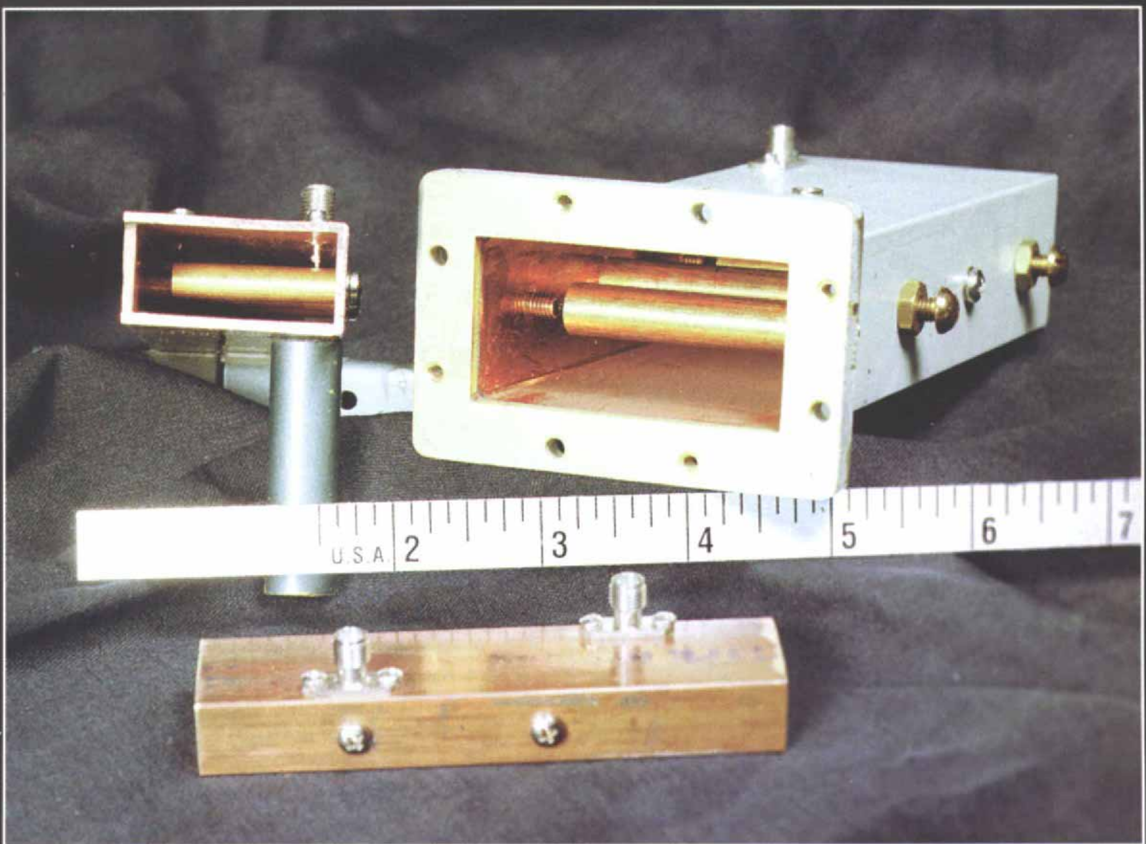
QEX

$5

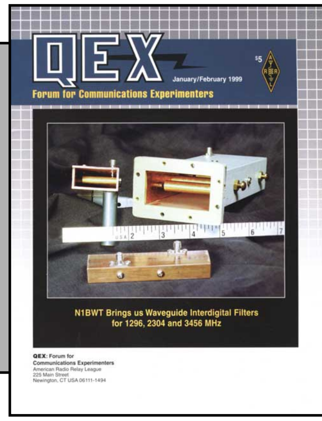
January/February 1999

Forum for Communications Experimenters



N1BWT Brings us Waveguide Interdigital Filters for 1296, 2304 and 3456 MHz

QEX: Forum for
Communications Experimenters
American Radio Relay League
225 Main Street
Newington, CT USA 06111-1494



Paul Wade, W1GHZ, shows how to build interdigital filters for three bands from waveguide and tubing. See page 3.

QEX (ISSN: 0886-8093) is published bimonthly in January 99, March 99, May 99, July 99, September 99, and November 99 by the American Radio Relay League, 225 Main Street, Newington CT 06111-1494. Subscription rate for 6 issues to ARRL members is $22; nonmembers $34. Other rates are listed below. Periodicals postage paid at Hartford CT and at additional mailing offices.

POSTMASTER: Form 3579 requested. Send address changes to: QEX, 225 Main St, Newington CT, 06111-1494

Issue No. 192

Offices

225 Main St, Newington, CT 06111-1494 USA
Telephone: 860-594-0200
Telex: 650215-5052 MCI
Fax: 860-594-0259 (24 hour direct line)
e-mail: **qex@arrl.org**

Subscription rate for 6 issues:

In the US: ARRL Member $22, nonmember $34;

US, Canada and Mexico by First Class Mail: ARRL Member $35, nonmember $47;

Elsewhere by Surface Mail (4-8 week delivery): ARRL Member $27, nonmember $39;

Elsewhere by Airmail: ARRL Member $55, nonmember $67.

Members are asked to include their membership control number or a label from their *QST* wrapper when applying.

In order to insure prompt delivery, we ask that you periodically check the address information on your mailing label. If you find any inaccuracies, please contact the Circulation Department immediately. Thank you for your assistance.

Features

Columns

Jan/Feb 1999 QEX Advertising Index

Telephone: 860-594-0200
Telex: 650215-5052 MCI
MCIMAIL (electronic mail system) ID: 215-5052
FAX: 860-594-0259 (24-hour direct line)

Empirically Speaking

We amateurs have often had to be concerned about our image with our elected officials in Congress, our friends in the United Nations and the public. Concern has peaked around those times when commercial or governmental interests have made a grab for some of our spectrum. We've had to continually refresh society's memory about our history of service, especially our service during natural disasters and other emergencies. Perhaps our best publicity has come in times of need.

Satellite and cellular services, however, are rapidly encroaching on this area, one of our primary *raisons d'etre*. In addition, digital-audio broadcasting is right around the corner. International broadcasters may soon need more of our HF bandwidth. It might not be long before more folks are asking: "What have you done for us lately?" We must increase the emphasis on other unique aspects of our avocation. Of course, these will still include public service, but our ability to advance our art, to make the best use of our spectrum and to bring along new interest in purely technical pursuits will be pivotal. Among other things, those are what fill the pages of *QEX*.

I've heard from hams—some who are ardent equipment builders or software writers—that they want the best forum for their construction projects or utility programs. Others are saying they want more projects to build, but don't want to bathe in higher mathematics to do so. Still others are chiefly interested in keeping informed about new theory and technology. I think that many of us are intermediate to those views, but *QEX* will continue to have room for them all. What is your perception?

In This *QEX*

Okay builders, put the coffee on and warm up the iron—it's construction time! Paul Wade, N1BWT, has found one of the magical combinations of readily available parts that, together with a bit of drilling and soldering, produce useful results for you microwave fans. Michael McKay, W4AZR, examines things that happen around those "no-tune" transverters, such as birdies and spurious responses, which may inspire you to get out that drill and iron.

Patrick Wintheiser, W0OPW, gives us a "Compact Mobile Tuner," an excellent project for getting more of the band from your antenna, whether you're mobile or not. Winding coils may be good therapy for some of us, but Bob Dildine, 7J1AFR/W6SFH, is back with a way to reduce fatigue and increase repeatability.

C. A. Hoover, K0VXM, shows how to put together his "Cheap Sweep" with parts gleaned from a junk box. This neat construction project helps to tune UHF filters. For the more ambitious builder, Richard Hanson, K5AND, helps you assemble a pair of 3CX800s to get ready for more 6-meter openings. Heck, with this unit, make your own openings.

Curtis Preuss, WB2V, conquers the frequency-versus-temperature problem with his DDS compensation method. I have little doubt this technique will find its way into many designs, if it hasn't already. From across the pond comes an article of interest especially to receiver designers. Designs with poor low-order IMD performance normally pack up in a hurry there. Jos van der List, PA0JOZ, analyzes phase-noise effects—from both receivers and transmitters—and provides criteria and a fixture for measuring them accurately. This article also appears in two issues of the Dutch magazine *Electron*.

Robert Dick shares his research on how to "Tune SSB Automatically." Yes, Virginia, it can be done—try it! Larry Dobranski, VA3LGD, reasons that it needn't be so difficult to add support for new computer-controlled equipment every time something different hits the market. I suspect that's right, and this is a subject worthy of some discussion. Stu Bonney, K5PB, looks at how well an antenna will survive that gale rolling in from the Gulf in the first part of his two. Zack Lau, W1VT, offers Microwave PA tips in his RF column.—*73, Doug Smith, KF6DX,* **kf6dx@ arrl.org.**

Waveguide Interdigital Filters

Build sturdy, predictable microwave filters from waveguide. Here are designs for three bands: 1296, 2304 and 3456 MHz.

By Paul Wade, W1GHZ (N1BWT)

Most microwave transverters, especially the "no-tune" variety, need some additional filtering to operate in locations with "RF pollution"—accessible mountaintops are notoriously bad environments. Waveguide post filters provide superior performance at 10 GHz[1] and 5760 MHz,[2] but become large and heavy at lower frequencies. Interdigital filters are excellent performers at the lower microwave frequencies,[3] but the usual construction techniques require some machining, mostly tedious tapping of threads in many holes. One of the beauties of waveguide filters is that the basic structure is accurately defined by the waveguide, so construction requires only drilling and soldering. Since surplus waveguide is reasonably plentiful, I wondered if it could be used to build interdigital filters for the lower microwave bands. As we shall see, my experiments were quite successful.

[1]Notes appear on page 8.

161 Center Rd
Shirley, MA 01414
wade@tiac.net

Interdigital Filters

The basic structure of an interdigital filter, shown in Fig 1, is a group of coupled resonators in a metal housing. Each resonator is an electrical $\lambda/4$ long, but physically shortened by capacitance at the open end. The resonators are *interdigitated*, with the position of the open ends of the resonators alternating as depicted in Fig 1. (A similar filter with all resonators aligned in the same direction is called a comb-line filter.) The coupling between resonators is controlled by their separation. Several methods are commonly used to make input and output connections, but a simple one is to use taps on the input and output resonators.

The starting dimension for an interdigital filter is the width of the housing, which should be $\lambda/4$ at the operating frequency. All of the other dimensions are interrelated—a change in one affects others—so that empirical design of a filter would be difficult and frustrating. Fortunately, computer programs are available to design interdigital filters. A BASIC program,[4] by N6JH, appears in *ham radio* magazine. I translated this into PASCAL and compiled it. My version, *INTFIL.EXE*, is available for downloading at **http://www.qsl.net/~n1bwt/intfil.zip**. One 1296 MHz filter that I built using this program was carefully measured using an automatic network analyzer and found to match the predicted performance almost perfectly with no tuning. This gave me confidence in the accuracy of the program.

Filter Design

The first part of filter design is the same for all types of filters—calculation of coupling coefficients and other parameters to achieve the desired performance. These are tabulated in *The ARRL Handbook*[5] and other reference books[6] for the most common types of filters: the Butterworth (maximally-flat) and the Chebyshev response, which trades some passband ripple (amplitude variation) for somewhat steeper skirts at the passband edges. The tabulated parameters, g_{mn}, are for a normalized prototype filter, so that further calculations are required to find actual component values for a desired frequency and impedance.

The second part of filter design is to convert the normalized parameters into component values or physical dimensions. The calculations are quite tedious, so graphical solutions were often published[7] before computers were commonly available. Now these calculations are easily performed on a PC, allowing us to evaluate multiple filter designs before choosing one to build.

Design of an interdigital filter begins with the choice of a required bandwidth. Simple filter programs such as ***INTFIL*** are only reliable for bandwidths between about 1% and 10% of the center frequency, and very-narrow-bandwidth filters are lossy and require tight tolerances in construction. Therefore, a 3% to 5% bandwidth is recommended as a good starting point. The next step is to decide how steeply the skirts roll off at the passband edges. For example, steeper skirts are required to reject an image

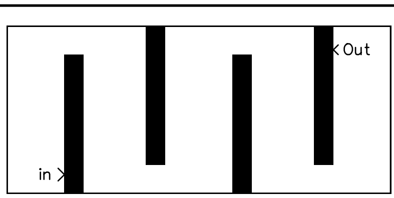


Fig 1—Interdigital filter cross-section sketch.

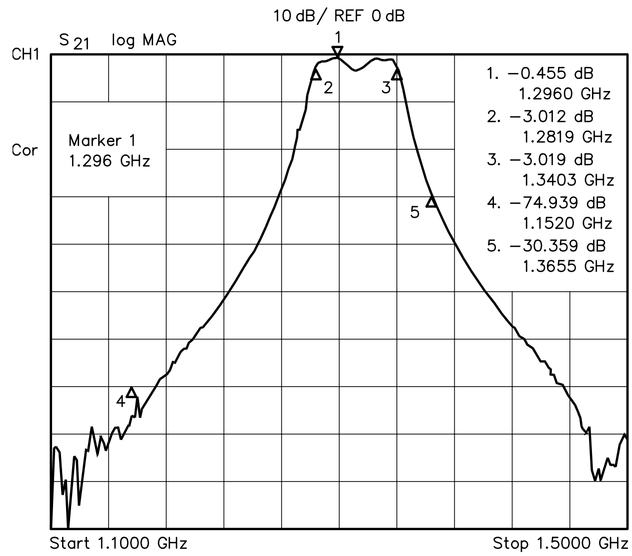


Fig 2—Performance of 1296-MHz filter—WR-229 waveguide.

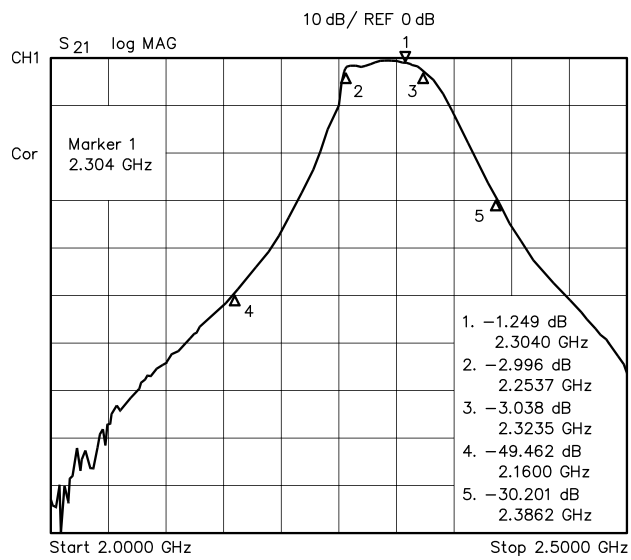


Fig 3—Performance of 2304-MHz filter—WR-137 waveguide.

Table 1: Waveguide Dimensions for Interdigital Filters

Waveguide	*Wide Dimension*	*Frequency (λ/4, MHz)*
WR-340	3.4″	868
WR-284	2.84″	1039
WR-229	2.29″	1289
WR-187	1.872″	1577
WR-159	1.59″	1857
WR-137	1.372″	2152
WR-112	1.12″	2636
WR-90	0.90″	3280
WR-75	0.75″	3937
WR-62	0.622″	4747
WR-50	0.51″	5789

close to the operating frequency. Generally, filters with steeper skirts require more resonators and have more loss, so a compromise may be in order. It is possible to calculate the number of resonators required, but a few trial designs on the computer should yield the same result and provide some insight as well.

Waveguide Interdigital Filters

Now let's design a filter in an available waveguide. As mentioned above, we should start with the width of the enclosure at λ/4 at the center frequency. This would be the large inside dimension of a waveguide used as the enclosure. Table 1 lists the inside dimensions for some commonly available waveguides and the frequencies for which the wide dimensions are λ/4. The large dimension for WR-229 is λ/4 at 1289 MHz, so it is a logical material for a 1296 MHz filter. Simply designing a filter for the 1289 MHz center frequency with enough bandwidth to include 1296 MHz does the trick. I chose a 50 MHz bandwidth and used ***INTFIL*** to calculate the rest of the dimensions for a resonator diameter of 3/8 inch. Since WR-229 is being scrapped as 4 GHz telephone microwave links are decommissioned, I was able to find all I could carry at a flea market for $5.

Once the filter was assembled, I found that the bandpass was slightly above 1296 MHz, a little higher than the design. This was easy to fix, however: I drilled and tapped holes for tuning screws opposite the open ends of the resonators and inserted screws to add capacitance and lower the frequency. On the other hand if the frequency ended up a bit low, then it would be necessary to shorten the resonators slightly.

I first adjusted the screws for minimum insertion loss at 1296 MHz, then readjusted them for minimum SWR at both ends. The second adjustment is a bit more involved since each adjustment affects both ends and a few reversals were needed. The final passband, shown in Fig 2, is about 58 MHz wide with an insertion loss less than 0.5 dB at 1296 MHz.

Tuning is straightforward for these filters, with moderate bandwidth and a reasonable number of resonators. However, filters with very narrow or wide bandwidths, or with many resonators, require a more complex tuning procedure. Dishal's procedure[8, 9] allows the tuning of one resonator at a time.

For other bands, no waveguide exactly matches λ/4, but there are some good candidates that fall within about 10% of the desired frequency. The ubiquitous X-band waveguide, WR-90, is close to λ/4 at 3456 MHz, while WR-137 (used in 6-GHz microwave links) is close to 2304 MHz. For these two, simply making a wide filter is not good enough. The bandpass would include the commonly used LO frequencies for a 144-MHz IF. Thus, we need a design procedure that can move the center frequency slightly.

The design procedure that I use makes two similar designs, one at the desired frequency and one at the λ/4 frequency waveguide, using the same percentage bandwidth (bandwidth÷ center frequency) for both designs. Using the same percentage bandwidth results in two designs differing only in the resonator lengths and tap positions, and the difference is small because the frequencies are close together. Since the higher-frequency design also calls for the λ/4 distance to be shorter, making the resonators this short would result in less capacitance and an actual resonant frequency higher than desired. My compromise is to split the difference between the two design lengths and make the resonator lengths halfway between the two designs.

I followed the above design procedure for two more filters, one for 3456 MHz in WR-90 waveguide with 108 MHz bandwidth and the other for 2304 MHz in WR-137 waveguide with 75 MHz bandwidth. Each design uses four resonators with a Butterworth response. After fabrication, both filters had passbands that included the design frequency, as shown in Figs 3 and 4. Thus, they are usable with no further tuning. Any elective tuning would optimize the input and output SWR at the desired frequency.

To simplify testing, the effects of the end walls are minimized by locating them relatively far from the end resonators. The result is that leaving the end walls off during testing makes

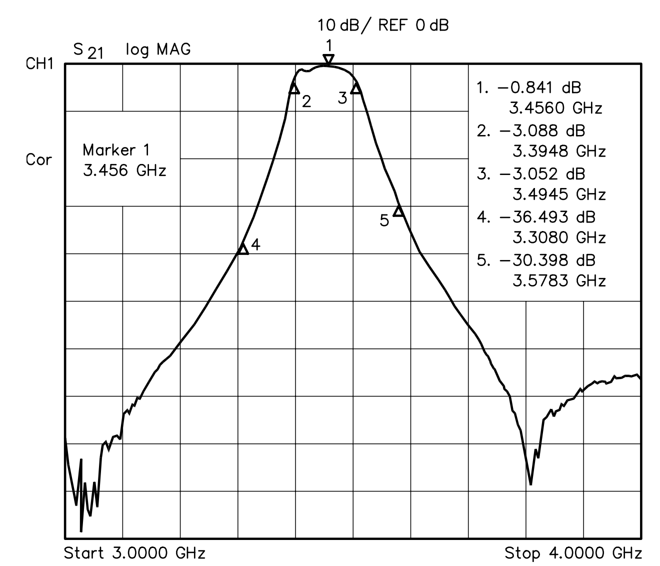


Fig 4—Performance of 3456-MHz filter—WR-90 waveguide.

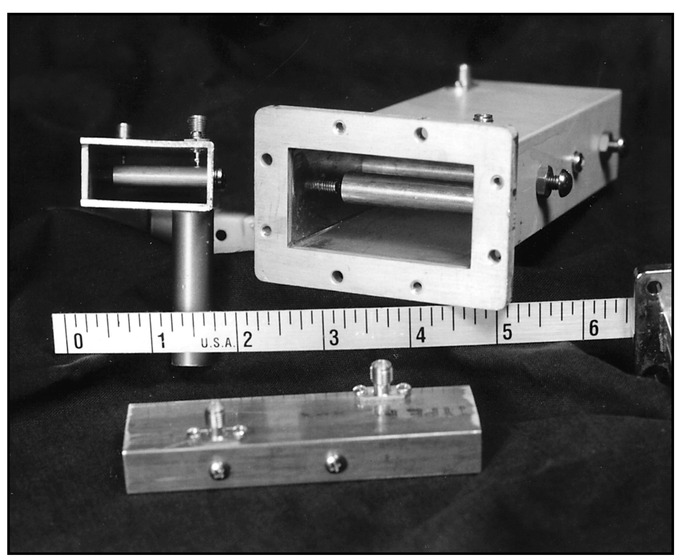


Fig 5—Waveguide interdigital filters for 1296, 2304 and 3456 MHz.

little difference in performance. I located the end walls one inch from the end resonators in the 1296 MHz filter, and could find only a slight performance difference with the end walls in place. There was no detectable difference for the higher frequency filters. Of course, the end walls should be installed for operation; stray leakage could otherwise negate the effect of the filter.

