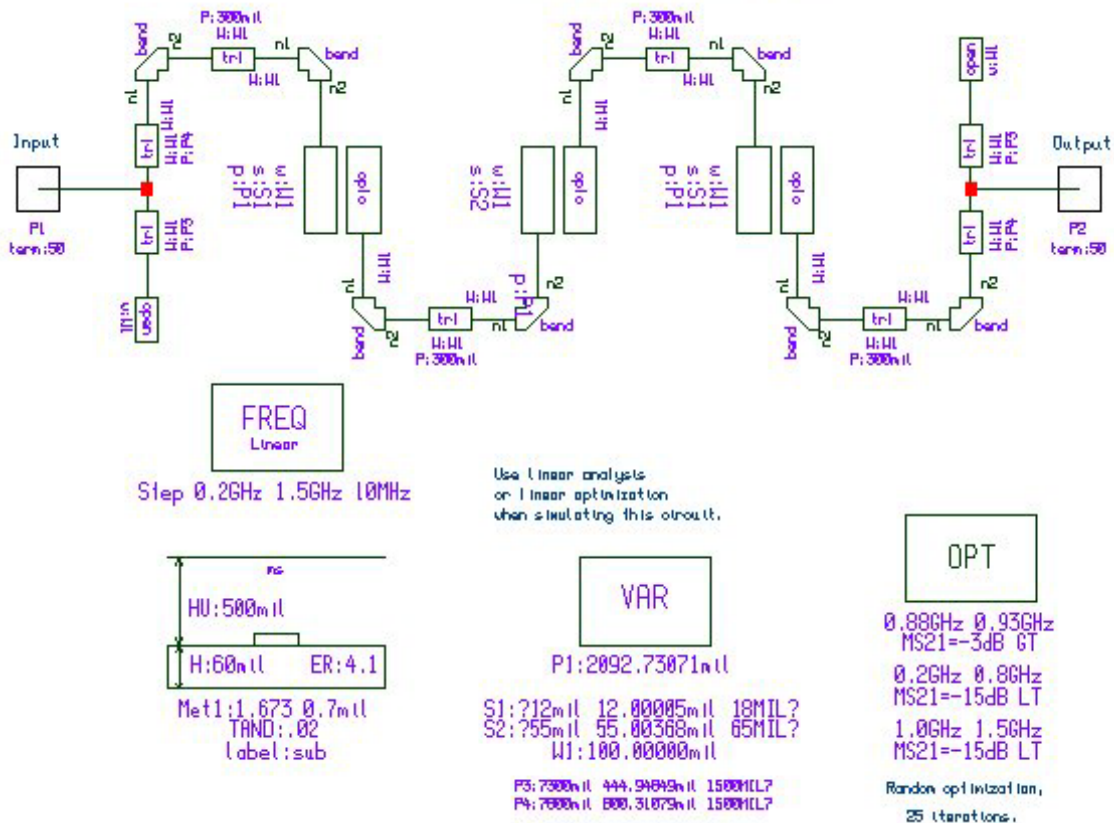
The comparison in Figure 6 suggests that we should be able to design a hairpin filter with predictable results, now that we know the actual dielectric constant. Since the test was done at 720 MHz, we could expect predictability at lower frequencies, and probably somewhat higher also, up to perhaps 1 GHz or so.

Of course, one of the first things to try is to redesign our test filter to the desired frequency. I made the changes in the schematic. Since the actual loss was better than expected, I tightened the out-of-band specification to -15 dB. After optimization, the new dimensions are shown in Figure 7 – the lines are a bit longer, which should lower the center frequency to 720 MHz.