Construction

The three completed filters are shown in Fig 5. Each resonator is attached by a screw through a narrow wall of the waveguide and the coaxial connectors are mounted in a wide wall of the waveguide with short leads to the tap points on the end resonators.

Resonator lengths and spacings are fairly critical, so accurate measurement is needed. The holes are best made with a drill press (see the sidebar "Tools for Interdigital Filter Construction." Start with a center drill or small drill bit to spot the hole, then follow with a drill bit of the desired diameter. The mounting holes in the end of the resonators should be tapped and countersunk slightly, so contact is made around the resonator perimeter. For initial testing, I don't solder the input and output connections, but rather make them slightly long with a sharp point contacting the tap point on the resonators.

Using waveguide as the housing makes the filters easy to build, and results in a robust, stable filter, suitable for rover operations. Some of my previous experiments in filter construction using hobby brass and PC board were less successful due to mechanical instability: Vibrations or the weight of connecting coax cables affected their performance. One notably bad filter was so unstable that the frequency response would vary during measurement.

After building and testing the filters in Fig 5, I wondered if there was an even simpler way to make these filters. Since the resonator length for the 3456 MHz filter is just a hair's breadth over $^3/_4$ inch, perhaps an ordinary $^3/_4$-inch-long, $^1/_4$-inch-diameter threaded standoff could be used as a resonator. The resonator spacings and the tap-point dimensions are the critical ones, so I calculated a filter with 75 MHz bandwidth using $^1/_4$-inch-diameter resonators, then built it with threaded standoffs. It took less than an hour to complete. A quick measurement showed nearly 3 dB loss at 3456 MHz,

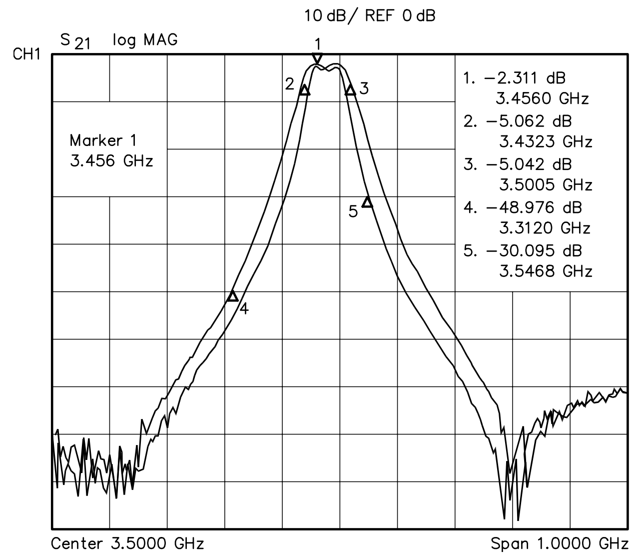


Fig 6—Performance of 3456 MHz filter built with threaded standoffs in WR-90 waveguide (lower frequency response is after tuning).

Table 2: Waveguide Interdigital Filter Examples

Waveguide	*WR-229*	*WR-137*	*WR-90*	*WR-90*	
Target Frequency	*1289*	*2304*	*3456*	*3456*	*MHz*
Bandwidth	*50*	*75*	*108*	*79*	*MHz*
Resonator (Designed for waveguide λ/4)					
Diameter	0.375	0.25	0.1875	0.25	inches
End Length	1.983	1.099	0.732	0.727	inches
Interior Length	1.971	1.095	0.73	0.728	inches
Tap Point	0.23	0.127	0.089	0.089	inches
λ/4 Frequency	1289	2155	3280	3280	MHz
Bandwidth	same	70	100	75	MHz
Resonator (Designed for target frequency)					
Diameter	″	0.25	0.1875	0.25	inches
End Length	″	1.187	0.777	0.773	inches
Interior Length	″	1.183	0.775	0.772	inches
Tap Point	″	0.136	0.093	0.095	inches
Spacing 1-2,3-4	1.47	0.864	0.58	0.653	inches
Spacing 2-3	1.632	0.951	0.636	0.709	inches
Compromise Dimensions					
Resonator					
Diameter	same	0.25	0.1875	0.25*	inches
End Length	″	1.144	0.756	0.75*	inches
Interior Length	″	1.14	0.752	0.75*	inches
Tap Point	″	0.132	0.09	0.092	inches
Loss, Calculated	0.2	0.3	0.4	0.6	dB
Loss, Measured	0.45	1.25	0.8	2.3	dB
Bandwidth, Measured	58	70	100	68	MHz
LO Frequency	1152	2160	3312	3312	MHz
LO Rejection, Calculated	–59	–47	–34	–45	dB
LO Rejection, Measured	–75	–49	–36	–49	dB

* = threaded standoff

however, so I took it apart to add tuning screws to see if I could improve it. Careful tuning only reduced the loss to 2.4 dB, versus 0.8 dB for the filter with machined brass resonators. Fig 6 shows the response before and after tuning. The response is slightly higher in frequency before tuning, but otherwise there is little difference. I attribute the higher loss to three factors:

- I arbitrarily designed this version for a 75 MHz (2%) bandwidth, compared to a 108 MHz (3%) bandwidth for the other 3456 MHz filter. As previously mentioned, filters with narrow bandwidths tend to be more lossy.
- The threaded standoffs are plated with nickel, a lossy metal.
- The standoffs are chamfered at the ends, so the contact area is smaller at the shorted end where currents are highest.

Performance

The waveguide interdigital filters exhibit excellent performance (as shown in Figs 2, 3, 4, and 6) with low insertion loss in the passband and high rejection of undesired frequencies. Steep skirts provide good rejection of possible spurious signals, such as LO leakage only 144 MHz away from the operating frequency. LO rejection is much greater for the 1296 MHz filter (–75 dB) than for the others (–49 dB at 2304 MHz and –36 dB at 3456 MHz) because the relative LO separation is much greater at 1296 MHz. It's 9% of the operating frequency at 1296 MHz, versus 6% at 2304 MHz and 4% at 3456 MHz.

Table 2 lists the filter dimensions and compares the measured performance with the design values, as calculated by ***INTFIL***. The measured performance is quite close to the design values. The dimensions shown are for the two designs for each filter, plus the compromise values that I fabricated, to illustrate the design procedure.

The only performance flaw for these filters is poor harmonic rejection. At frequencies much higher than the operating frequency, the waveguide enclosure behaves as a waveguide rather than just an enclosure. This behavior is not unique to the waveguide interdigital filters—all conductive enclosures will propagate waveguide modes at frequencies above the cutoff frequencies for the interior dimensions. Fig 7 shows the transmission characteristics of the 3456 MHz filter from 50 MHz to 20 GHz. Out-of-band rejection is excellent (> 70 dB) below

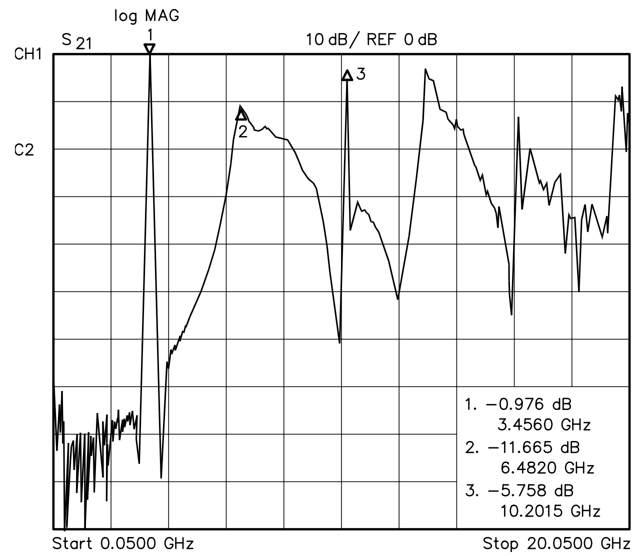


Fig 7—Wide-range performance (0 to 20 GHz) of 3456-MHz filter—WR-90 waveguide.

Tools for Interdigital Filter Construction

I use a small metal lathe to trim the interdigital-filter resonators to length. A lathe is the ideal tool for this work, but is a luxury for most hams. Some other tools are great for homebrewing, however, and inexpensive imports have made them quite affordable. The two that I find almost indispensable are a drill press and a dial caliper.

A drill press is a sturdy drill on a stand, with an adjustable table to hold the work. For the interdigital filters, it drills the holes square and true. The mounting holes in the resonator rods can be first drilled and then threaded, by chucking a screw tap in the drill press and feeding it *by hand* (power off!), with the rod held in a vise. Imported tabletop drill presses are available for less than $60.[A, B] I used and abused one of these constantly for 16 years before splurging on a larger floor-mounted model; the old one is still in use for local "Elmer" sessions.

A dial caliper is a measuring instrument capable of resolving dimensions to a precision of 0.001 inches on a dial. Most of the dimensions in the interdigital filters, and for much microwave work, must be more precise than I can measure with a ruler, so my dial caliper also sees constant use. I even scribe dimensions directly with the caliper tips—a gross abuse of the tool that I justify by its low replacement cost. Imported 6-inch dial calipers are available for less than $20,[C, D] a very modest investment compared to the alternative: eyestrain and frustration.

Everything else can be done with common hand tools—plus patience. A hacksaw and file can cut the waveguide to length and trim the resonators, measuring frequently with the dial calipers. The holes are carefully marked, center-punched, and drilled to size with a drill press. A set of "number-sized" drills provides many more choices of hole size than ordinary fractional sizes, and is available at reasonable cost from any of the suppliers already mentioned.—*W1GHZ*

A modest investment in tools—inexpensive, but not cheap—can add to the pleasures of homebrewing and improve the results.

Sources

[A]Harbor Freight Tools, 3491 Mission Oaks Blvd, Camarillo, CA 93011; tel 800-423-2567; **http://www.harborfreight.com**

[B]Grizzly Industrial, Inc, 1821 Valencia St, Bellingham, WA 98226; tel 800-523-4777; **http://www.grizzlyimports.com**

[C]MSC Industrial Supply Co, 151 Sunnyside Blvd, Plainview, NY 11803; tel 800-645-7270; **http://www.mscdirect.com**. The street address is for the corporate offices. Call for a location near you.

[D]Eastern Tool & Supply, 149 Grand St, New York, NY 10013; tel 800-221-2679.

the passband and good above the passband up to about 6 GHz. Above 6 GHz, various waveguide modes are propagated and limit the attenuation.

Conclusion

Good filters are important for operation in locations with RF pollution, and are recommended when no-tune transverters are followed by broadband power amplifiers. Interdigital filters offer excellent performance for the lower microwave bands. The filters are easily constructed in a waveguide housing without extensive machining and require little tuning, resulting in a robust, high-performance filter.

Notes

[1]G. Elmore, N6GN, "A Simple and Effective Filter for the 10-GHz Band," *QEX*, July 1987, pp 3-5.

[2]P. Wade, N1BWT, "A Dual Mixer for 5760 MHz with Filter and Amplifier," *QEX,* August 1995, pp 9-13.

[3]R. Fisher, W2CQH, "Interdigital Bandpass Filters for Amateur VHF/UHF Applications," *QST*, March 1968, p 32.

[4]J. Hinshaw, N6JH, and S. Monemzadeh, "Computer-Aided Interdigital Bandpass Filter Design," *ham radio*, January 1985, pp 12-26.

[5]R. Dean Straw, N6BV, Editor, *The ARRL Handbook for Radio Amateurs,* ARRL, 1997. See pp 30.22 and following.

[6]G. Matthei, L. Young, and E. M. T. Jones, *Microwave Filters, Impedance Matching Networks, and Coupling Structures*, McGraw-Hill, 1968.

[7]W.S. Metcalf, "Graphs Speed Design of Interdigital Filters," *Microwaves*, February 1967, pp 91-95.

[8]M. Dishal, "Alignment and Adjustment of Synchronously Tuned Multiple-Resonator-Circuit Filters," *Proc. IRE*, November 1951, pp 1448-1455.

[9]G. Matthei, L. Young, and E. M. T. Jones, *Microwave Filters, Impedance Matching Networks, and Coupling Structures*, McGraw-Hill, 1968, pp 668-673.

□□

Tune SSB Automatically

Need a new challenge? How about using DSP to accurately (within 2-3 Hz) tune SSB transmissions in about five seconds? Here's an outline of the research and theory to do it.

By Robert Dick

Editor's Note: This article covers a lot of ground quickly and assumes a fairly sophisticated knowledge of signal processing. The important message is that you can tune SSB automatically, and if you really want to get into it, obtain a copy of Reference 1. Those of you who try this with a ham receiver, let us know how it worked and the hardware/software you used. We can then publish a follow-up article.

It is possible for your personal computer to tune SSB voice signals using a high-level language and standard digital-signal-processing (DSP) techniques. First you must tune to within about plus or minus one kilohertz, then your computer can finish the job with an accuracy of about plus or minus three hertz. The tuning method is based on the property of human speech that most of the time it is periodic—and this period varies with time. This article describes the entire DSP process and gives code in C that does the signal processing. To do the whole job you will need, in addition, a real-time A/D and either a D/A or a way to let your computer tune your receiver.

I discovered the method while doing US Government sponsored research trying to remove voice-on-voice interference. This was one of those cases where a byproduct of research was more successful than trying to accomplish the main aim. Reference 1 reports on my work, and contains code in FORTRAN that goes most of the way to estimating SSB mistuning. It also contains code for SSB retuning via DSP that I have improved on for the present article.

Human speech is made up of two types of sounds: voiced and unvoiced, depending on whether or not the speaker's vocal cords are engaged while producing a certain sound. Unvoiced speech is generally much lower in volume than voiced speech and, in the 3 kHz bandwidth of Amateur Radio, much of the unvoiced sound is filtered out. Voiced speech is nearly periodic over the short term (tens to hundreds of milliseconds), with a period that varies over time.

A periodic sound, when viewed in the frequency domain, shows a series

13 Speer St
Somerville, NJ 08876
rdick@idt.net

of peaks (harmonics). The frequency difference between neighboring peaks is the reciprocal of the time period of repetition. This range of peaks has one significant property for our purposes. If the range of peaks is extrapolated downward in frequency there will always be a resultant peak at zero frequency (dc). When the pitch changes, all the peaks shift position, except the (resultant) one at zero hertz. When SSB speech is mistuned, there will be one and only one (resultant) peak at some non-zero frequency, which does not shift with time. This represents the mistuning.

Fig 1 shows a "waterfall" plot giving an evolution of the (properly tuned) speech square-root-of-power spectra versus time. Each line shows a spectrum of zero to 3200 Hz. A series of peaks is evident in most lines. There are no peaks at zero hertz on the left, of course, but if each "comb" of peaks were continued to the left, it would have a tooth at zero hertz.

Note that each spectrum line in the waterfall plot contains an envelope that resembles a sinusoid. This essential fact enables us to use the fast Fourier transform (FFT) to estimate the period and phase of the series of frequency-domain peaks. This estimator FFT goes from the frequency domain back to the time domain. I call this process—going from time to frequency and back to time—complex correlation. The key to complex correlation is what is done in the frequency domain between the two FFTs: The square root of the power spectrum is taken, and the negative-frequency portion is zeroed. Then the inverse FFT is taken. Because the inverse FFT's input was not symmetric, its output is complex. This is an essential part of the signal processing.

The complex correlation has a peak in its magnitude at the time interval corresponding to the period of repetition of the speech waveform put into it. The phase at this magnitude-peak represents something of the mistuning of the speech. If the speech is exactly properly tuned the phase will be zero at the peak and the complex correlation will be purely real and positive.

If zero frequency is exactly halfway between (resultant frequency-domain) peaks, the complex correlation will be purely real and negative at its magnitude-peak. Other conditions will result in other complex phases at the peak.

Fig 2 shows another waterfall plot, this time of the magnitude of the complex correlation. All the horizontal

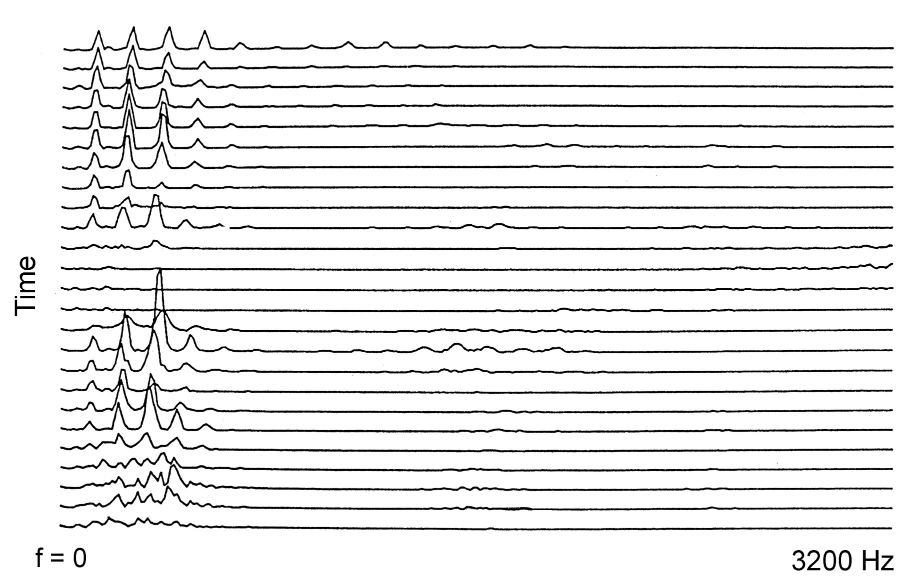


Fig 1—Square roots of speech-power spectra.

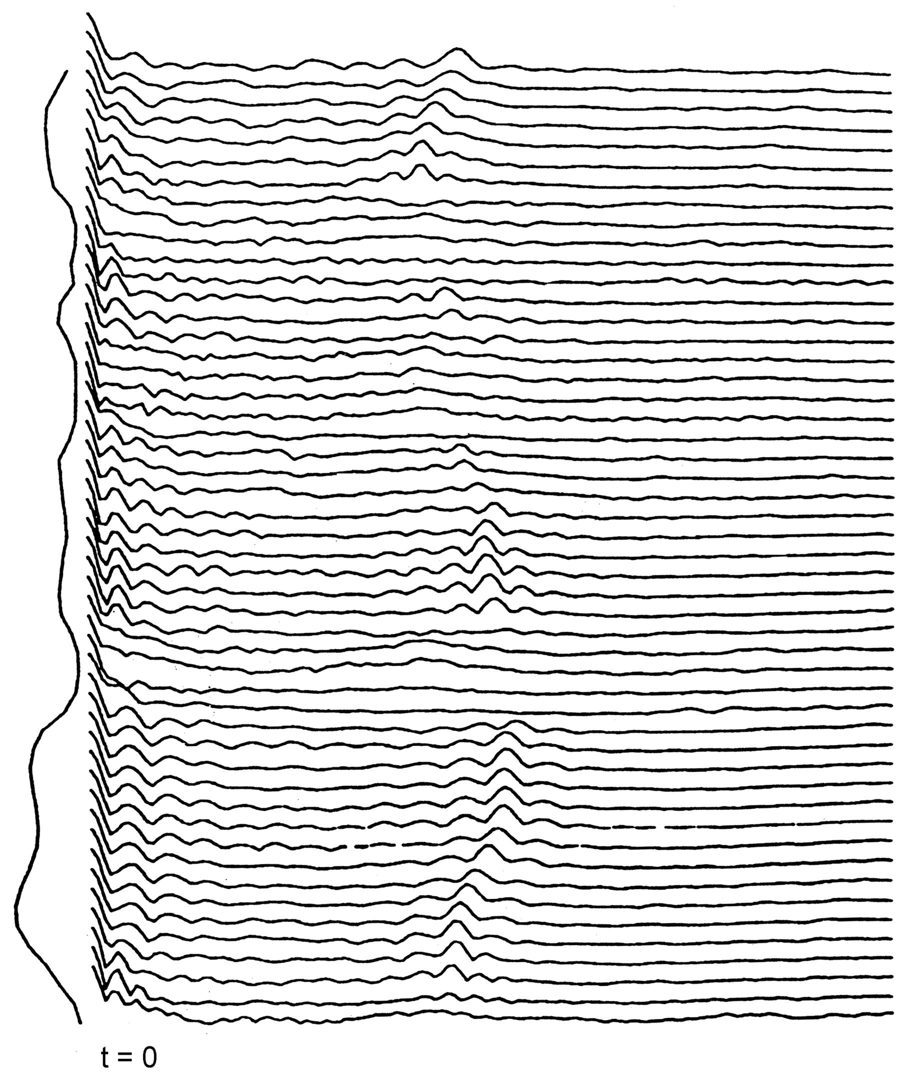


Fig 2—Magnitudes of speech "complex correlations" (see text for an explanation of this term).

plots are normalized to have equal power. On the left of the waterfall is a vertical wavy line showing, by its shape, the power in the waterfall. The leftward position of each line segment corresponds to the power in the horizontal line. The maximum magnitude of each (horizontal) complex-correlation line occurs at t = 0, on the left, but another (local) maximum occurs at the speech period. This waterfall plot shows the speech period peak wavering down the page. On certain lines, corresponding to unvoiced speech, the peak disappears. Note however, that the magnitude-trace on the left shows that at these times there is little power in the speech.

After the complex correlation, a histogram is incremented at each point in the frequency domain where there was a peak (or resultant peak) in the magnitude spectrum. The best value to use for an increment is the squared magnitude of the complex correlation at its speaker-pitch local peak. Because of the properties of voiced versus unvoiced speech, a common problem in speech processing can simply be finessed. This is the problem of distinguishing between voiced and unvoiced speech. Unvoiced speech is useless for estimating mistuning. However, if we use as a histogram increment the squared magnitude of the complex-correlation peak, this increment will be doubly small during unvoiced speech: First, because the total power is small for unvoiced speech. Second, since this speech is not periodic, the magnitude-squared of the complex-correlation peak will be small, even relative to the total power, which itself is small.

Fig 3 shows the results of incrementing a set of histograms. Using a set of histograms like this instead of just one histogram is not necessary to my method, but here it illustrates the properties of speech signals. Note that the complex correlation provides a series of estimates of speaker pitch, along the way to estimating mistuning. Therefore, we may divide the histogram into a set of subhistograms sorted by speaker pitch. Fig 3 resulted from 10 seconds of Amateur Radio SSB that had not been exactly tuned. Zero-hertz mistuning (after only approximate tuning) is represented by the vertical line in the center. Each subhistogram resulted from a 10-Hz range of speaker pitch. The histograms are noisy, but they clearly show the mistuning. Sighting up and down the figure, we see that the histogram peaks form slanting lines in most portions of the spectrum. For example, one slanting-line of peaks cuts through zero Hz going from the lower left to the upper right. The next series of peaks to the right of this forms a vertical line. This line shows the correct estimate of the mistuning. To the right of the actual mistuning value, the peaks form a line from the lower right to the upper left.

Thus, when a single histogram is formed, a single peak will stand out—if the frequency resolution is coarse. However, for maximum resolution in the estimate, a fine resolution may be used. There will not in general in this case be a single big peak, but rather there will be a small cluster of (more or less) large values in the histogram bordered on both sides by valleys. For example, in Fig 3, we see not only that the peaks line up at the actual mistuning, but also that there are valleys on both sides of the actual mistuning. Therefore, looking for a concentration in the histogram will allow finer resolution and a more accurate estimate than looking for a single peak.

One method that works well is to find the narrowest region of the histogram that contains (say) 80% of the total of increments to date. Then estimate the mistuning to be halfway between the lower and upper limits of this region.

This is the basis of my method. The principle of operation is straightforward. First speech transmitted by SSB is roughly demodulated. As Fig 3 shows, mistuning of ±400 Hz can be tolerated easily. Even greater mistuning is tolerable. The critical limit is reached (1) when too much of the signal is excluded or (2) if the signal is not properly filtered for anti-aliasing. In the latter case, the signal spectrum is folded over onto itself and the "comb" structure is jumbled. Given proper filtering, I expect that mistunings up to +1 kHz can generally be tolerated.

Second, the roughly tuned signal is digitized, preferably at a rate only somewhat greater than 6 kHz. Of course, we must follow the sampling theorem that requires us to sample at more than twice the highest desired frequency. Conversely, this means we must set the low-pass anti-aliasing filter cutoff frequency at somewhat less than half the sampling rate. In addition, the signal should be high-pass filtered at about 300 Hz (ac coupled). In particular, the signal must have a very small dc component.

Next, successive sections of the digitized signal stream are selected. Each section is multiplied by a "window" that has a maximum in its middle and tapers to zero at each end. I use the raised-cosine window.

What length of window is suitable? Each window must have a number of samples that is an integral power of two. In addition, each window should preferably contain three or four periods of voiced speech. In the figures shown above, I used a sampling rate of 6400 Hz and a window of 512 samples

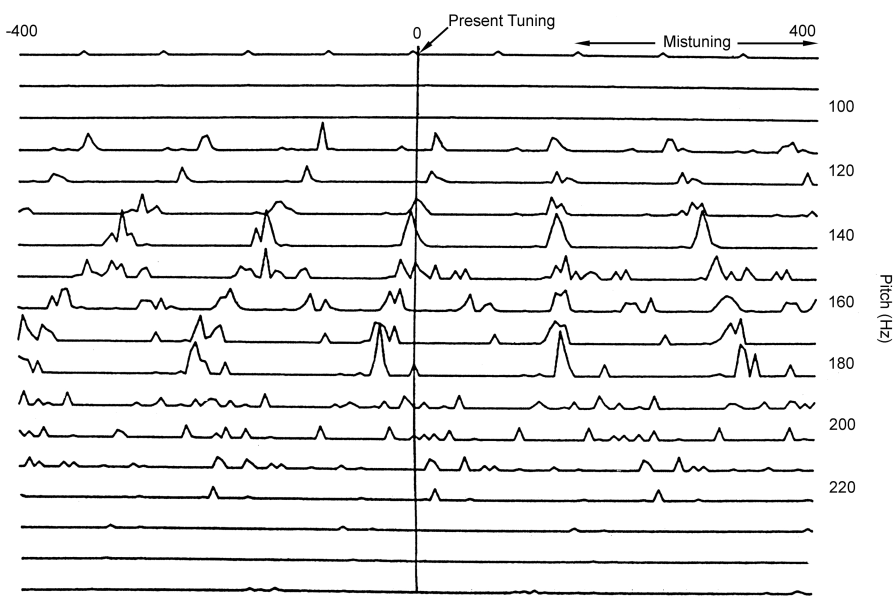


Fig 3—SSB speech pitch versus mistuning.

/ 6400 Hz = 80 milliseconds. This window may have been a little long. I suggest a window somewhere in the range of 40 to 100 milliseconds.

The windows selected may have gaps between them. The duration of the gaps depends on how long it takes your computer to perform a complex correlation and update a histogram. I would estimate that a rate of selection as low as one window per second could be tolerated. What if you have a very fast computer and need no gaps? Then you can let the windows overlap. However, it is best that the windows not overlap by more than 50%, so they are not too redundant.

Appendix 1: Parabolic Interpolation and Peak Finding

Here is how to interpolate three equally spaced data points using a second-order polynomial (ie, a quadratic or parabola).

For x = –1, 0, 1, and y = ym, yz, and yp, we have the three points (–1, ym), (0, yz), (1, yp).

For polynomial coefficients cf0, cf1 and cf2, the formula

$y = cf_0 + cf_1 x + cf_2 x^2$ may be written

$$y = cf_0 + x\left(cf_1 + \left(x \times cf_2\right)\right)$$

It is easy to show that

$$cf_0 = yz,\ cf_1 = \frac{(yp - ym)}{2}, \text{ and } cf_2 = \frac{(yp + ym)}{2} - yz$$

fits a curve to our three (x, y) points. This curve may be variously called a second-order curve, a quadratic or a parabola.

For ym < yz and yz > yp, the peak of this curve is at x = xm, where

$$xm = \frac{0.5(yp - ym)}{2yz - ym - yp}$$

which is the direct formula for

$$xm = \frac{-cf_1}{2cf_2}$$

This is a standard result of analytic geometry. You may have seen it as

$$xm = \frac{-b}{2a} \text{ for } y = ax^2 + bx + c$$

The Software

I have put together functions written in C that do all the steps described above, and also functions that do SSB retuning within the computer. They compiled and ran successfully under Borland Turbo C (Version 1.5) for DOS, an old but serviceable compiler. I also compiled and ran some of the software under Borland Turbo C++ for Windows. I used the fastest FFT high-level language computation method that I was aware of several years ago. I have since learned of Reference 3, which presents a faster FFT than those I used.

What I did requires a little explanation. A complex correlation requires two successive FFTs. Each FFT comes in two parts: one a series of so-called "butterfly" computations and the other a shuffling of the data. This shuffling has the property that doing it twice is a no-op, the same as no shuffling at all. Since it is done in both FFTs, there is a way to avoid shuffling the data. This requires two different FFT methods for the two FFTs. I encoded the two methods separately. The omitted data shuffling is represented in the "bitrev" function. Because the shuffling is omitted, the frequency domain is scrambled. Positive frequencies are represented by the even-numbered locations. Those wishing to experiment to see what things look like in the frequency domain can apply "bitrev" to unscramble it and then plot it.

I have included a little function called "srss." It approximately computes the square root of the sum of two squares *without* computing square roots. The estimated approximation error is given in the source code. It was determined by means of a little calculus, but it can be verified by experiment.

The module with the function "ccor" contains, in addition, a timing routine to estimate how fast your computer can perform complex correlations. (The preprocessor directive "#define MAIN" enables this routine.) I use a digital watch to time from the "Enter" command until 1000 complex correlations are complete. My 486 DX2 PC with a 66 MHz clock (running Windows) can do one "ccor" in about 25 milliseconds for 256 points and about 54 milliseconds for 512 points. Under DOS, it takes 33 ms for 512 points. It turns out that using brute-force methods in "ccor" adds about 20% to the computing time. My 200 MHz Pentium MMX does 512 points in about 6 ms under DOS. A word of caution, if you write your own timing experiment: All-zero data gives an unrealistically optimistic speed of computation. My timing programs use (admittedly nonsensical) nonzero data.

The module "timproc" puts the whole process together and illustrates how to call the various functions I wrote. For timing purposes, I simulated an 8 kHz sampling rate, 512-point ccors, 8 ccors per second, and mistuning estimate once per second. On my 486 it takes 0.6 second of processing for each second of data. The Pentium takes 0.1 second. (Both numbers are for DOS.) As computer speed increases, calculation time may soon be negligible.

The subroutine called "est" takes the output of ccor and estimates speaker pitch and one possible mistuning. Every other possible mistuning is then a multiple of the pitch frequency away from this first estimate. Most of the code in est is the computation of parabolic interpolation. You need this interpolation to get the maximum resolution of mistuning estimates. This is especially necessary when tuning high-pitched speech. For high pitches, even a fairly large change in pitch results in only a small shift of the ccor peak. Further, the same mistuning delta is more accurately measured at low pitches than at high pitches. An appendix shows the mathematics of parabolic interpolation.

Updating and reading the histogram are the final stages of the mistuning-estimator code. I have represented them by key functions. Function "inchist" increments a histogram with data from an array (bins), where the number of array elements is nbins. Function "srchist" searches the histogram for the minimum-width portion containing some fraction (say 80%) of the sum of the increments to date. The only tricky thing about this code is being careful not to go beyond the end of the histogram. As you can see, "srchist" tests for this repeatedly.

The histogram should be zeroed at the start of mistuning estimation and should only be searched after at least several increments have been added. You will probably need to adjust the parameters in your estimator to operate best with your sampling rate and rate of choosing windows. Generally, you'll need about five seconds of speech to refine the tuning within two or three hertz.

The method of retuning is standard for DSP, and I present it here with no claim of invention. Function "retune" runs the process. Given a waveform to be retuned, it is first anti-alias filtered to prevent spectrum folding after the retuning. Function "aafilt" passes the waveform through one of several high-pass FIR filters for a large downshift; through a band-pass filter for a small shift; or one of several low-pass filters for a large upshift. The filters were designed using the "remez" program included with Reference 3. The function "fir" is also from Reference 3.

A single-channel baseband waveform always produces double-sideband modulation. One sideband must be removed to make SSB. To do so one attaches as an imaginary channel, the negative of what is known as the Hilbert transform of the waveform. This is done using function "hilb." Multiplying this complex waveform by a complex exponential retunes it. Only one channel of the result need be saved for output.

References

1. R. J. Dick, "Cochannel Interference Separation," RADC-TR-80-365, Rome Air Development Center, 1980. Available for $19.50 as document ADA096059 from the National Technical Information Service, NTIS, US Department of Commerce, Springfield, VA 22161.
2. A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing* (Englewood Cliffs, New Jersey: Prentice-Hall, 1975).
3. P. M. Embree, *C Algorithms for Real-Time DSP* (Upper Saddle River, New Jersey: Prentice-Hall PTR, 1995). Includes diskette.

Robert Dick, while not a licensed radio amateur, has extensive education and experience in communication and digital signal processing (DSP). He holds a Bachelor of Science in electrical engineering from MIT and a Master of Science and doctorate from Cornell in the mathematical theory of communication. He has worked with delta modulation, sequential and Viterbi decoding, algebraic coding, HF channel simulation, radio "fingerprinting," a decentralized radio network protocol and pattern classification, among other subjects. His most recent DSP work is in multichannel adaptive filtering.

Appendix 2: C Source Code

SOFTWARE.DOC for "Tune SSB Automatically."

This software is furnished as-is, without warranty of any kind. Do not attempt to create working software from this document! It has been reformatted for publication. You can download workable source code from the ARRL at **http://www.arrl.org/files/**. Look for the file SSBTUNE.ZIP. The file includes executable code for TI's DSK3 DSP kit. Here is a list of the functions included:

Function	*File*	*Description*
aafilt	aafilt.c	Anti-alias filter for retuning.
fir	aafilt.c	Finite Impulse Response filter.
retune	aaretun,c	Retune a waveform block. retune process.
ccor	ccor.c	Complex correlation. Does one window.
main	ccor.c	Complex correlation timer.
	ccor.exe	DOS executable of ccor.c.
fwfft	ffts.c	time-to-frequency FFT, minus bit reversal shuffling.
bitrev	ffts.c	Bit reversal data shuffling (not used).
srss	ffts.c	Square-root of sum-of-squares approximation.
rvfft	ffts.c	frequency-to-time FFT, minus data shuffling and 1/n.
hilb	hilb.c	Hilbert transform filter for retuning.
est	histr.c	Estimates pitch and one possible mistuning.
inchist	inchist.c	Increment histogram.
srchist	srchist.c	Search histogram for mistuning.
main	timproc.c	Time whole estimation and
	timproc.exe	DOS executable of timproc.c

/* AAFILT.C */

```
#include <stdio.h>
/*#define DEBUG*/
#define NFILTS 7  /* Number of filters in filter bank */
#define NTAPS 21  /* Number of taps in each fir filter */

float fir(float in, float *tr_coef, int n, float *history);

void aafilt(int remember, float rel_sft, int n_pts, float in[],
        float out[])
/*
 *  Anti-aliasing fir filters routine.  Selects filter from filter bank,
 *  and zeros filter memory, when remember==0.
 *
 *  remember==0:  Determine filter, zero its memory, proceed.
```

```
 *  remember!=0:  Use previously determined filter and its memory
 *  rel_sft = Proposed frequency shift divided by sampling frequency
 *  n_pts = Number of data points to anti-alias filter
 *  in = n_pts data vector in
 *  out          out
 */
{
  static float filtmem[NTAPS-1];  /* filter init cond-memory */
  static int ifilt;  /* index to selected filter */
  /* these are static so can be remembered between calls */

  int idx;

  static float aadat[NFILTS][NTAPS+1] = {
    {-0.12,     /* hipass  0.16  rpl  0.163,  stop  0.12  max gain  0.054 */
    0.0348,     -0.0712,    -0.0334,    0.0024,     0.0401,     0.0585,     0.0321,
    -0.0461,    -0.1540,    -0.2478,    0.7150,     -0.2478,    -0.1540,    -0.0461,
    0.0321,     0.0585,     0.0401,     0.0024,     -0.0334,    -0.0712,    0.0348},

    {-0.08,     /*  hipass  0.12  rpl  0.160, stop  0.08  max  gain  0.053  */
    -0.0613,    0.0386,     0.0432,     0.0450,     0.0345,     0.0060,     -0.0400,
    -0.0963,    -0.1507,    -0.1902,    0.7954,     -0.1902,    -0.1507,    -0.0963,
    -0.0400,    0.0060,     0.0345,     0.0450,     0.0432,     0.0386,     -0.0613},

    {-0.04,     /*  hipass  0.08  rpl  0.150,  stop  0.04  max  gain  0.050  */
    0.0773,     0.0017,     -0.0067,    -0.0202,    -0.0379,    -0.0581,    -0.0790,
    -0.0982,    -0.1139,    -0.1241,    0.8724,     -0.1241,    -0.1139,    -0.0982,
    -0.0790,    -0.0581,    -0.0379,    -0.0202,    -0.0067,    0.0017,     0.0773},

    {0.04,      /*  bandpass  0.04  to  0.46  rpl  0.160,  stops  0.01  &  0.49  */
    -0.1105,    0.,         -0.0686,    0.,         -0.0832,    0.,         -0.0950,
    0.,         -0.1027,    0.,         0.8946,     0.,         -0.1027,    0.,
    -0.0950,    0.,         -0.0832,    0.,         -0.0686,    0.,         -0.1105},

    {0.08,      /*  lopass  0.42  rpl  0.150,  stop  0.46  max  gain  0.050  */
    0.0772,     -0.0017,    -0.0066,    0.0202,     -0.0379,    0.0581,     -0.0790,
    0.0983,     -0.1139,    0.1240,     0.8724,     0.1240,     -0.1139,    0.0983,
    -0.0790,    0.0581,     -0.0379,    0.0202,     -0.0066,    -0.0017,    0.0772},

    {0.12,      /*  lopass  0.38  rpl  0.160,  stop  0.42  max  gain  0.053  */
    -0.0613,    -0.0387,    0.0431,     -0.0450,    0.0346,     -0.0060,    -0.0401,
    0.0963,     -0.1507,    0.1902,     0.7954,     0.1902,     -0.1507,    0.0963,
    -0.0401,    -0.0060,    0.0346,     -0.0450,    0.0431,     -0.0387,    -0.0613},

    {0.16,      /*  lopass  0.34  rpl  0.163,  stop  0.38  max  gain  0.054  */
    0.0348,     0.0712,     -0.0334,    -0.0024,    0.0401,     -0.0585,    0.0321,
    0.0461,     -0.1541,    0.2478,     0.7151,     0.2478,     -0.1541,    0.0461,
    0.0321,     -0.0585,    0.0401,     -0.0024,    -0.0334,    0.0712,     0.0348}};

  if(remember==0)
  {  for(idx=0; idx<NTAPS-1; idx++)
     {  filtmem[idx]=0.;
     }
     for(ifilt=0; ifilt<NFILTS-1 && aadat[ifilt][0]<rel_sft; ifilt++);
     /* want range 0 <= ifilt < NFILTS */

  }

#ifdef DEBUG
  printf("Ifilt %d, relcut %g.\n",ifilt,aadat[ifilt][0]);
#endif

  for(idx=0; idx<n_pts; idx++)
  {  out[idx]=fir(in[idx], &aadat[ifilt][1], NTAPS, filtmem);
  }
}
```

/* FIR filter routine */

```
float fir(float in, float *tr_coef, int n, float *history)
/*
 *  finite impulse response (fir) filter computation,
 *  adapted from Embree listing 4.1.
 *
 *  in = one floating-point input
 *  tr_coef = time-reversed or time-symmetric impulse response
 *  n = number of terms of tr_coef.  Must be odd and >=5.
 *  history = n-1 size array of past inputs and/or initial conditions
 *
 *  returns one floating-point output
 */
{
  int i;
  float *hist_ptr, *hist1_ptr, *tr_coef_ptr;
  float out;

  hist_ptr=history;
  hist1_ptr=hist_ptr;  /* use for history update */
  tr_coef_ptr=tr_coef;  /* fir coefs must be symmetric or time reversed */

  /* accumulate output */
  out=*hist_ptr++ * (*tr_coef_ptr++); /* 1st contribution */
  *hist1_ptr++ = *hist_ptr;  /* update a term of history array */
  out += (*hist_ptr++) * (*tr_coef_ptr++);  /* 2nd contribution */
  for(i=3; i<n; i+=2)
  {  *hist1_ptr++ = *hist_ptr;
     out += (*hist_ptr++)*(*tr_coef_ptr++);  /* ith contribution */
     *hist1_ptr++ = *hist_ptr;
     out += (*hist_ptr++)*(*tr_coef_ptr++);  /* (i+1)st contribution */
  }
  *hist1_ptr=in;  /* put input into history */
  out += in * (*tr_coef_ptr);  /* nth contribution */

  return(out);
}
```