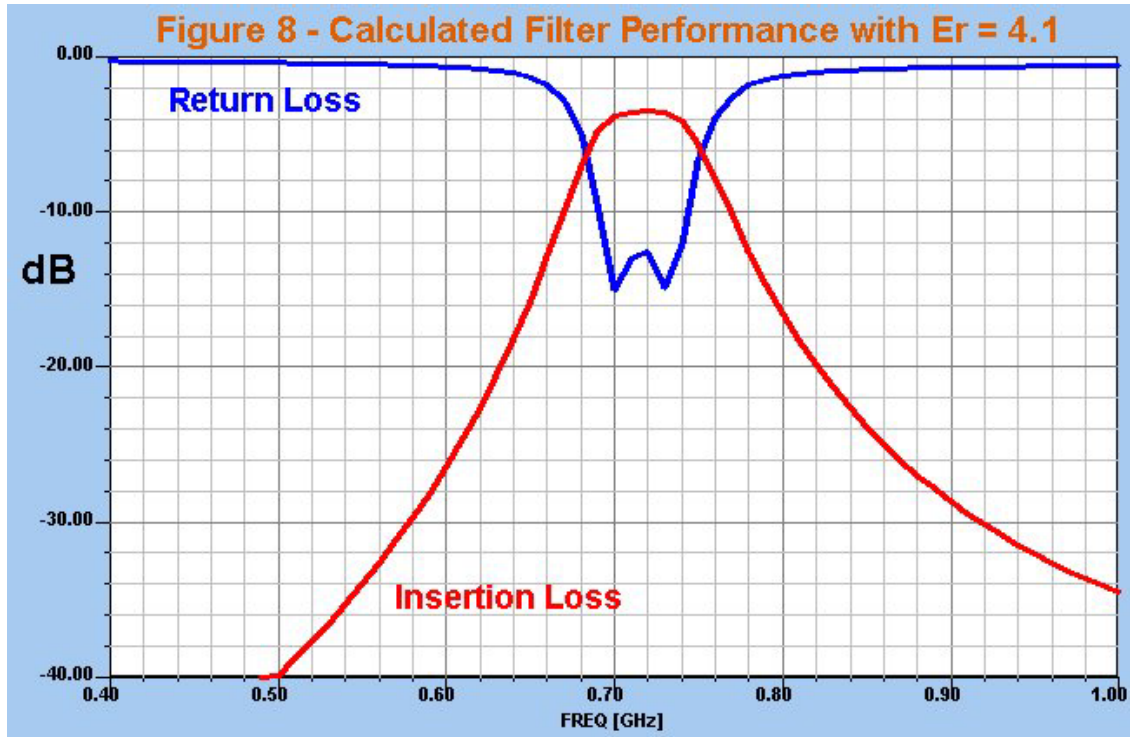
Figure 7 - Redesign Filter for Er = 4.1

Edge-Coupled Hairpin Filter For 720 MHz

Analysis and Optimization



The new predicted performance is plotted in Figure 8. I haven't had new boards made yet, but am confident they will work as predicted. Stay tuned for the rest of this project.



Other frequencies

One easy way to design a hairpin filter for another frequency or bandwidth would be to copy the files from my web page: <http://www.w1ghz.org>. Both the Serenade design files and the ExpressPCB board layout are available.

New software

When I went back to see if there were any updates to the Ansoft Serenade SV software, I found that it has been replaced by Ansoft Designer SV. I've only given it a quick test, but it appears to lack one of the essential features: there is no provision for optimization of variables, so any tuning would have to be done manually, requiring a new analysis for each change. Therefore, I plan to stick with Serenade SV. If you are unable to locate a copy, I may have a suggestion – send me an email.

Summary

It is now possible to design a printed hairpin filter with predictable performance using free software. This capability, combination with other programs for interdigital and waveguide filters, allows the design of good filters over the entire microwave range.

References

1. R. Campbell, KK7B, "A Single-Board, No-Tune 902-MHz Transverter," *QST*, July 1991, p. 25.
2. www.ansoft.com
3. J. Hinshaw, N6JH, & S. Monemzadeh, "Computer-Aided Interdigital Bandpass Filter Design," *ham radio*, January 1985, p. 12.
4. D. G. Sweeney, WA4LPR, "Design and Construction of Waveguide Bandpass Filters," *Proceedings of Microwave Update '89*, ARRL, 1989, p. 124.
5. P. Wade, N1BWT, "Weak Signal Sources for the Microwave Bands," *Proceedings of the 1992 (18th) Eastern VHF/UHF Conference*, ARRL, 1992, p.88.
6. P. Wade, W1GHZ, "Waveguide Interdigital Filters," *QEX*, January 1999, p. 3.
7. www.expresspcb.com