/* AARETUN.C */

```
#include <stdio.h>
#include <math.h>
#include "c:\turboc\ssb\hilb.c"
#define SNGL_LEN 1024

void aafilt(int remember, float rel_shift,
        int n, float in[], float out[]);

void retune(int remember, float rel_shift,
        int n, float in[], float out[])
/*
 *  Anti-aliasing SSB retuner.
 *  Use -1.<rel_shift<0.5.  Fraction of sampling rate to shift.
 *  Example:  rel_shift==-0.5 inverts frequencies.
 *  n is num points in and out.  Need 15 < n <= SNGL_LEN.
 *  Anti-alias filter, attach imag part, retune.
 *  Calls aafilt for anti-aliasing.  Calls hilb to null neg freqs.
 */
{
  static float inbuf[SNGL_LEN];  /* input anti-alias filtered */
  static float rbuf[SNGL_LEN], hilbuf[SNGL_LEN], ibuf[SNGL_LEN];
  static float oldrbuf[15];  /* static so preserved between calls */
  /*  hilbuf=Hilbert transform output buffer==minus imag part.
   *  oldrbuf=old reals buffer for 15-sample delay to line up with
   *        Hilbert transform output.
   */
  float pi=3.14159265358979, twstr, twsti, turnr, turni;
  /*  twst=unit-magnitude complex number with angle in radians
   *      2*pi*rel_shift=phase angle advance per sample
   *  turn=repeatedly twisted unit-magnitude complex number.
   */
  int idx;
  float tmp;

  if(remember==0)
  {  for(idx=0;idx<15;idx++)
     {  oldrbuf[idx]=0.;
     }
     turnr=1.;
     turni=0.;
  }
  twstr=cos(2.0*pi*rel_shift);
  twsti=sin(2.0*pi*rel_shift);

  /* prefilter to prevent postshift aliasing */
  aafilt(remember, rel_shift, n, in, inbuf);

  for(idx=0;idx<15;idx++)
  {  rbuf[idx]=oldrbuf[idx];
     oldrbuf[idx]=inbuf[n-15+idx];
  }
  for(idx=0;idx<n-15;idx++)
  {  rbuf[15+idx]=inbuf[idx];
  }
```

```
   hilb(remember, n, inbuf, hilbuf);
   for(idx=0;idx<n;idx++)
   { /*  Imaginary part is minus Hilbert transform of real part  */
     ibuf[idx]=-hilbuf[idx];
   }

   for(idx=0; idx<n; idx++)
   { /*  Mult by complex exponential to shift freq.  Save reals.  */
     out[idx]=rbuf[idx]*turnr - ibuf[idx]*turni;
     tmp=twstr*turnr - twsti*turni;
     turni=twstr*turni + twsti*turnr;
     turnr=tmp;
   }
}

#ifdef MAIN
int main()

/*  Timing test of retuner via 1000 point blocks  */

{
  int remember, nblks, idx, jdx;
  static float dat[SNGL_LEN], out[SNGL_LEN];
  float rel_shift;

  for(idx=0;idx<SNGL_LEN;idx+=32)
  {  /* fill input data buffer with dummy data for test purposes  */
    for(jdx=0;jdx<32;jdx++)
    {  if(idx+jdx<SNGL_LEN)
      {  dat[idx+jdx]=15-jdx;
      }
    }
  }

  nblks=1;
  while(nblks>0)
  {  printf("Enter no-resamp relative frequency shift.\n");
    scanf("%f",&rel_shift);
    printf("Enter number of 1000-point blocks to retune.\n");
    scanf("%d",&nblks);
    remember=0;
    for(idx=0;idx<nblks;idx++)
    {  retune(remember,rel_shift,1000,dat,out);
      remember=1;
    }
    printf("Done.\n");
  }
}
#endif
```

/* CCOR.C */

```
#include<stdio.h>
#include"c:\turboc\ssb\ffts.c"   /*  laptop version  */
#include"c:\turboc\ssb\histr.c"  /*  laptop version  */
/*#define MAIN*/

void ccor(int n,float dat[],float w[],float xr[],float xi[])

/*  "Complex Correlation" t->f->t function             */

/*  int   n;              Must be a positive power of 2 */
/*  float dat[];          Input (time) waveform section */
/*  float w[];            Data Window (raised cosine)   */
/*  float xr[], xi[];     Real & Imag data vectors      */
{
  int idx;

/*  Windowed data -> real vector, zeros -> imag vector */
  for (idx = 0; idx < n; idx++)
  {
    xr[idx] = dat[idx]*w[idx];
    xi[idx] = 0.0;
  }
  fwfft(n,xr,xi);
/*  Now in scrambled-index frequency domain            */
/*  Form magnitudes at pos freqs, zeros elsewhere      */
/*  Scrambled pos freq <=> even-numbered index         */
  xr[0]=0.0; /* DC is still zero index when scrambled */
  xi[0]=0.0;
  xr[1]=0.0;
  xi[1]=0.0;
  for (idx = 2; idx < n; idx = idx + 2)
  {
    xr[idx]   = srss(xr[idx],xi[idx]);
    xi[idx]   = 0.0;
    xr[idx+1] = 0.0;
    xi[idx+1] = 0.0;
  }
  rvfft(n,xr,xi);
/*  Now xr & xi are "complex correlation"  */
}

#ifdef MAIN
```

/* "Main" Routine */

```
int main()

/*  Timing test of complex correlation plus mistuning estimator.  */

{
  int n, idx, jdx, iter, niter;
  float pi=3.14159265358979, smprt=6400., peak, pitch, mstn;
  static float dat[2048], w[2048], xr[2048], xi[2048];

  for(idx=0;idx<2048;idx=idx+32)
  {
    for(jdx=0;jdx<32;jdx++)
          dat[idx+jdx]=15-jdx;
  }
  n=1;
  while(n>0)
  {
    printf("\n  Enter fft size.  ");
    scanf("%d",&n);
    for(idx=0;idx<n;idx++)
          w[idx]=1.-cos((2.*pi*idx)/n);
    printf("  Enter num ccor & est its.  ");
    scanf("%d",&niter);
    for(iter=0;iter<niter;iter++)
    {
          ccor(n,dat,w,xr,xi);
          est(n,xr,xi,smprt,&peak,&pitch,&mstn);
    }
    printf("  Done.");
    printf("\n  Pitch %4.4f, mistun %4.4f",pitch,mstn);
    printf("\n  Enter 0 to stop.");
    scanf("%d",&n);
  }
  return 0;
}
#endif
```

/* FFTS.C */

```
#include<math.h>
```

/* FWFTT Routine */

```
void fwfft(int n,float xr[],float xi[])

/*  t->f FFT omitting post-butterflies bit-reversal  */
/*  From Oppenheim & Schaffer Fig. P6.5 page 332     */

/*  int   n;       Must be a positive power of 2      */
/*  float xr[], xi[];      Real & Imag data vectors  */
{
  float pi = 3.14159265358979;
  float tmpr, tmpi;  /*  Temporaries              */
  int idx, iptr, jptr, leap, jump; /*  Indices      */
  float twstr, twsti, turnr, turni;

  leap = n;
  while (leap > 1)
  {
    jump = leap>>1;  /*  Right shift 1  */
    turnr = 1.0;
    turni = 0.0;
    twstr =  cos(pi/jump);
    twsti = -sin(pi/jump);  /*  For t->f  */
    for (idx = 0; idx < jump; idx++)
```

```
  {
    for (iptr = idx; iptr < n; iptr = iptr + leap)
    {
      jptr = iptr + jump;
      tmpr = xr[iptr] - xr[jptr];
      tmpi = xi[iptr] - xi[jptr];
      xr[iptr] = xr[iptr] + xr[jptr];
      xi[iptr] = xi[iptr] + xi[jptr];
      xr[jptr] = tmpr*turnr - tmpi*turni;
      xi[jptr] = tmpr*turni + tmpi*turnr;
    }
    tmpr  = turnr*twstr - turni*twsti;
    turni = turnr*twsti + turni*twstr;
    turnr = tmpr;
  }
  leap = jump;
 }
```

/* Bit-reversal data shuffling omitted from here */

```
}

int bitrev(int n,float xr[],float xi[])

/*  BIT-REVersal data shuffling
    From "Cochannel" page B-8
    From Oppenheim & Schafer Fig. P6.5 page 332
    Doing it twice is a no-op

int n;    Must be a positive power of 2
float xr[], xi[];    Data vectors  */
{
  int nv2;              /*  n/2  */
  int ibtrv, idx, iptr;  /*  ibtrv is "bit-reversed" index  */
  float tmpr, tmpi;      /*  Temporaries  */

  ibtrv = 0;
  nv2 = n/2;
  for (idx = 0; idx < n-1; idx++)
  {
    if (idx < ibtrv)
    {  /* Swap buffer points */
      tmpr = xr[ibtrv];
      tmpi = xi[ibtrv];
      xr[ibtrv] = xr[idx];
      xi[ibtrv] = xi[idx];
      xr[idx] = tmpr;
      xi[idx] = tmpi;
    }
    /*  Increment the bit-reversed index  */
    iptr = nv2;
    while (ibtrv >= iptr)
    {
      ibtrv = ibtrv - iptr;
      iptr = iptr/2;
    }
    ibtrv = ibtrv + iptr;
  }
  return 0;
}
```

/* SRSS Routine */

```
float srss(float x,float y)

/*  Square Root of Sum of Squares Approximator  */
/*  Approx sqrt(x*x+y*y) without squareroot     */
/*  RMS error 2.4 percent (32 dB down)          */

{
  float magx, magy, ans;

  magx=fabs(x);
  magy=fabs(y);
  if (magx > magy)
    ans = 0.95*magx + 0.4*magy;
  else
    ans = 0.95*magy + 0.4*magx;
  return ans;
}
```

/* RVFFT Routine */

```
void rvfft(int n,float xr[],float xi[])

/*  Reverse FFT (f->t) omitting bit-rev and div by n  */
/*  Based on "Cochannel" CFFT pages B-8 to B-9        */

/*  int   n    A positive power of 2                  */
/*  float xr[], xi[]    Real & Imag data vectors      */
{
  float pi = 3.14159265358979;
  float tmpr, tmpi;  /*  Temporaries  */
  int idx, iptr, jptr, leap, jump;  /*  Indices  */
  float twstr, twsti, turnr, turni; /*  Rotations  */

/*  Bit-reversal data shuffling omitted from here  */
  leap = 1;
  while (leap < n)
  {
    jump = leap;
    leap = leap*2;
    twstr = cos(pi/jump);
    twsti = sin(pi/jump);  /*  for f->t  */
    turnr = 1.0;
    turni = 0.0;
    for (idx = 0; idx < jump; idx++)
    {
      for (iptr = idx; iptr < n; iptr = iptr + leap)
      {
        jptr = iptr + jump;
        tmpr = xr[jptr]*turnr - xi[jptr]*turni;
        tmpi = xr[jptr]*turni + xi[jptr]*turnr;
        xr[jptr] = xr[iptr] - tmpr;
        xi[jptr] = xi[iptr] - tmpi;
        xr[iptr] = xr[iptr] + tmpr;
        xi[iptr] = xi[iptr] + tmpi;
      }
      tmpr  = turnr*twstr - turni*twsti;
      turni = turnr*twsti + turni*twstr;
      turnr = tmpr;
    }
  }
/*  Dividing data by n omitted from here  */
}
```

/* HILB.C */

```
#define BUFLEN 1054

void hilb(int remember, int n, float in[], float out[])
/* remember==0 is signal to zero internal memory before proceeding
 * n is number of points in and out.  Need 0<n<=BUFLEN-30
 * in is input
 * out is output, delayed 15 points
 */
{
  static float buf[BUFLEN];  /* 30 pts old input then n pts new  */
  float tap[8] = {0.0205, 0.0213, 0.0326, 0.0488,
              0.0730, 0.1140, 0.2040, 0.6338};
  /* 31-tap filter, every second tap zero, antisymmetric
   * gain 0.976 to 1.024 in 0.03 to 0.47 sample rate
   */
   int ipt, idx;
   float sum;

  if(remember==0)
  {
    for(idx=0;idx<30;idx++)
    {
      buf[idx]=0.;
    }
  }
  for(idx=0;idx<n && idx<BUFLEN-30;idx++)
  {  /*  add new input data to buffer  */
    buf[idx+30]=in[idx];
  }
  for(idx=0;idx<BUFLEN-30 && idx<n;idx++)
  {  /*  form one point of output  */
    sum=0.;
    for(ipt=0;ipt<8;ipt++)
    {  /*  add effects of two taps with same magnitude  */
      sum+=(buf[idx+16+2*ipt] -
```

```
            buf[idx+14-2*ipt])*tap[7-ipt];
    }
    out[idx]=sum;
  }
  for(idx=0;idx<30 && idx+n<BUFLEN;idx++)
  {  /*  save last 30 points of input  */
    buf[idx]=buf[idx+n];
  }
}
```

/* HISTR.C */

```
#include<math.h>
```

/* EST Routine */

```
void est(int n,float xr[],float xi[],float smpfrq,
        float *ppeak,float *ppitch,float *pshift)
/*  Estimator of pitch and one (out of many possible) freq shift  */
/*  See "Cochannel" page B-21                                    */
/*  n;          Buffer size (power of 2)                         */
/*  xr[],xi[];    Real, imag buffers from ccor                   */
/*  smpfrq;       Sampling frequency                             */
/*  peak;         Peak power at speaker pitch                    */
/*  pitch;        Estimated speaker pitch                        */
/*  shift;        One possible frequency shift                   */
{
  float pi=3.14159265358979;
  float lolim,hilim; /* low, high limits for speaker_pitch search range */
  int lidx,hidx; /* low, high index limits for tau search range */
  float temp;
  int idx, idpk; /* index and index_to_peak                   */
  float bsq,usq; /* below_square and upper_square              */
  float x; /* fractional spacing of interpolated peak, -1<=x<=1 */
  float tau; /* sampling_frequency/speaker_pitch               */
  float cf1,cf2; /* coefficients for parabolic interpolation    */
  float partr,parti; /* real, imag, parts of ccor peak          */
  float angl; /* Shift_angle in radians versus speaker pitch    */

  lolim=50; /* Assumes speaker pitch >= 50 Hz */
  hilim=250; /* Assumes speaker pitch <= 250 Hz */
  lidx=smpfrq/hilim; /* hi pitch is low time delta  */
  if(lidx<4)lidx=4;
  hidx=smpfrq/lolim;
  if(hidx>n/2-2)hidx=n/2-2;
  *ppeak=0.;
  idpk=4;  /*  2-18-98  */
  for(idx=lidx;idx<=hidx;idx++)
  {
    temp=xr[idx]*xr[idx]+xi[idx]*xi[idx];
    if(*ppeak<temp)
    {
      *ppeak=temp;
      idpk=idx;
    }
  }
  /* Find quadratic-interpolation peak */
  bsq=xr[idpk-1]*xr[idpk-1]+xi[idpk-1]*xi[idpk-1];
  usq=xr[idpk+1]*xr[idpk+1]+xi[idpk+1]*xi[idpk+1];
  x=1.;
  if(*ppeak>usq)
    x=0.;
  if(bsq>=*ppeak)
    x=-1.;
  if(x==0.)
    x=0.5*(usq-bsq)/(2.* *ppeak-bsq-usq);
  tau=idpk+x;
  *ppitch=smpfrq/tau;
  /* Interpolate real and imag parts */
  cf1=0.5*(xr[idpk+1]-xr[idpk-1]);
  cf2=0.5*(xr[idpk-1]+xr[idpk+1])-xr[idpk];
  partr=xr[idpk]+x*(cf1+x*cf2);
  cf1=0.5*(xi[idpk+1]-xi[idpk-1]);
  cf2=0.5*(xi[idpk-1]+xi[idpk+1])-xi[idpk];
  parti=xi[idpk]+x*(cf1+x*cf2);
  *ppeak=partr*partr+parti*parti;
  /* calculate 4-quadrant arctangent (-pi/2 to 3pi/2) */
  if(partr>0.)
    angl=atan(parti/partr);
  if(partr==0.)
  {
    if(parti>=0.)
      angl=0.5*pi;
    else
      angl=-0.5*pi;
  }
  if(partr<0.)
    angl=pi-atan(-parti/partr);
  *pshift=*ppitch*angl/(2.*pi);
}
```

/* INCHIST.C */

```
/*  Increment Histogram  */

void inchist(int nbins, float bins[],
                float hpitch, float hshift, float incr)

/*  nbins   Number of histogram bins               */
/*  bins    The array of histogram bins            */
/*  hpitch  Est spkr pitch in num of bins spanned  */
/*  hshift  One poss freq shift in histogram units */
/*  incr    Amount to increment each selected bin  */

/*  For lowf=min freq of bin 0 (lowf may be < 0),   */
/*  And hif=min freq of bin (nbins-1),              */
/*  hpitch = pitch*nbins/(hif-lowf) and             */
/*  hshift =(shift-lowf)*nbins/(hif-lowf).          */

{
  float fidx;  /*  Floating point histogram index  */
  int   idx;   /*  Integer histogram index         */

  if(hpitch>=1.0  && 0.0<=hshift && hshift<nbins && incr>0.0)
  {
    fidx=hshift;
    while(fidx>=0.0)
    {
          idx=fidx;
          bins[idx]=bins[idx]+incr;
          fidx=fidx-hpitch;
    }
    fidx=hpitch+hshift;
    while(fidx<nbins)
    {
          idx=fidx;
          bins[idx]=bins[idx]+incr;
          fidx=fidx+hpitch;
    }
  }
}
```

/* SRCHIST.C */

```
/*  Search Histogram  */

void srchist(int nbins, float bins[], float thresh,
                int *pminwid, float *pctr)

/*  nbins   Number of histogram bins                  */
/*  bins    The histogram bins themselves             */
/*  thresh  Threshold for sum of bins                 */
/*  minwid  Min width-1 of interval with sum >= thresh */
/*  ctr     Center of minwid bins interval            */
/*  Note:   Call using arguments &minwid and &ctr     */

{
  int lidx,hidx,oldlow,minlow;
  float sum;

  *pminwid=nbins;
  lidx=hidx=oldlow=minlow=0;
  sum=bins[0]; /* 2-18-98 */

  while(hidx<nbins)
  {
    while(sum<thresh && hidx<nbins)
    {  /*  Advance forward limit until sum>=thresh  */
          hidx++;
          if(hidx<nbins)
            sum=sum+bins[hidx];
    }
    while(sum>=thresh && lidx<=hidx && hidx<nbins)
```

```
    { /*  Advance rear limit until sum<thresh  */
            sum=sum-bins[lidx];
            oldlow=lidx;
            lidx++;
    }
    if(hidx<nbins)
    {
            if(hidx-oldlow<*pminwid)
            { /*  If intvl narrowest so far note it  */
              *pminwid=hidx-oldlow;
              minlow=oldlow;
            }
    }
  }
  *pctr=minlow+*pminwid*0.5;
}
```

/* TIMPROC.C

```
 *  Time complete est-tune and retune process using dummy
 *  speech data.  Retune BEFORE tuning estimator for test
 *  purposes.
 *  Compile with AAFILT.C, AARETUN.C, CCOR.C, INCHIST.C,
 *  and SRCHIST.C.
 */

#include <stdio.h>
#include <math.h>

void retune(int rem,float shf,int n,float in[],float out[]);
void ccor(int n,float in[],float win[],float xr[],float xi[]);
void est(int n,float xr[],float xi[],float smf,
            float *pk,float *ptch,float *shf);
void inchist(int nb,float bs[],float hptch,float hsft,float incr);
void srchist(int nb,float bs[],float thr,int *mwid,float *ctr);

#define SAMP_RATE  8000.0 /*  Nominal sampling rate in Hz     */
#define FFT_N      512   /*  FFT and IFFT block length       */
#define BLKS_TO_EST   8   /*  Num input blks per mistune est  */
#define LOW_HIST_F -800.0 /*  Low limit histogram frequency   */
#define HI_HIST_F   800.0 /*  Highest hist bin bottom freq    */
#define HIST_N     1600   /*  Number of histogram bins        */
#define SUM_PCT      80.0 /*  Percent of sum of histogram increments
                               *  to be in min width histogram interval
                               */
#define HZTOBINS (HIST_N)/((HI_HIST_F)-(LOW_HIST_F))  /*  Bins per Hz  */
#define RETUNE_N ((int)(SAMP_RATE)/(BLKS_TO_EST)) /* Retune blocklen */
```

/* "Main" Routine */

```
main ()
{
  static float in[RETUNE_N], retout[RETUNE_N];
  /*  in = admittedly nonsensical input data
   *  retout = retuner output data
   */
  static float win[FFT_N], fftr[FFT_N], ffti[FFT_N];
  /*  win = raised-cosine data window
   *  fftr = fft and complex correlation real part
   *  ffti =                             imag part
   */
  static float histbins[HIST_N];
  /*  The bins of the power versus frequency histogram
   */
  float act_shift, ccorppow, pitch, estshift, suminc;
  /*  act_shift = Actual sim mistuning freq shift in sim Hz
   *  ccorppow = height of ccor (interpolated) peak mag-squared
   *  pitch = value of ccor estimated speech pitch in Hz
   *  estshift = ccor est of (one possible) freq shift in Hz
   *  suminc = sum of hist increments since last his reset
   */
  int idx, jdx, kdx, itau, nmloop, mloopx, minwid;
  /*  minwid = num-1 hist bins in min width interval w sum >= thresh
   */
  float ctr, shift_hz, f_delt;
  /*  ctr = est in binwidths of mistuning minus LOW_HIST_F
   *  shift_hz = ctr converted to Hz minus zero frequency
   *  f_delt = 0.5 * (minwid+1) * (bin width in Hz)
   */

  for(idx=0; idx<FFT_N; idx++)
  { /*  Define raised-cosine window  */
    win[idx]=1.0-cos(2.0*3.1415926*(idx+0.5)/(FFT_N));
  }
  nmloop=1;
  while(nmloop>0)
  {
    printf("FFT pts %d, hist bin %g Hz wide, range %g to %g.\n",
              FFT_N,       1./(HZTOBINS), LOW_HIST_F,HI_HIST_F);
    printf("Min width hist intvl %g pct incr fm reset.\n",SUM_PCT);
    printf("Enter retune freq Hz (samp rate %g Hz).\n",SAMP_RATE);
    scanf("%f",&act_shift);
    printf("Enter number of times thru (nom 1 sec) loop.\n");
    scanf("%d",&nmloop);
    for(mloopx=0;mloopx<nmloop;mloopx++)
    {
          for(idx=0;idx<HIST_N;idx++)
          { /*  Clear histogram  */
            histbins[idx]=0.;
          }
          suminc=0.;  /*  Reset sum of histogram increments  */
          for(idx=0;idx<BLKS_TO_EST;idx++)
          { /*  Vary dummy rep period (tau) from low to high  */
            itau=40 + 11*idx;  /*  tau increment shd be prime  */
            for(jdx=0;jdx<RETUNE_N;jdx+=itau)
            { /*  sawtooth sims speech  */
              for(kdx=0;kdx<itau && jdx+kdx<RETUNE_N;kdx++)
              {  in[jdx+kdx]=(1-(2.*kdx+1)/(float)itau);
              }
            }

            retune(0, act_shift/(SAMP_RATE), RETUNE_N, in, retout);
            /*  Skip first 64 points of retuner output  */
            ccor(FFT_N, &retout[64], win, fftr, ffti);
            est(FFT_N,fftr, ffti, SAMP_RATE,
                    &ccorppow, &pitch, &estshift);
            suminc+=ccorppow;
            inchist(HIST_N, histbins,
                      pitch*HZTOBINS, (estshift-LOW_HIST_F)*HZTOBINS,
                      ccorppow);
          }

          srchist(HIST_N, histbins, (SUM_PCT*0.01)*suminc,
                      &minwid, &ctr);
          shift_hz=LOW_HIST_F+(ctr+0.5)/(HZTOBINS);
          f_delt=(minwid+1.0)/(2.0*HZTOBINS);  /* minwid==0 for 1 bin */
    }
    printf("Done.\n");
    printf("Est shift %g Hz, +/- %g.\n",shift_hz,f_delt);
  }

}
```

The Need for Standard Application-Programming Interfaces (APIs) in Amateur Radio

Are you tired of requesting your favorite ham software developer to support your favorite rig or device?

By Lawrence G. Dobranski, VA3LGD/VE3TVV

The Problem

Some recent discussions on Internet e-mail reflectors supporting some of the popular contesting and logging software reinforced the lack of programming standards within the Amateur Radio community. When a new radio or other product is released, this lack of standardization causes developers to scramble; they must modify their programs to interface to the new equipment. Many software developers do not provide Amateur Radio software as their primary occupation. Finding the specifications, getting access to the equipment and verifying the interfaces can be a continual and significant hardship.

14 Sandhead Terrace
Nepean, Ontario, K2J 1L4
va3lgd@amsat.org

The Solutions

Two solutions exist for this problem. The first is to have ham-radio equipment manufacturers develop standard interfaces and command sets. Given the ham-radio equipment manufacturers' inability to agree on the wiring of the microphone connector, the possibility of getting a standard command set developed, approved and used is unlikely.

The second solution is based on a similar problem that has already been solved in the software-development world. Today's operating systems provide many features to application developers. The features and services are usually accessed through an *application-programming interface* (API). APIs provide a set of function calls that application developers use. The software that implements these calls performs the lower-level functions of the device or operating system.

What is an API?

To understand what an API is, let's review how one works through an example. Most contesting and logging programs use the computer interface on Amateur Radio transceivers. Through this interface, they read and set frequency, band and other information. To set the radio's frequency, the program must convert the data to a numerical format the radio can understand, format the data within an appropriate command structure and send the result to the radio. [Acknowledgment from the transceiver may also be required.—*Ed.*] Similar operations are performed to set or read other rig data. Today, these programs must support many different rigs from many different manufacturers, each with its own command set and data format. The protocols may be quite different, and the command sets mutually exclusive.

Instead of the application software composing the radio command directly, it might call a standard software function instead—HAM_API_Rig_SetFrequency(x)—where x is the desired frequency. This API call is translated by a radio-specific library into the radio's command format. When new radios are released, a new radio-specific library is developed, rather than modifying the application software. With the addition or update of the new library, all existing applications that use the HAM API would then be able to interface to the radio.

Where in Amateur Radio Would We Use Them?

Amateur Radio is becoming increasingly computerized. In many of our shacks, computers are interfaced to our rigs, TNCs, rotators, voice keyers, CW keyers, antenna switches, GPSs, etc. When software is developed to aid the amateur, it must be built to support specific equipment. If Amateur Radio APIs existed, then we would only need to develop specific libraries. The application would no longer have to be modified.

Table 1 lists a sample of the API functions that could be developed for amateur use.

Two Sample APIs

To see the effectiveness of APIs, Tables 2 and 3 define samples that might be used for rig control and interfacing to a PacketCluster. The style used in the definition depicts the API as a set of functions. It could be defined in terms of object-oriented-programming constructs as well.

How Do We Develop Them?

If this approach to computer control of amateur equipment is acceptable, interested amateurs must develop working groups to author the respective APIs. These working groups could discuss their development—using the Internet, for example—and author the various libraries. Once an API is developed, it would not be considered a reference standard until two unrelated applications use the API to control two different devices. A test suite is then developed to verify that future API implementations meet the standard. This conformance test ensures the user that the API implementation will work with their application.

Before API standard development begins, a standard naming convention for Amateur Radio APIs should be developed. For example, a proposed naming convention is as follows:

HAM_API_*xxx_yyyy*(). *xxx* is the API name (ie, rig, tnc, rotor) and *yyyy* is the function name. Variables and constants are named in a similar way.

How Do We Implement Them?

Linux

Linux and all UNIX derivatives support run-time libraries. A chapter in *Linux Application Development*[1] describes *Linux's* shared libraries and how to implement them. Each Ham API device-specific library would be implemented as a shared library.

Windows

Win16 (the formal name for the Win 3.1X environment) and Win32 (Win 9X and NT) provide support for run-time libraries. In the Windows environment, these libraries are called Dynamic Link Libraries (DLLs). Their file name extension is ".dll". A good portion of the Windows operating system is implemented in DLLs. Each Ham API device-specific library would be implemented as a DLL.

DOS

The Microsoft *DOS* operating environment presents an interesting challenge when trying to implement

[1] *Linux Application Development*, by Michael K. Johnson and Erik W. Troan, published by Addison Wesley Longman Inc, 1998.

Table 1
Proposed API Classifications

Amplifier Control
Antenna Switch
Call book interfaces
CW Contest Keyers
Digital Voice Keyers
GPS Data
PacketCluster
Rig Control
Rotor Control
Satellite Trackers
TNC Control

Table 2
A Sample API for Rig Control

Function	*Description*
HAM_API_Rig_getName()	Returns the rig name and model number
HAM_API_Rig_getCaps()	Returns—in a standard data structure—information about the rig's capabilities: mode, frequency range, output power, etc
HAM_API_Rig_selectRig()	Sets the active rig for subsequent commands. Use when more than one rig is controlled by the computer
HAM_API_Rig_getSettings()	Returns—in a standard data structure—current rig settings: frequency, mode, split, etc
HAM_API_Rig_getFrequency()	Returns the rig frequencies
HAM_API_Rig_setFrequency()	Sets the rig frequency
HAM_API_Rig_setEventFunction()	Sets the function to be executed if a rig generated event happens, ie, frequency changed from rig's front panel
HAM_API_Rig_getEvent()	Returns the event that caused the setEventFunction to be activated
HAM_API_Rig_setRIT()	Sets the receive incremental tuning (RIT)
HAM_API_Rig_setXIT()	Sets the transmit incremental tuning (XIT)
HAM_API_Rig_setMode()	Sets the rig's mode
HAM_API_Rig_getMode()	Get the rig's mode

Table 3
A Sample API for Interfacing to a PacketCluster

Functional	*Description*
HAM_API_Packet_Cluster_login()	Login in to the PacketCluster
HAM_API_Packet_Cluster_setName()	Set the operator's name
HAM_API_Packet_Cluster_setQTH()	Set the operator's QTH
HAM_API_Packet_Cluster_doSetCommand()	Sends a set command to the PacketCluster. The command is contained in a standard data structure that also contains the result of the command
HAM_API_Packet_Cluster_doDirCommand()	Sends a Dir command to the PacketCluster. The command is contained in a standard data structure that also contains the result of the command
HAM_API_Packet_Cluster_doShowCommand()	Sends a Show command to the PacketCluster. The command is contained in a standard data structure that also contains the result of the command
HAM_API_Packet_Cluster_delete()	Sends a command to delete a mail message
HAM_API_Packet_Cluster_send()	Sends a mail message
HAM_API_Packet_Cluster_announce()	Sends an announcement. The type of announcement is contained in standard data structure
HAM_API_Packet_Cluster_quit()	Sends the command to log off the node
HAM_API_Packet_Cluster_dx()	Announces a DX station
HAM_API_Packet_Cluster_reply()	Reply to a read message
HAM_API_Packet_Cluster_talk()	Enter talk mode
HAM_API_Packet_Cluster_type()	Enter a command to display a file. File contents are returned in the data structure
HAM_API_Packet_Cluster_upload()	Uploads a bulletin file
HAM_API_Packet_Cluster_wwv()	Gets the solar flux
HAM_API_Packet_Cluster_read()	Sends a command to read a message into the data structure.

standard libraries supporting the API. No one standard run-time-library module has emerged. Standard linking libraries are used at compile and link time, but often, no real run-time library module exists. Instead, software developers have made use of the architecture of the Intel iAPX-86 family, for which *DOS* was developed. They use the same method by which *DOS* communicates with underlying basic input/output system (BIOS) firmware and software; that is, via software interrupts and terminate-and-stay-resident (TSR) techniques.

The Intel iAPX-86 architecture provides support for up to 256 software interrupts. Like their close cousins, hardware interrupts, software interrupts are invoked by asserting an interrupt request (IRQ). Instead of being generated via hardware, software interrupts are requested through software instructions. For example, *DOS* provides a function for printing a character to the standard output device. It is invoked by the following fragment of iAPX-86 assembly language code:

```
mov   al,32   ; move an ASCII 32 (space)
              ; Into the <AL> register
mov   al,dl   ; in <DL> for DOS call
mov   ah,2    ; destination in standard output
int   21h     ; execute the DOS library call
```

The HAM_API library can be developed in a similar way. A suitable, unused interrupt in the DOS architecture would have to be chosen. To allow portability across machines, this value should be set by a *SET* command at boot time. For example, the AX register pair would provide 256 different API families, each with 256 different functions. The API TSR would be a dispatcher that loads and invokes specific APIs as required, as configured by *SET* commands.

Where Do We Go from Here?

To ensure that this approach is suitable for Amateur Radio, discussion is needed. Comments and observations are needed from application developers, product developers and amateurs on the feasibility of this approach. Once agreement on its viability is reached, ARRL-sponsored working groups should be created to develop the respective API descriptions. These working groups need not meet physically to develop the standard, but can use the Internet and Amateur Radio for communications.

Once the API is completed, volunteers would develop reference implementations for use by application developers. If these are successful, the working group develops conformance-testing criteria to certify that API implementations meet the standard. The ARRL then publishes the standard.

After publication, the working group convenes to maintain the standard on a regular basis. As APIs are implemented, lessons will be learned and improvements made in the functions and descriptions.

Lawrence G. Dobranski, VA3LGD, has a BS (with honors) in Engineering-Physics from Dalhousie University in Halifax, Nova Scotia, and a MS (Engineering) in Physics from Queen's University in Kingston, Ontario. He is presently a Senior Consultant with the EXOCOM Group of Companies in Ottawa, Ontario, specializing in Information Technology Security. Lawrence has been involved in the standards development of APIs for Information Technology Security Services. Lawrence's interests lie in the technical side of ham radio. He operates mainly HF mobile, with some dreams of serious contesting. □□

A Pair of 3CX800s for 6 Meters

Do you want a little power for the magic band? This small box will do the trick!

By Dick Hanson, K5AND

Editor's Note: This article also appears in The Proceedings of the 1998 Southeastern VHF Society Conference, *published by the ARRL.*

It's that time of year when some of us begin thinking about the F2 layer, VHF DXpeditions and other fun stuff relating to 6 meters—especially with the sunspot cycle on its way back up. While it is true that you can have a lot of fun on this band with 10 W, more is most often better when you're trying to make a contact 12,000 miles away as the band is fading, fading, fading....

As most of you know, there aren't many commercial amplifiers being manufactured today for 50 MHz. Since you never know how many more sun-spot cycles you have, I decided to build a legal-limit unit to suit my needs. Having had good luck with 3CX800s in other amplifier projects, and because these tubes are available at hamfests and from dealers,[1] they are my tube of choice.

Design Features

- Two-part construction: RF deck separate from high-voltage supply
- Small package: 6×13×12 inches (HWD)
- T-network input presents good match to transceiver
- Pi-L output network for good harmonic suppression
- Pressurized anode compartment for efficient cooling
- Transistor-keyed PTT line works with +12 V or ground to transmit
- Adjustable time delay on PTT line to allow for cathode warm-up
- Grid-trip circuit to protect grids
- PTT inhibited without B+ present

Circuit Details and Mechanical Construction

Not being an engineer, I depend on books, articles and technical friends for guidance. I also fall back on 43 years of building experience: memories of what doesn't work. One nifty new design tool—at least for me—is software from Chuck Reichert, KD9JQ,[2] which *really* streamlines the design process for triode amplifiers. Before I got this package, called TAP2, my projects involved *lots* of trial and error. Now things are much more predictable, and time frames for completing projects are shortened. Life is good!

Notice below the values for the input network Q, plate load impedance and output Q. Also notice that the combined output capacitance of the tubes —about 12 pF—has been subtracted from the total plate capacitance required. Tables 1, 2 and 3 (from the software report generator) show the design parameters for various anode voltages. As you can see from the reports, the plate load changes some with lower anode voltages.

After selecting your operating parameters, you can begin the component-selection process and plan the chassis configuration.

[1]Notes appear on page 26.

7540 Williamsburg Dr
Cumming, GA 30041
k5and@prestige.net
Tel 770-844-7002 (eves)
fax 770-889-8297

Circuit Details

Refer to Figs 1 and 2. A coaxial relay (K4) at the input maintains low SWR in the bypass mode. A tuned **T** input network matches transceiver outputs from 50 to 27 Ω. A vacuum plate-tuning capacitor is used in this design. Air variables are also fine, but you need one that has *little* minimum capacitance with a respectable working-voltage rating. A vacuum output relay (K5) provides QSK operation, although an open-frame relay with the proper voltage and current ratings is okay if you don't desire QSK.

Mechanical Construction

Figs 3, 4, 5, 6, 7 and 8 show some construction details. Some readers will no doubt recognize the chassis assem-

Table 1
SW.I.F.T. Enterprises©1989
Triode Amplifier Program Version 1.2
For Grounded Grid Operation
Courtesy of KD9JQ
(2) AB2 Biased 3CX800A7 @ 50.0 MHz
Rated for Forced Air

DC Plate Voltage (VP)*	=	2800.0	V
Max Plate Voltage	=	2800.0	V
Peak Plate Swing	=	2550.0	V
Min Plate Voltage	=	250.0	V
Plate Current Peak	=	3.137	A
Plate Current dc	=	1.056	A
Grid Current dc	=	0.056	A
Cath Current Peak	=	3.398	A
Design Plate RL	=	1625.6	Ω
RL for Matching	=	1654.6	Ω
Plate Diss (PD)*	=	956.9	W
Grid Diss (PG)*	=	3.0	W
Cathode Bias	=	10.8	V
Peak Grid Voltage	=	34.7	V
Zin at Cathode	=	26.8	Ω
Pin Drive (PEP)	=	38.6	W
Po @ Plate (PEP)	=	2000.0	W
Po to Load (PEP)	=	2035.7	W
DC Power Input	=	2956.9	W
Efficiency	=	67.6	%
Power Gain	=	17.2	dB
Cath to Grid Cap	=	52.00	pF
Plate to Grid Cap	=	12.20	pF
Conduction Angle	=	190.4	Deg

* Maximum allowable VP = 2500; PD = 1600 W; PG = 8 W

T Input Network

RS	50.0	Ω
L2	0.540	µH
C1	42.688	pF
L1	0.458	µH
RFC	0.195	µH
Zin	24.1 –*j* 11.7	Ω
QL	5.0	
F0	50.00	MHz

Pi-L Output Network

RP	1654.6	Ω
RFC	3.176	µH
C1	14.075	pF
L1	0.614	µH
C2A + C2B	78. 889	pF
RX	285.1	Ω
L2	0.345	µH
RL	50.0	Ω
QL	12.0	

Table 2
SW.I.F.T. Enterprises©1989
Triode Amplifier Program Version 1.2
For Grounded Grid Operation
Courtesy of KD9JQ
(2) AB2 Biased 3CX800A7 @ 50.0 MHz
Rated for Forced Air

DC Plate Voltage (VP)*	=	**2400.0**	V
Max Plate Voltage	=	2400.0	V
Peak Plate Swing	=	2150.0	V
Min Plate Voltage	=	250.0	V
Plate Current Peak	=	3.721	A
Plate Current dc	=	1.253	A
Grid Current dc	=	0.072	A
Cath Current Peak	=	4.043	A
Design Plate RL	=	1155.6	Ω
RL for Matching	=	**1181.6**	Ω
Plate Diss (PD)*	=	1006.0	W
Grid Diss (PG)*	=	3.9	W
Cathode Bias	=	8.3	V
Peak Grid Voltage	=	40.0	V
Zin @ Cathode	=	**23.9**	Ω
Pin Drive (PEP)	=	48.8	W
Po @ Plate (PEP)	=	2000.0	W
Po to Load (PEP)	=	2044.9	W
DC Power Input	=	3006.0	W
Efficiency	=	66.5	%
Power Gain	=	16.2	dB
Cath to Grid Cap	=	52.00	pF
Plate to Grid Cap	=	12.20	pF
Conduction Angle	=	190.4	Deg

*Maximum allowable (VP) = 2500; (PD) = 1600 W; (PG) = 8 W

T *Input Network*

RS	50.0	Ω
L2	0.514	µH
C1	45.831	pF
L1	0.409	µH
RFC	0.195	µH
Zin	22.0 –*j* 9.3	Ω
QL	5.0	

Pi-L Output Network

RP	1181.6	Ω
RFC	2.268	µH
C1	24.595	pF
L1	0.450	µH
C2A + C2B	96.531	pF
RX	240.4	Ω
L2	0.311	µH
RL	50.0	Ω
QL	12.0	

blies as modified Down East[3] components. I have also had excellent results using chassis made by Charles Byers, K3IWK.[4]

I personally favor smaller, rather than larger, chassis. So the almost-ready-to-go units from Down East are attractive. The only problem with chassis this small is where to put all the control stuff, the filament transformer, etc. Honestly, it would be easier with slightly more room.

Table 3
SW.I.F.T. Enterprises ©1989
Triode Amplifier Program Version 1.2
For Grounded Grid Operation
Courtesy of KD9JQ
(2) AB2 Biased 3CX800A7 @ 50.0 MHz
Rated for Forced Air

DC Plate Voltage (VP)*	=	2250.0	V
Max Plate Voltage	=	2250.0	V
Peak Plate Swing	=	2000.0	V
Min Plate Voltage	=	250.0	V
Plate Current Peak	=	3.000	A
Plate Current dc	=	1.010	A
Grid Current dc	=	0.055	A
Cath Current Peak	=	3.250	A
Design Plate RL	=	1333.3	Ω
RL for Matching	=	1361.1	Ω
Plate Diss (PD)*	=	772.1	W
Grid Diss (PG)*	=	2.6	W
Cathode Bias	=	8.1	V
Peak Grid Voltage	=	33.5	V
Zin @ Cathode	=	25.6	Ω
Pin Drive (PEP)	=	33.8	W
PO @ Plate (PEP)	=	1500.0	W
PO to Load (PEP)	=	1531.2	W
DC Power Input	=	2272.1	W
Efficiency	=	66.0	%
Power Gain	=	16.6	dB
Cath to Grid Cap	=	52.00	pF
Plate to Grid Cap	=	12.20	pF
Conduction Angle	=	190.4	Deg

*Maximum allowable (VP) = 2500; (PD) = 1600 W; (PG) = 8 W

T ***Input Network***

RS	50.0	Ω
L2	0.530	µH
C1	43.848	pF
L1	0.438	µH
RFC	0.195	µH
Zin	23.3 –*j*10.7	Ω
QL	5.0	

Pi-L Output Network

RP	1361.1	Ω
RFC	2.613	µH
C1	19.742	pF
L1	0.512	µH
C2A + C2B	88.627	pF
RX	258.2	Ω
L2	0.325	µH
RL	50.0	Ω
QL	12.0	

Amplifier Control Circuit

The control board is the combination of several circuits that I have used or seen in other projects. The board is now commercially available from FAR Circuits,[5] and it provides a neat, compact layout (2×4 inches). This circuitry is designed to satisfy the various switching requirements of popular triodes used in modern amplifiers, while providing protection against their worst enemies: loss of anode voltage, insufficient warm-up time and excessive grid current.

Control Logic Operation

After you switch the power on, nothing else happens until time-delay relay K1 closes *and* high voltage is present. The TR sequence can't begin unless both of these conditions are met. This is good, because you don't want to apply drive until the cathode has reached its operating temperature and B+ is present. If you apply even a small amount of drive to a triode without B+ present, the grid will draw *lots* of current, which could ruin the tube instantly—not a pleasant thought!

So, when you switch on the ac power, it starts the following sequence: The blower starts. The time-delay relay begins its cycle. The green **AC ON** LED illuminates. The B+ and PTT lines are held open for a preset delay time (four minutes for example). Then the delay relay, K1, closes, which applies 24 V dc to the various control transistors, and the **AMP READY** LED illuminates. The other set of K1 contacts makes 120 V ac available to remotely control the high-voltage power supply.

If anode voltage is present, Q1 turns on when its base voltage reaches 0.7 V. R6 sets the turn-on voltage; *lowering* the resistance of R6 *raises* the voltage necessary to turn on Q1. With the value shown in the schematic, the turn-on point is roughly 1400 V.

Q3 is one of two TR switching transistors. Grounding its base causes relays K3 through K5 to switch. Maximum current drawn from the transceiver's TR circuit is in the 7 to 10 mA range. Q4, the other TR switching transistor, conducts when a nominal +12 V is applied to its base circuit. This, likewise, causes the TR relays to switch as above. So both +12 V (active high) and grounding (active low) keying outputs can control the amplifier. Regardless of whether Q3 or Q4 is chosen to switch the amplifier, the **TRANSMIT** LED will light and the TR relays will switch.

Grid over-current protection is provided by Q5 and K2. R15 allows the trip current to be set in the range of 40 to 120 mA, which should cover most tube combinations. When the preset trip current is reached, Q5 conducts closing and latching K2, the over-current relay. One set of the DPDT contacts locks the relay on; the other set opens the +24-V line to all relays so that the amplifier is taken off line. A **TRIP** LED also illuminates to indicate the over-current condition. Depressing normally closed switch, S3, resets the trip circuit.

Alignment

One way I have used to adjust the trip value without having to apply B+ or drive is as follows: Apply +5 V through a 1-kΩ, current-limiting pot to the junction of M1 and M2. Start with R15 at maximum resistance. Gradually decrease R15—whilst keeping an eye on the grid current—until the circuit trips at the value you have selected. You will most likely need to reset the circuit several times (at least) while you're fooling around. Mine is set to trip at 80 mA. During actual operation, you may notice that the LED begins to flicker as grid peaks approach the trip value.

Hint: Watch the LED for an indication of too much drive, too little loading or both.

A handy tip for testing all of the above circuitry without B+ present is to temporarily connect +5 V—through a 1-kΩ resistor—to the junction of R5 and R6. This provides the necessary base voltage to turn on Q1, enabling the rest of the circuitry for testing purposes without high voltage lurking around.

The input and output circuits can be "fiddled" into range in several ways. You can use a conventional grid-dip meter—or you can use the new MFJ-259 do-almost-everything box. I will describe how to use the '259 to tune both the input and output circuits. This is done with the tubes plugged in, but with *no voltages applied*.

To tune the input, simply tack a 27-Ω carbon resistor from the cathode strap to ground; connect the '259 to the input. Then adjust the capacitor and both coils for near zero reflected power at 50.1 or 50.2 MHz. Be sure to remove the resistor when you're done!

To tune the output, tack a 1.6-kΩ resistor from the plate strap to ground, connect the '259 to the output, and start adjusting things until you achieve minimum SWR. The coil dimensions can be calculated from the standard formula:

$$L = \frac{a^2 n^2}{9a + 10b} \qquad \text{(Eq 1)}$$

or

$$n = \frac{\sqrt{L(18a + 40b)}}{a} \qquad \text{(Eq 2)}$$

where a equals the coil radius, n equals the number of turns and b equals the length of the coil. (Remember that $n \times wire\ diameter \le b$!—*Ed.*) The L4 coil shouldn't need adjustment, but L3 will probably require some. It would be nice if the coil resonated with the plate capacitor near maximum capacitance, so that you can easily move up the band by simply adjusting the plate-tuning vacuum capacitor toward minimum capacitance. This would also indicate that the Q is pretty close to the calculated value. Remove the plate-tuning resistor after your adjustments are complete.

Having built one or two amplifiers before, I confess a certain amount of wide-eyed wonder (and pleasure) at seeing an amplifier work right off the bat with only the minor adjustments I've described. At 6 meters yet!

Power-Supply Recommendations

You can use anode voltages ranging from 2200 to 2800 V. Just beware that the plate load impedance changes with anode voltage. For example, this amplifier is designed for 2800 V with a plate-load impedance of about 1600 Ω. If you change the plate voltage to say, 2400 V, the plate load will change to 1181 Ω, and the output-tank values will change accordingly. (Refer to Table 2.) You might want to consider using the provided remote ac control to switch the high-voltage power supply from the amplifier. It's very handy and one less thing to remember when you're in a hurry.

Parting Thoughts

This amplifier works just like an HF amplifier, so the tune-up procedure is very conventional. For 1500 W output, the drive power will be in the neighborhood of 45 W. Grid current will be about 40 to 50 mA, and plate current

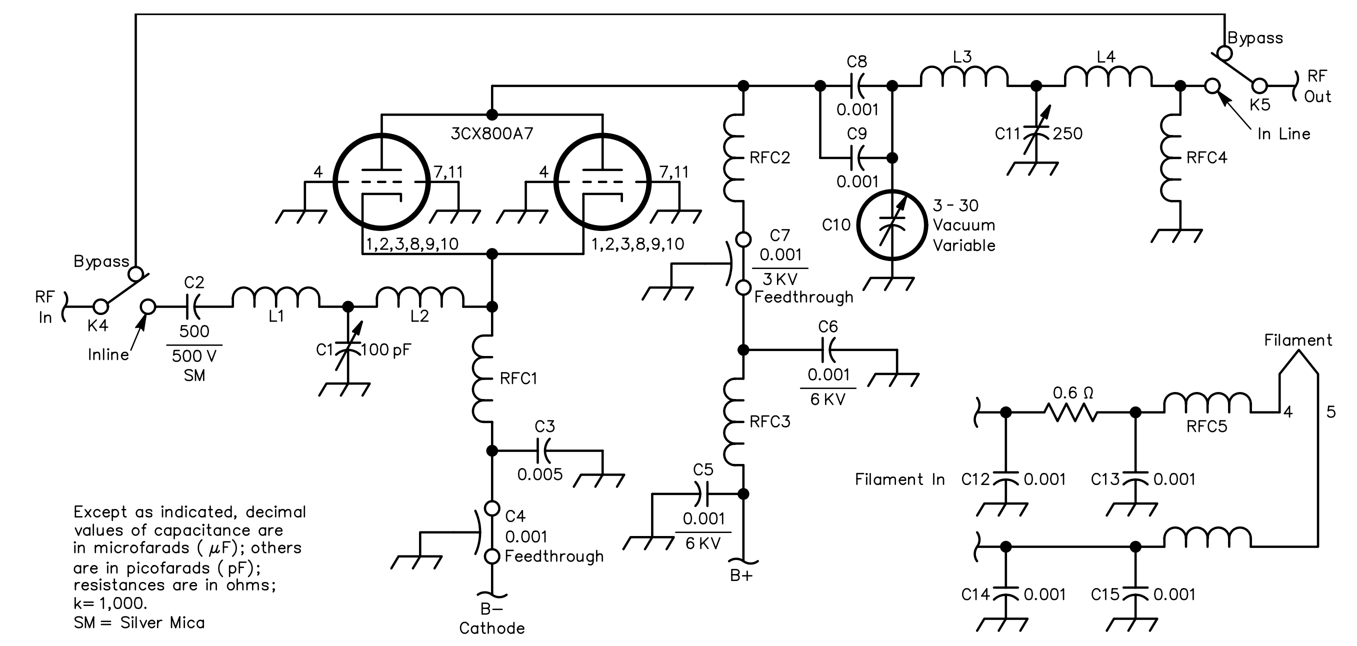


Fig 1—Schematic of the 6-meter amplifier RF deck. B+ is 2200 to 2800 V at 2 A. The coils of K4 and K5 are shown in Fig 2. Use $^1/_4$ W, 5%-tolerance carbon-composition or metal-film resistors unless otherwise specified. Equivalent parts may be substituted.

L1—10 turns, #16 AWG, $^1/_2$-inch diameter, 1-inch long
L2—7 turns, #16 AWG, $^1/_2$-inch diameter, $^5/_8$-inch long
L3—3$^3/_4$ turns, $^1/_4$-inch-diameter tubing, 1$^3/_4$-inch diameter, 3 inches long
L4—5 turns, #6 AWG, 1-inch diameter, 1$^3/_4$-inch long
RFC1, 3, 4—7 μH
RFC2—40 turns, #22 AWG on $^1/_2$-inch diameter Teflon rod, winding is 1$^3/_8$-inch long; alternative, 34 turns of #18 AWG on a $^3/_4$-inch-diameter fiberglass threaded form.
RFC5—19 bifilar turns, #18 AWG on $^5/_8$-inch-diameter form

will be about 1.2 A. These conditions represent an efficiency of 65% or so, which is fairly respectable. I always use an ICE #426 low-pass filter on the output of my high-power, 50-MHz amplifiers. I will be happy to answer questions by telephone, fax or e-mail.

Special thanks to Chuck Reichert, Steve Kostro at Down East Microwave and Pat Stein at Command Technologies for software, weird parts, ideas and general support. Projects like this don't happen without help and encouragement; I was fortunate to have had both.

Notes

[1]Tubes: There are so many used tubes available now at hamfests that you should be able to buy "guaranteed" pulls for $150 to $225 a piece. One source is Allen Bond, who can be reached at **mgs@avana.net** or at 770-973-6251.

[2]*Triode* software: Download from KD9JQ's site **www.imaxx.net/~kd9jq**.

[3]Chassis: Down East Microwave, Steve Kostro; tel 908-996-3584; **www.down-eastmicrowave.com**.

[4]Chassis: Byers Chassis Kits, Charles Byers; tel 717-292-4901 6-9 pm EST; **k3iwk@juno.com**.

[5]PC Board: FAR Circuits; tel 847-836-9148; **www.cl.ais.net/farcir**.

[6]Obviously, some of the parts used are from flea markets, junk box, etc but the schematic captions show additional sources.

Dick was first licensed as WN0UUU in 1954. He completed a BA at the University of Texas in 1966. Dick worked with the medical electronics division of Hewlett-Packard for 18 years. He was the medical manager for the southern region when he left to purchase Southern Staircase in 1983. Since then, he has been the President and CEO of Southern Staircase.

Dick enjoys building amplifiers and antennas. He also likes operating on 6 and 2 meters from rare countries during F2 propagation. He has been to most of the Caribbean islands for DXpeditions.

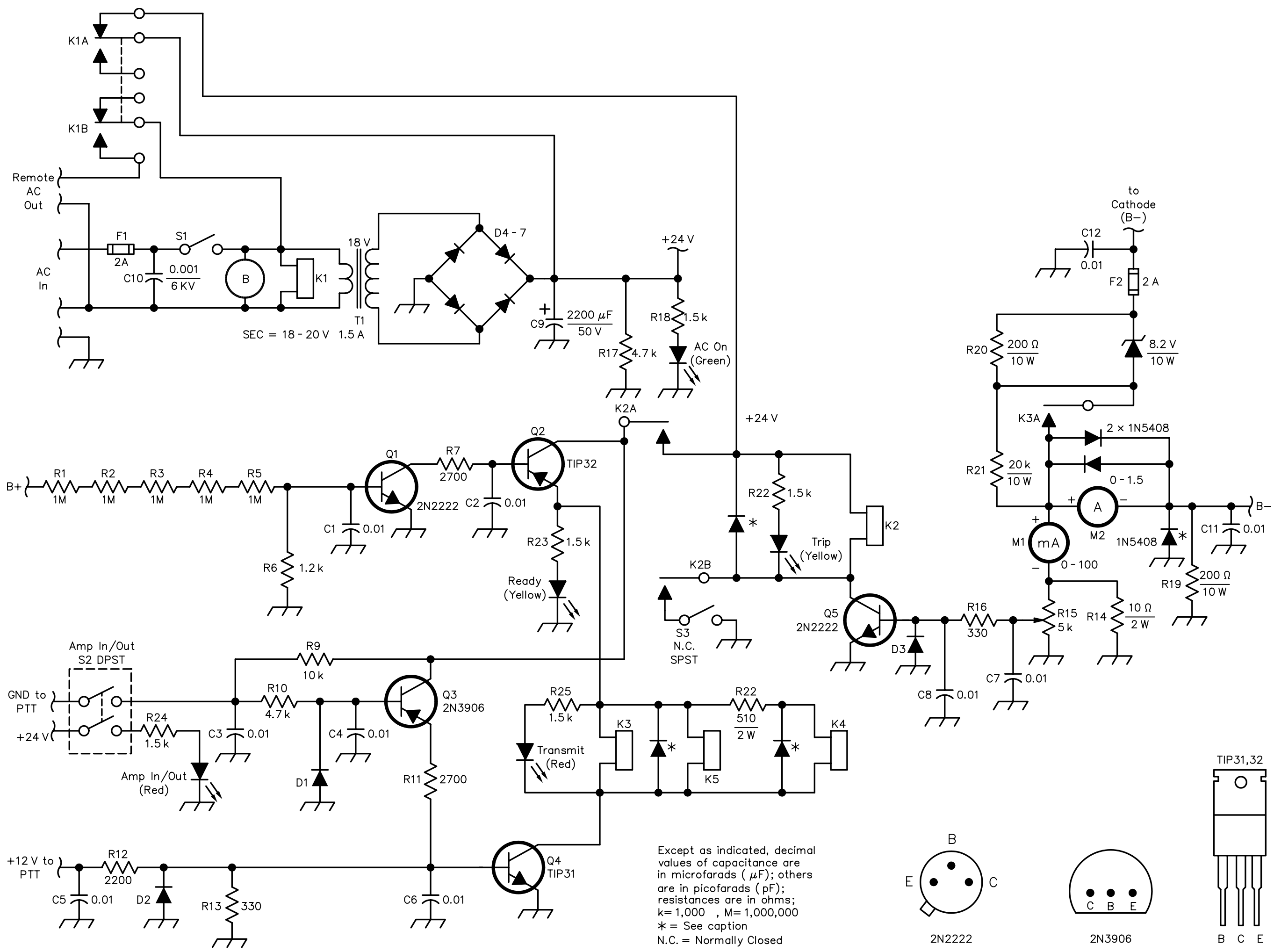


Fig 2—Schematic of the 6-meter amplifier control circuits. The contacts of K4 and K5 are shown in Fig 1. All unmarked diodes are 1N4001s. LEDs do not have part numbers, but they are labeled with their function and color. Use 1/4 W, 5%-tolerance carbon-composition or metal-film resistors unless otherwise specified. Equivalent parts may be substituted.

B—Dayton 4C440 blower
K1—120 V ac coil, time-delay (0 to 5 minute), such as Omron H3Y-2
K2, K3—24 V dc, DPDT units such as PB KHAE-17D12-24, OEG SRET-203DP or various Radio Shack equivalents
K4—24 V dc operated coaxial relay, available from Allen Bond (see Note 1).
K5—24 V dc operated vacuum relay, available from Allen Bond (see Note 1).
T1—120 V primary, 18 V, 1.5 A secondary, such as Hammond K166K18
T2—120 V primary, 13.5 V, 3 A secondary

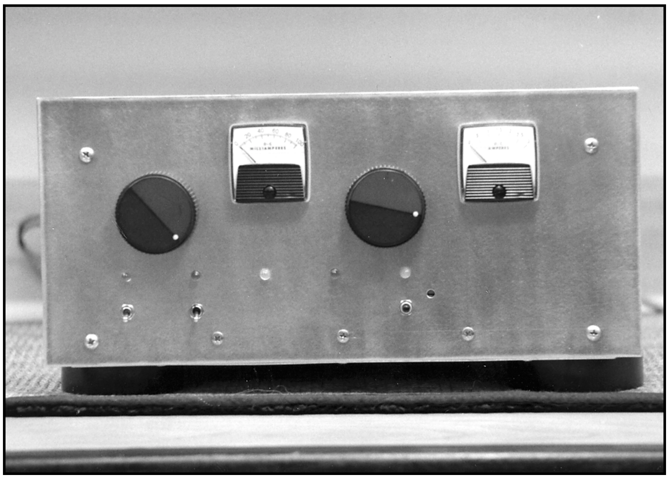


Fig 3—A view of the amplifier front panel (without labels or paint). There are three horizontal rows of controls and indicators: Top row (left to right): PLATE LOAD, GRID CURRENT **meter,** PLATE TUNE **and** PLATE CURRENT **meter. Second row (LEDs, left to right):** AC ON, AMP IN/OUT, READY, TRANSMIT **and** GRID TRIP. **Third row (switches, left to right):** POWER, AMP IN/OUT **and** RESET.

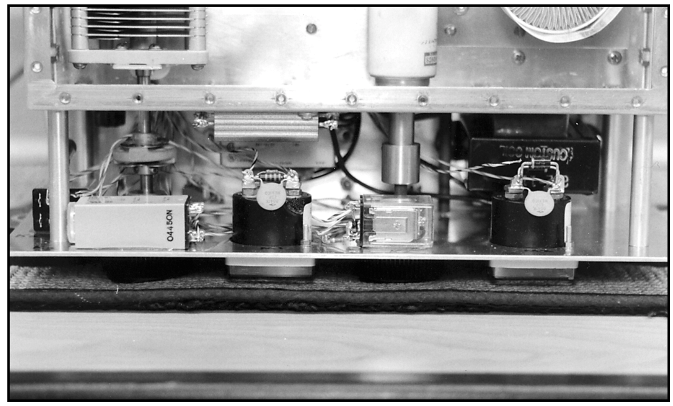


Fig 4—The front subpanel. The filament transformer is at the lower right. K1, K2 and K3 are visible. The filament resistor (0.6 Ω) is visible above the left-hand meter.

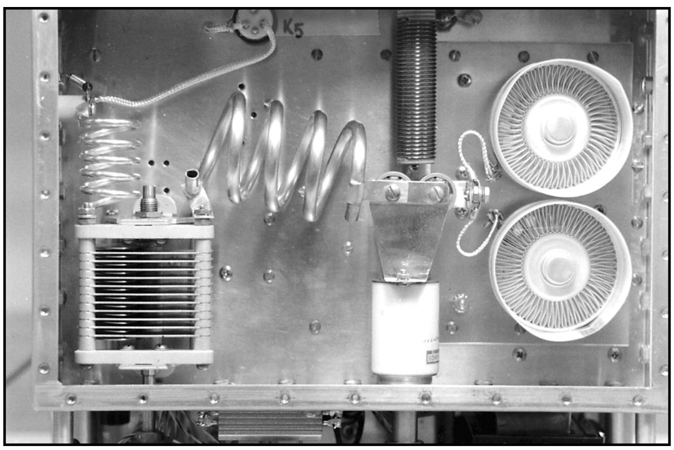


Fig 5—A close-up of the RF compartment. Starting at the lower left, we see C1. The white can is C11 (vacuum variable), which connects to C8 and C9 (type 858 blocking capacitors) via a metal plate. The large (silver plated) coils are L3 and L4. The darker, wire coil is RFC2 (on a threaded fiberglass form). The output vacuum relay (top left center) is labeled K5 on the chassis. The two 3CX800A7s are at right, with homebrew Teflon chimneys.

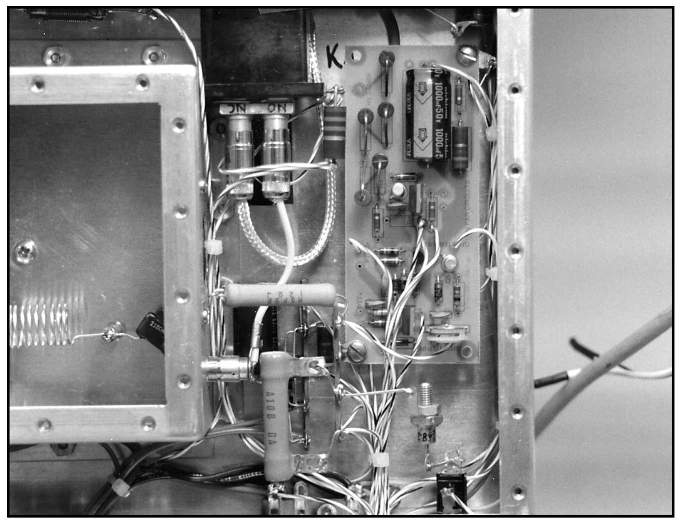


Fig 6—Control-circuits PC board. The coaxial input relay is on the chassis at the upper left of the PC board.

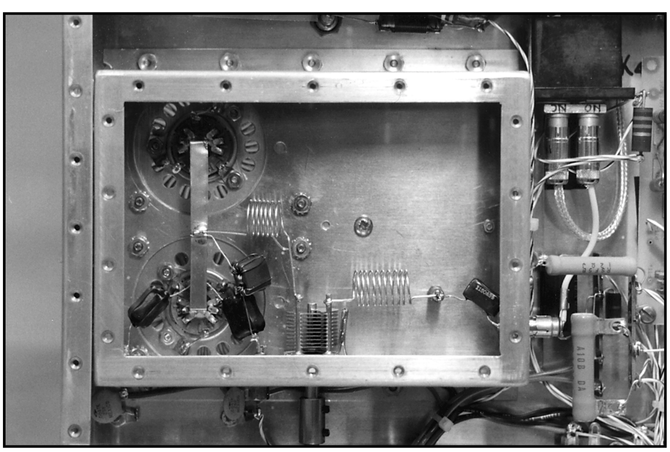


Fig 7—The cathode compartment shows RFC1 and RFC5. Rings of holes around the sockets allow air past the tube bases to the cooling fins.

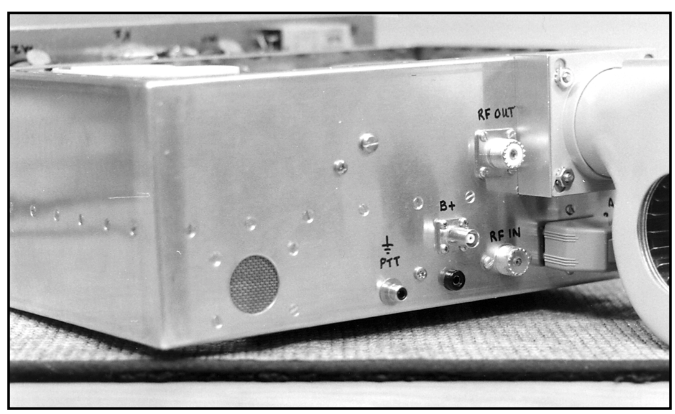


Fig 8—The amplifier rear panel. The unmarked banana jack is for the B–. RF OUT **is a Teflon-insulated SO-239. A multipin connector for the ac, remote ac to high-voltage supply and blower connection is visible below the blower. Notice the blower plenum extension that permits the blower to clear the multipin plug. You can eliminate the plenum by orienting the blower 90° or 180° (ccw) from the position shown.** □□

A Compact Mobile Tuner

Build this handy antenna tuner and tailor it to your mobile station. A band-switched inductance favors easy tuning while on the road.

By Patrick Wintheiser, W0OPW

The Compact Mobile Tuner (CMT) is a very small, band-switched antenna tuner that is intended for use with short vertical antennas. It can extend the bandwidth of a whip antenna so that you need not leave the car to make antenna adjustments on most HF bands. You can also build it in a miniaturized form for QRP use. Not only that, it is much more efficient than you would expect in such a small package. This small tuner easily fits on the dash of a car and takes little space in a backpack. It is band switched so there is little hunting for a correct match. Just switch to the desired band and adjust two variable capacitors for the lowest SWR. Notice, however, that this tuner will *not* load a fence post in the desert! It is designed only for extending the useful bandwidth of resonant antennas.

The need for this tuner arose last winter, when I went mobiling in the Midwest. My commercially made T-network tuner is too bulky to be useful on the dashboard, and the band-changing adjustments became tedious. In addition, I was never sure whether I was loading the antenna or the tuner. In fact, I had my back-up lights flashing Morse code somewhere in North Dakota!

I decided to see if it was possible to build a compact, band-switched tuner. The original design is by Ulrich Rhode,[1] and I adapted from it freely. It is a low-pass L-network tuner for the 80 through 10-meter bands. A typical L-network design would use a roller inductor, which is monstrous for my needs. Through some trial and error, I replaced the roller inductor with a small toroid, and by adding a variable capacitor in series with the output, I was able to extend the tuning range. The space and weight savings are well worth the effort.

This tuner is capable of matching almost any vertical antenna that has been resonated at one point in the band. It extends the tuning range of short verticals beyond resonance, but only so far. Using 75 meters as a worst-case example, I found that my mobile operating bandwidth is now increased by a factor of about four, before the losses get excessive. In my case, that's a tuning range of 160 kHz, compared to 40 kHz without the tuner. On higher frequencies, I can safely tune across an entire band. The tuning range of longer verticals is much greater.

Please notice that 160 kHz is the *useful* bandwidth with this tuner. This—or any—tuner may show a 1:1 SWR over a much greater bandwidth, where tuner losses are much greater than the antenna's radiation resistance. Because of this, the radiated power does drop off when attempting to load highly reactive antennas beyond what Mother Nature allows. I've determined this with a very noisy sodium-vapor lamp above my town-house patio, located about 12 feet from the mobile antenna. It spews radiation across a very wide bandwidth, and I can use my S meter (on receive) to monitor the power loss away from resonance. It is amazing to see the receive

12251 SE 59th St, #106
Bellevue, WA 98006

[1]Notes appear on page 30.

signal drop off 3 to 4 S-units even though the SWR bridge indicates a 1:1 SWR. (My commercially made **T**-network tuner is even *worse* under these conditions. A recent *QEX*[2] article shows why. It's interesting to note that a **T**-network design can have seven times the loss of a simple **L**-network.

Construction Notes

Begin by winding the 4:1 toroid transformer, T1, which steps the 50-Ω transceiver output down to the typical impedance of a short vertical. (If your short HF vertical is a nice match for 50 Ω all by itself, it is very lossy!—*Ed.*) I used 12 turns of RG-174 coax on an FT-114-43 ferrite toroid. The center conductor is the primary and the shield is the secondary. (The two windings are series connected to form an autotransformer, as shown in Fig 1.—*Ed.*)

After winding the core, tie the two ends of the coax together with some string and liberally coat the winding with glue to hold the turns in place. Let the glue dry overnight. Next, glue T1 on the back of the cabinet somewhere out of the way.

Wind tuning inductor L1 with 22 turns of #14 enameled copper wire. Glue L1 onto the bottom of the case under the band switch, S1. If you do not plan to use this tuner on 75 meters, you can eliminate L1 and use the extra space to increase the size of L2, if desired.

L2 is an air-core inductor ($8\frac{1}{2}$ turns of #14 wire, about $1\frac{1}{2}$ inches long). This winding adds about 1.5 µH of inductance for the 40 to 10-meter range. Wind L2 using a plastic 35-mm film canister as a form: Tape some waxed paper around the form and add binding screws at each end of the form to hold the wires in place. Wind the turns on the form and then wrap $\frac{1}{16}$-inch nylon braid between the wire turns to evenly space them. Use 5-Minute epoxy to secure the turns in place and let the coil dry overnight. Remove the binding screws in the morning; the coil should easily slide from the form. Remove the waxed paper, but you can leave the nylon braid in place, as part of the coil.

For the enclosure, I used a $7\frac{1}{2} \times 4\frac{1}{4} \times 2\frac{3}{8}$-inch plastic box from Ocean State Electronics.[3] If you use a metal enclosure, be sure to well insulate the inductors and capacitors from the case: The shaft of C2 is "hot" and needs to be well insulated.

Examine the enclosure and plan where to drill the holes for the coax and connector (the back of the box), the variable capacitors and band switch (the front). I did not use a connector for the line to the transceiver, choosing instead to hard-wire the coax directly into the circuit. This eliminates a connector from the box, and by mounting the input and output connections directly behind C1, keeps the ground connections very short. A DPDT tuner-bypass switch is optional, but it's handy when operating near resonance, or when you just want to see what difference the tuner makes. C1 and C2 are heavy-duty 20 to 400 pF units from Fair Radio Sales[4] that will safely handle several hundred watts. Broadcast-receiver tuning capacitors will also work fine, and they're much smaller. The physical size of C1 and C2 determines the size of the enclosure. Unfortunately, it seems that variable capacitors only come in two sizes; too big or too small! If you operate at higher power levels, use larger toroids for T1 and L1. S1 can be anything from 1P6T to 1P12T, depending on how many taps you need.

Adjustments

If you use the recommended components, you should be able to install my tap points on L2 without any modifications. Otherwise, attach an SWR bridge to the tuner's input side and a resonant antenna to the output. With low power (less than 5 W), apply RF to the tuner. Watch the SWR and—with C1 and C2 both $\frac{1}{2}$ to $\frac{2}{3}$ meshed—try different tap positions on L1 or L2. When a minimum occurs on the SWR meter, solder a jumper from the S1 to the tap. I need taps for 75, 40, 30, 20 and 17 to 10 meters on the switch. Although 40 and 30 meters can be tuned on one switch position, it may be better to provide separate bandswitch positions. The very last switch position covers 17 through 10 meters. This tuner could be dedicated to an individual antenna without switching, but I've found that it matches every vertical I could test. An additional surprise is that it also tunes dipoles just fine!

Test each band by loading the antenna every 50 kHz. If the antenna will not load sufficiently across each band, then move the tap up or down the coil until the loading is satisfactory. Note that this **L**-network design will not tune antennas very far from resonance. If C2 is fully open without a match, the antenna is too reactive for this tuner. In that case, a **T**-network design might be better. However, beware of losses in any tuner when the operating frequency is far removed from antenna resonance!

An Optional Output-Current Meter

A simple output-current (proportional to the square root of power) meter is a useful addition to this tuner—or to any tuner for that matter. (See Fig 2.) I added one to the CMT by routing the wire from the output to S2B through a T-37-2 powdered-iron toroid to form a one-loop primary. The secondary is about 20 turns of #28 enameled wire. For the sensing cir-

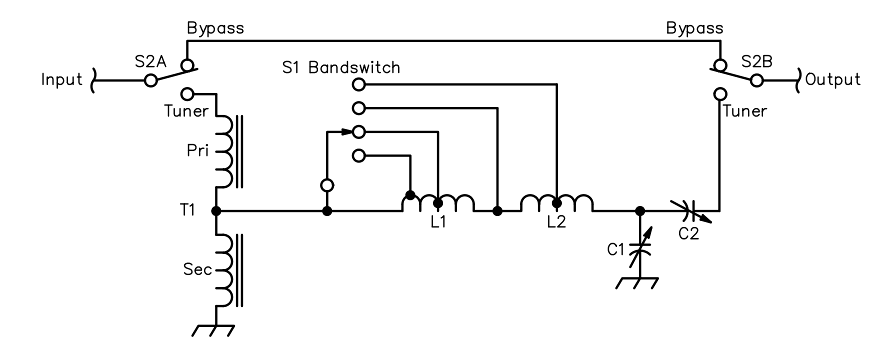


Fig 1—A schematic of the Compact Mobile Tuner.

C1, C2—20-400 pF variable capacitor
L1—22 turns of #14 enameled copper wire (for 75 meters) on a T-130-6, or larger, powdered-iron toroid core.
L2—$8\frac{1}{2}$ turns of #14 bare copper wire on a $1\frac{1}{4}$-inch OD form, with taps at turn 1 (bottom, for 40 meters), turn 4 (30 meters), turn 6 (20 meters), turn $8\frac{1}{2}$ (top, for 17-10 meters). See text.
T1—4:1 balun using 12 turns of RG-174 coax on a FT-114-43, or larger, ferrite toroid. Connect the output end of the inner conductor to the input end of the outer conductor.
S1—1P6T rotary switch.

cuit, add a germanium diode (1N34, 1N60, etc) and a 0.01 mF bypass capacitor to the output of the secondary and connect it to a 100 kΩ potentiometer and a 0 to 200 mA meter outside the tuner case. (See Fig 2. Any sensitive meter up to about 1 mA should work just fine.)

I calibrated the meter face by setting the pot for a full-scale (100%) reading with a 100 W transceiver output feeding a resonant antenna with the tuner bypassed. The other percentage marks were established with lesser power settings. This simple circuit and meter indicates relative output power. In essence, if output current *increases* while adjusting the tuner (*with constant input power*), the antenna match is improving. Since I have band switched my tuner and I know that I'm capable of good transceiver match, I've eliminated the SWR bridge from the system and simply tune for maximum output current.

If the output current *decreases* while adjusting the tuner (*with constant input power*), the tuner is delivering less power to the load and dissipating the difference! In a series of articles for *QST*, Maxwell[5] described the reasons for using antenna input current as a means of monitoring the conjugate match.

I ran a comparison test against my commercial **T**-network tuner. After adjusting the CMT for minimum SWR on 20 meters, I set the pot for a full-scale reading. I then switched the tuner out of the line, connected the **T**-network tuner and adjusted it for minimum SWR. By observing the output current, I could see that the **T**-network tuner output current was only 50 to 75%, relative to the CMT output, depending on the switch and capacitor settings. This dramatically proves to me the advantages of monitoring output current when using any tuner. It also shows that a simple **L**-network is about as efficient as a tuner can be. Using the current meter, I also discovered that any tuner *loses* power in attempts to improve the match of an antenna with an SWR of 1.5:1 or less!

On the Road

I took my new tuner on the road to the Midwest, and it performed extremely well. I deliberately omitted the SWR bridge and tuned for a maximum on the output-current meter.

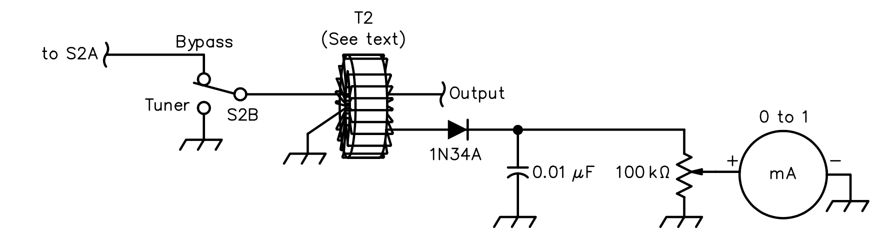


Fig 2—A schematic of the optional output-current meter. T2 is a transformer with a one-turn primary formed by the wire from S2 to the output cable, which passes through the center of a T-37-2 powdered-iron toroid core. The secondary is 20 turns of #28 enameled wire.

The tuner easily works on 75 through 17 meters and takes up little dashboard real estate. I received excellent signal reports, including 579 from ON7GB, while I was operating CW—from South Dakota! I was able to give him his 50th state for 30-meter WAS, which was a thrill for me.

Notes

[1] Ulrich Rhode, KA2WEU, "Some Ideas on Antenna Couplers," *QST*, Dec 1974, pp 27-30.

[2] Kevin Schmidt, W9CF, "Estimating T-Network Losses on 80 and 160 Meters," *QEX* Jul 1996, pp 16-20.

[3] Ocean State Electronics, PO Box 1458, 6 Industrial Dr, Westerly, RI 02891; tel 401-596-3080, 800-866-6626, fax 401-596-3590; www.oselectronics.com.

[4] Fair Radio Sales Co Inc, PO Box 1105, 1016 E Eureka St, Lima, OH 45804; tel 419-227-6573, 419-223-2196, fax 419-227-1313; fairadio@wcoil.com; URL http://www2.wcoil.com/~fairadio/.

[5] M. Walter Maxwell, W2DU, "Another Look at Reflections, Part 7," *QST*, Aug 1976, pp 15-20.

First licensed as KN0OIW in 1957, in St Peter, Minnesota (recently devastated by the tornado of March 29, 1998). He has been employed as a computer systems engineer for 32 years. Pat only recently became actively involved in Amateur Radio again after being off the air for 20 years. He is a member of The Mike and Key ARC in Renton, Washington, and also operates from his summer lake cabin in Meeker County, Minnesota.

Experiments with Phase-Noise Measurement

Do you want to measure the phase noise of your latest oscillator, but don't own a sophisticated spectrum analyzer? Learn how to measure oscillator phase noise with equipment you can afford!

By Jos F. M. van der List, PA0JOZ

This article, was previously published in Dutch. It appears in the May and June 1992 issues of Electron *magazine, which is published by VERON the Dutch IARU Society.—Ed.*

Phase Noise

An ideal oscillator produces a clean and unmodulated signal. The output is a pure sine wave without changes in amplitude or in the time between the zero-crossings. If we observe such a signal on an ideal spectrum analyzer, we see a single spectral line. (See Fig 1.)

Real oscillators are different, however. Both the amplitude and the time between the zero-crossings are prone to noisy variations. (See Fig 2.) On the same ideal spectrum analyzer, we see the main spectral line accompanied by sidebands caused by amplitude and phase modulation, the results of the noisy variations. (See Fig 3.) I must stress here that the spectrum analyzer is not adding to the sidebands caused by the amplitude modulation and the phase modulation. Both types of modulation cause sidebands to appear on either side of the carrier, and the spectrum analyzer can only measure the amplitude of the results.

The local oscillator (LO) is used in a receiver to convert signals from one frequency to another, as represented in Fig 4. It's evident that the signals

Fluitekruid 20
2201 SM Noordwijk
Netherlands
jvdrlist@gironet.nl

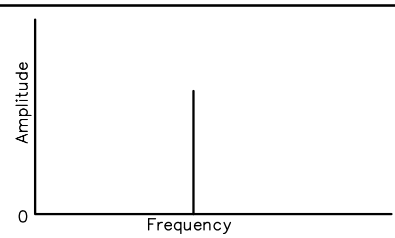


Fig 1—The one single spectral line of an ideal oscillator.

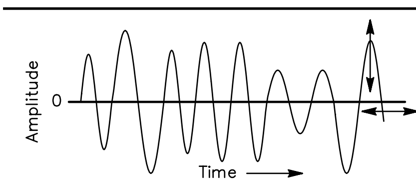


Fig 2—Exaggerated amplitude and phase noise of a non-ideal oscillator.

converted to the IF frequency may be noisy, although the input signals are clean. The noise on the IF signals is caused by the sideband noise of the LO. To show that this is true, we have to make a little excursion into modulation theory. Before doing so, let's have a few words about amplitude variations on the LO signal.

In a well-designed mixer, the amplitude of the LO signal is much stronger than the strongest signal to be converted. Also, the mixer gain—or loss—will not depend very much on the oscillator amplitude. (See Fig 5.) The conversion loss of a normal, double-balanced diode mixer is shown as a function of the oscillator power. Note that the conversion loss does not change very much with the oscillator power above a certain threshold. From this, it is demonstrated that amplitude variations of the LO do not play an important role in a well-designed mixer and LO combination. The level of the mixed signal is directly proportional to that of the input signal, and does not depend on the level of the LO.

Let's return to Fig 4. In this figure, we can see the adverse effects of what is called "reciprocal mixing." This is where the noise sidebands of the LO mix with a strong, nearby signal to produce noise in the IF. The noise degrades the signal-to-noise ratio (S/N) of any desired signal there. Even if the IF bandpass filter is an ideal "brick-wall" type, the receiver selectivity suffers as a result of the reciprocal mixing effect. This is the reason for the attention paid to reciprocal mixing and to phase-noise reduction on oscillator signals.

The adverse effects of phase noise play their roles from the transmitting end, also. (See Fig 6.) In this example, the receiver is considered to be ideal, but one of the received signals has noise sidebands. When this signal is converted to the IF, we have the same problem as in Fig 4. Now the S/N of the weaker signal is degraded by the noise on the nearby, strong, undesired signal. When our ideal receiver converts all signals linearly to the IF, the noise on the adjacent-channel signal causes interference to the weaker, desired signal.

Frequency and Phase Modulation

To gain a better understanding of the deleterious effects caused by reciprocal mixing, it is useful to have a look at some of the principles of modulation, especially phase and frequency modulation. Although we will ultimately deal with noise as a modulation source, it is easier to begin by using sinusoidal signals. This makes the figures easier to understand and makes the calculations easier.

The output signal of an oscillator can be written as:

$$U(t)=A(S)\cos\left[2\pi t\, f(S)+\phi(S)\right] \qquad \text{(Eq 1)}$$

where A is the amplitude of the oscillator. If A were a function of the modulating signal, we would be talking about amplitude modulation. This type of modulation is not discussed further here.

f is the frequency of the oscillator. When f is a function of the modulating signal, we are talking about frequency modulation. The symbol ϕ is the relative phase of the oscillator signal. From the fact that both f and ϕ determine the argument of the cosine function, it is obvious that frequency and phase modulation are close kin. Both types of modulation are forms of "angle modulation." In this format, the angle is made a function of the modulating signal.

For example: Suppose that the phase portion of the argument $\phi(S)=2000\,\pi\, t$, then:

$$U(t)=A\cos\left[2\pi f t+2000\pi t\right] \qquad \text{(Eq 2)}$$

and this can be rewritten as:

$$U(t)=A\cos\left[2\pi t(f+1000)\right] \qquad \text{(Eq 3)}$$

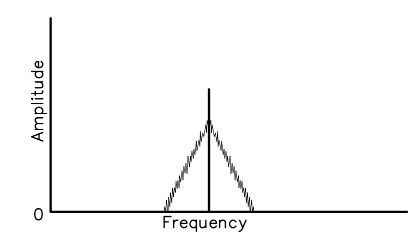


Fig 3—Spectrum of a non-ideal oscillator.

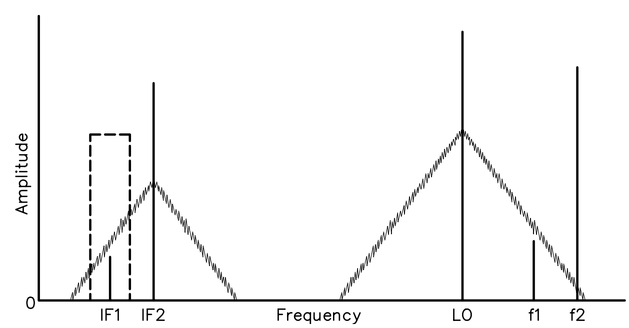


Fig 4—Mixing in a receiver with a non-ideal oscillator.

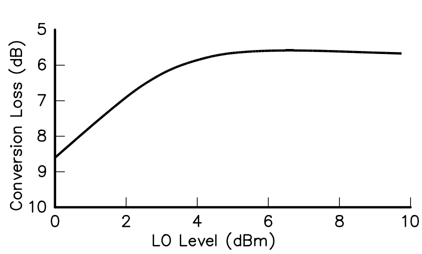


Fig 5—Conversion loss of a double-balanced diode mixer as a function of LO power.

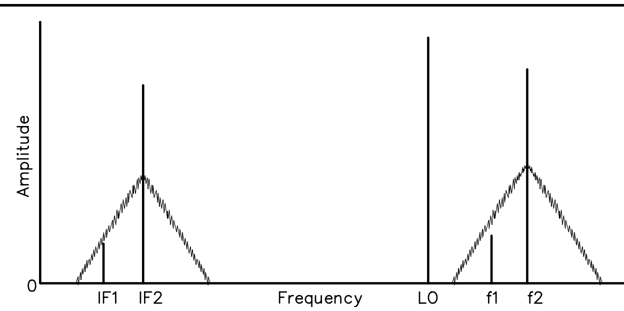


Fig 6—The effect of reciprocal mixing: A weak input signal is masked by the phase noise from a nearby strong signal.

The frequency of the oscillator has become 1000 Hz higher. In other words, a constant increase in the rate of change of phase is the same as a higher frequency; a constant decrease in the rate of change of phase is the same as a lower frequency. In mathematical terms, we say that frequency is the derivative of phase.

If we deal with sinusoidal information, it can be shown mathematically (for angle modulation) that the modulated signal can be represented as:

$$U(t) = A\cos\left[2\pi f\, t + m \sin\left(2\pi f_m t\right)\right] \quad \text{(Eq 4)}$$

with f_m the frequency of the modulating signal, and m the so-called modulation index. For FM, the modulation index is:

$$m = \left(\frac{\textit{peak deviation}}{\textit{modulation frequency}}\right) \quad \text{(Eq 5)}$$

For PM, the modulation index is a constant, and represents the peak phase deviation, in radians. The peak frequency deviation of a phase-modulated signal is $f_m \times m$. For PM, and using a constant-level modulation source, the peak deviation is proportional to the modulating frequency.

For single-tone modulation, there is no way to tell the difference between frequency and phase modulation. As an example, I have recorded a spectrum analyzer plot of an angle-modulated signal. (See Fig 7.) The carrier frequency is 100 MHz and the modulating frequency is 2 kHz, as is evident from the spacing of the sideband components. The modulation index is four, so the frequency deviation is 8 kHz and the phase deviation is four radians.

Note that this signal has many sideband components. Theoretically, an angle-modulated signal has an infinite number of sidebands. In practice, happily, the strength of the components far from the carrier diminishes very rapidly, and they disappear in the noise. On the plot, the level of corresponding components at either side of the carrier is equal, and the spacing of the components is equal to the modulating frequency. The amplitude of each of the sidebands can be calculated using so-called Bessel functions.

In Fig 8, these functions are plotted graphically. When discussing phase noise, we normally deal with phase-modulated carriers having low modulation indices, *ie*, below 0.5. For this low index, the amplitude of the carrier and the first few sidebands can be estimated with sufficient accuracy from the following formulas: $J_0 = 1$, which means that the amplitude of the carrier is virtually equal to the amplitude of the unmodulated carrier; $J_1 = m / 2$, which means that the amplitude of the first pair of sidebands is well below the amplitude of the carrier (–12 dB for m = 0.5); $J_2 = m^2 / 8$ (–30 dB for m = 0.5); $J_3 = m^2 / 48$, etc.

As an example, I have plotted the spectrum of an angle-modulated signal at 100 MHz as Fig 9. The modulation frequency is 2 kHz again, but the frequency deviation is only 400 Hz. So, the modulation index is 0.2, and you can verify that the above estimates of the sideband strengths are good. It is also evident that the occupied bandwidth of the signal is very much less than that of the signal in Fig 7. One can say that for very low modulation indices, only the first sideband pair is significant. In this example, the second pair is more than 20 dB below the first pair.

Application to Sideband Noise

After this somewhat theoretical part, let's go back to our subject. If we

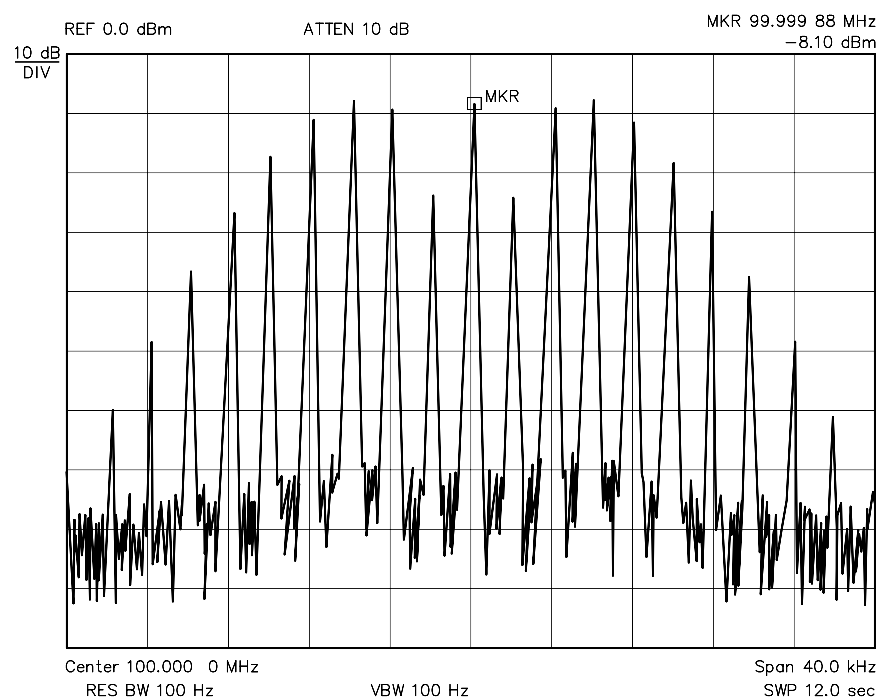


Fig 7—Spectrum of an FM-modulated signal at 100 MHz; the modulation frequency is 2 kHz, deviation is 8 kHz, modulation index is 4. The reference level of the spectrum analyzer is adjusted to the amplitude of the unmodulated carrier.

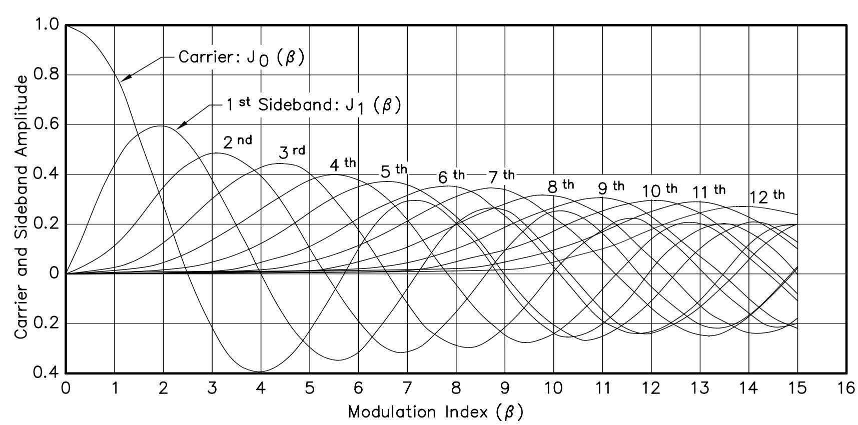


Fig 8—Graphic representation of the Bessel functions that are used to calculate the amplitude of the carrier and the sidebands of an angle-modulated signal.

regard the signal in Fig 9 as the input signal of a receiver and mix it with an ideal LO, we get a signal at the difference frequency—10.7 MHz, for example—that retains the modulation characteristics. (See Figs 10 and 11.) The relationship between the carrier and the sideband components depends on the modulation, which is not changed by the mixing process. The mixer in this example is performing a frequency translation: IF = RF – LO. The result is that, at the IF, one cannot tell whether any FM or PM was caused by the input signal, or by the LO.

As I mentioned, these kinds of examples are more easily understood if we use only sinusoidal modulation. The same processes, of course, are also valid for more complex modulation sources, such as noise. One thing has to be clear when working with noise: The way we specify the difference between the carrier and the sideband levels. With sinusoidal modulation, it is easy to measure the difference between the carrier and sidebands, since they are each found at discrete frequencies. With noise, it is somewhat more complicated. The amount of noise that we measure depends on the measurement bandwidth—the power is directly proportional to the bandwidth.

Suppose we measure an oscillator signal at 100 MHz with a carrier level of +10 dBm, or 10 mW. In a measurement bandwidth of 1 kHz, we measure a noise level of –50 dBm at 10 kHz from the carrier. This is 60 dB down. If we repeat the measurement with a bandwidth of 100 Hz, the result would be a noise power of –60 dBm, and that is 70 dB below the carrier. Therefore, we usually normalize the results of noise power measurements to a bandwidth of 1 Hz. In this example, the normalized noise-power level would be –80 dBm, and that is 90 dB below the carrier level, thus –90 dBc/Hz. Of course, it is not necessary to measure in a 1 Hz bandwidth, since the normalization is easily performed.

Now for one more example in which all the previous information is used. Suppose we have a receiver with an IF of 9 MHz, a LO at 41 MHz, a preamplifier at 50 MHz with a gain of 20 dB and a mixer with a conversion loss of 6 dB. (See Fig 12.) The LO has phase noise of –100 dBc/Hz at 20 kHz from the carrier. There are two signals at the input of the receiver, one at 50.000 MHz with a level of –120 dBm, and the other at 50.020 MHz with a level of –50 dBm. According to the IARU S-meter standard for VHF, the first signal is approximately S4 or S5, the second signal is S9 + 43 dB. The input signal at 50.000 MHz is converted to exactly 9 MHz and it falls at the center of the IF bandwidth. After the mixer, its level is –106 dBm (–120 dBm + 20 – 6 dB). The second signal is converted to 9.020 MHz, outside the passband of the IF filter; it has a level of –36 dBm. It seems that there is no problem. The weak signal in the IF passband is received, and the strong signal outside the IF passband is rejected by the filter. The difference between the two levels is 70 dB, so it seems that any regular IF filter (with an attenuation greater than –70 dB at 20 kHz from the center) can do a fine job.

If we consider the reciprocal mixing effect, the result is quite different. At 20 kHz from the stronger signal, the noise level is –100 dBc/Hz. In a 2.5 kHz bandwidth, this is 34 dB more, or –66 dBc / 2.5 kHz. This means that the strong signal causes noise in the IF passband at a level of –36 – 66 dBm = –102 dBm—4 dB stronger than the weak signal in the passband. The weak signal is masked by the noise from the LO. From this example, it is very clear that not only the selectivity of the IF filters determines the overall selectivity, but also the phase noise of the LO(s). Both transmitters and receivers need to be considered.

Measuring Phase Noise

Now that we have shown the importance of phase noise, we can pose the

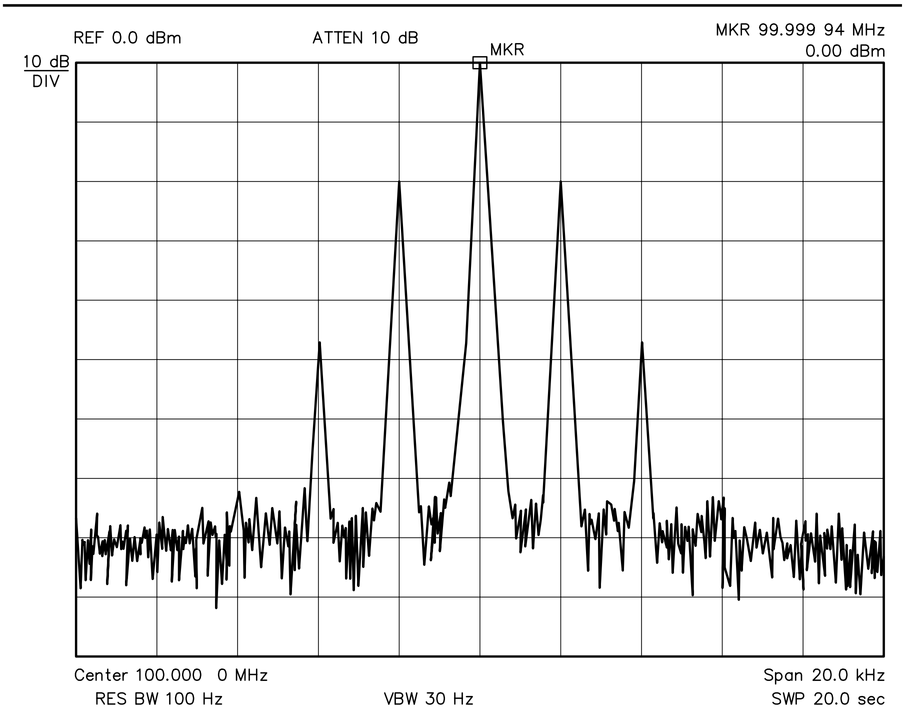


Fig 9—Spectrum of an FM-modulated signal at 100 MHz; the modulation frequency is 2 kHz, deviation is 400 Hz, modulation index is 0.2. The reference level of the spectrum analyzer is adjusted to the amplitude of the unmodulated carrier.

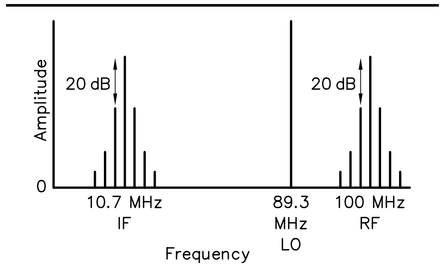


Fig 10—FM modulated signal at 100 MHz mixed to an IF of 10.7 MHz.

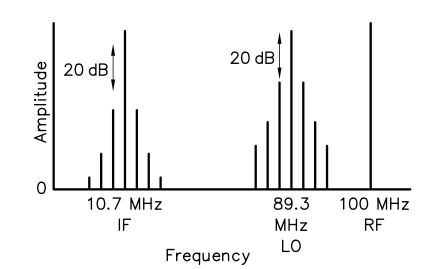


Fig 11—Unmodulated signal at 100 MHz mixed to an IF of 10.7 MHz by an FM-modulated LO.

question: How should we measure the phase noise levels of our oscillators? I present some of the possible methods below.

First, because phase noise can be regarded as unwanted frequency modulation, we could try to measure the deviation. This is not very practical, because the deviation is liable to be only a few hertz, and deviation meters are not suitable for these low levels. Furthermore, it would be difficult to get any information about the relation between the deviation and the modulating frequency.

Second, we could use a receiver or a spectrum analyzer to perform the measurement. This might be the best choice for oscillators that are not very good. Because of the phase-noise levels of the LOs used in spectrum analyzers, the sensitivity of the measurement is limited. Reciprocal mixing also occurs in spectrum analyzers! This method also measures the amplitude noise of the oscillator under test. Besides, not everyone has a spectrum analyzer at home.

Alternatively, we could use the signal under test as the LO in a receiver, and use a very clean signal from a crystal oscillator as the receiver's input signal. From the reciprocal mixing effects measured, it is possible to calculate the phase-noise level of the oscillator. This arrangement works fine, and I use it when performing selectivity measurements on commercial receivers. The method is limited, however, by the IF filter's selectivity. This is not a problem when the whole receiver is being reviewed, because the total selectivity is what matters. It does make things difficult when the receiver is used as test equipment, and we are only interested in the oscillator's performance.

There are many more measuring methods—such as those that use phase shifters and delay lines. They are not the most sensitive, although they have certain advantages. For more information about these methods, see the References listed at the end of this article. The method I describe is used in professional phase-noise measuring equipment. In fact, the test fixture I made is also a kind of receiver, with the carrier not converted to an IF but to 0 Hz. The fixture also has a sensitive phase detector.

The block diagram is presented in Fig 13. The blocks within the dotted lines are in the test jig. The components outside are external measuring and support equipment. The equipment needed includes an oscilloscope and an ac voltmeter. At the input port, a double-balanced mixer is used as a phase detector. Two signals are phase-compared: The signal to be characterized, and a reference signal from a good oscillator, preferably a crystal oscillator. The oscillators must be at the same frequency. The loop circuit ensures that the two oscillators are phase-locked in such a way that the dc output voltage of the phase detector is kept at 0 V. The switchable components in the loop provide sufficient control to establish phase lock. The goal is to maintain a low loop bandwidth. In this way, any drift and low-frequency FM of the oscillators is compensated, but the higher-frequency components of the phase noise appear at the output. In fact, we have made a direct-conversion receiver. The two sidebands of the tested oscillator are both converted to a frequency range starting just above 10 to 50 Hz and reaching to some 100 kHz. Actually, the four sidebands from *both* oscillators are converted to the audio range. (See Fig 14.)

At the phase detector's output is a low-noise audio preamplifier and some twin-T filters to suppress the hum that would otherwise overload the following filters and amplifiers. Then we have three filters/buffers in parallel. One of these is tuned to a nominal frequency of 1 kHz with a bandwidth of about 100 Hz. The next is tuned to 10 kHz with a 1-kHz bandwidth, and the last to 100 kHz with a 10-kHz bandwidth. The gains of the amplifiers after the filters are adjusted during calibration, as described below. This test fixture makes it possible to measure phase-noise levels at three discrete offsets (1, 10 and 100 kHz) from the carrier. Although this is not as nice as a plot from a spectrum analyzer, it provides sufficient information for oscillator experiments.

The full schematic diagram is shown in Fig 15. An MD108 is used as the phase detector. An SBL-1 may be substituted. The dc balance is achieved with a small resistor network from the –12 V supply. This is something that must be adjustable, or determined experimentally with every individual fixture. The double-balanced mixer is terminated with 50 Ω—for RF—by the series-connected 50-Ω resistor and the 3.3-nF capacitor. The LC low-pass filter keeps RF out of the audio circuits. The preamplifier with the BC149 (a European low-noise audio transistor)

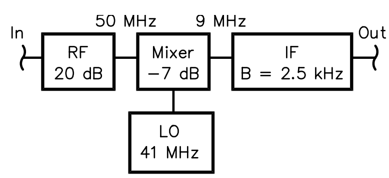


Fig 12—Block diagram of a 50-MHz receiver.

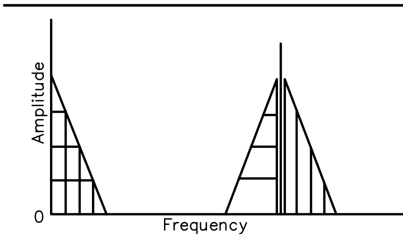


Fig 14—Both the lower and the upper sidebands are mixed to the frequency range from 0 to 100 kHz.

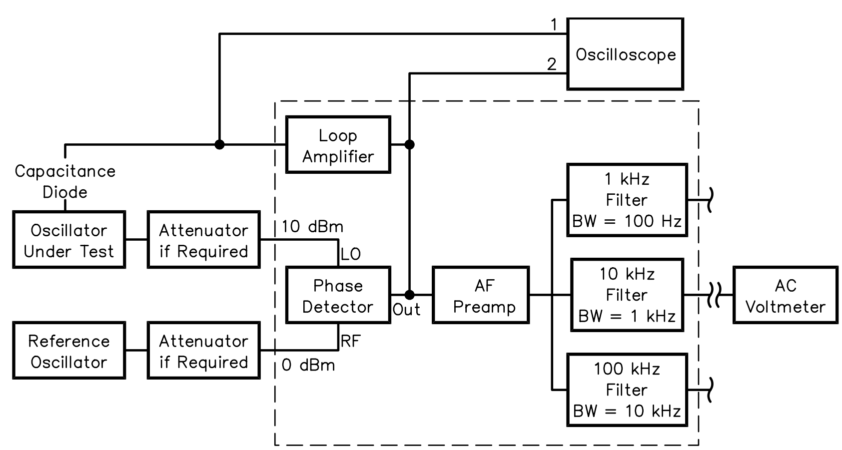


Fig 13—Block diagram of the phase-noise test setup.

and the NE5534 amplifies the phase noise and limits the frequency range somewhat. The two twin-T filters keep 50 Hz and 100 Hz away from the filters and amplifiers. They must be redesigned for 60 Hz and 120 Hz in the US. These two notch filters are important, because often in test setups, oscillators have some mains-induced residual FM. This FM can be strong enough to overload the amplifiers. On the output connector after the NE5534, one can connect a low-frequency spectrum analyzer. (A sound card in a PC can be used for this purpose.) At the time that I made this test jig in 1990, such cards were not available.

There are three filters after the preamplifier. The 1-kHz and 10-kHz filters are active, with three identical, series-connected sections. The 100-kHz filter is an inductively coupled bandpass filter. (I used two RF chokes.) The gain of each amplifier is adjusted by means of the feedback resistors between the output and the inverting input of the operational amplifiers. This is done during initial calibration.

The circuit with the three OP27 op amps is the loop amplifier and filter used to phase-lock the two oscillators. One of the oscillators must be electronically tunable over a few-kilohertz range. The tuning voltage is taken from the output of the last OP27. Switch S1 allows one to adjust the loop gain, S3 sets the loop time constant, and S2 adjusts the loop damping. I don't intend to cover basic PLL techniques in this article, and I suppose that knowledge of the required theory is available to most people trying this kind of thing. At the output of the first OP27, an oscilloscope can be connected to verify that the loop is behaving well.

The whole circuit is housed in a cast-aluminum box. The power supply is not in the box. Stray fields from the transformer would interfere with the very sensitive measurements. Further circuit-layout details and a PCB pattern are not available. The test jig was constructed using "ugly" (dead-bug) construction, and I never had time to give it a neat appearance. You must regard the schematic diagram as a starting point or as a source of ideas. Design your own circuit, and publish it [in *QEX*!—*Ed.*] if it contains nice new ideas!

Regarding the external measuring instruments: The oscilloscope must have a sensitivity of 10 mV/div and a bandwidth of at least 100 kHz. The ac voltmeter must have a best sensitivity of 1 mV full scale and a bandwidth of at least 100 kHz.

Measuring with the Test Fixture

After the setup is calibrated, measurements can be made. (I will deal with calibration later.) A measurement is performed as follows: Connect the test oscillator and the reference oscillator at the inputs. The power of the reference oscillator at the LO-port is not terribly critical. The power must be between +4 dBm and +13 dBm. The level of the test oscillator at the RF port *is* critical. Ideally, the level of this oscillator should equal the level used during calibration. Every decibel of change causes a decibel of change in the measured composite noise levels. I calibrated my setup at a level of 1 mW (0 dBm). The level at the RF port must not be very much higher than that, because the phase detector will become nonlinear above 0 dBm. If less than 0 dBm is used, the sensitivity of

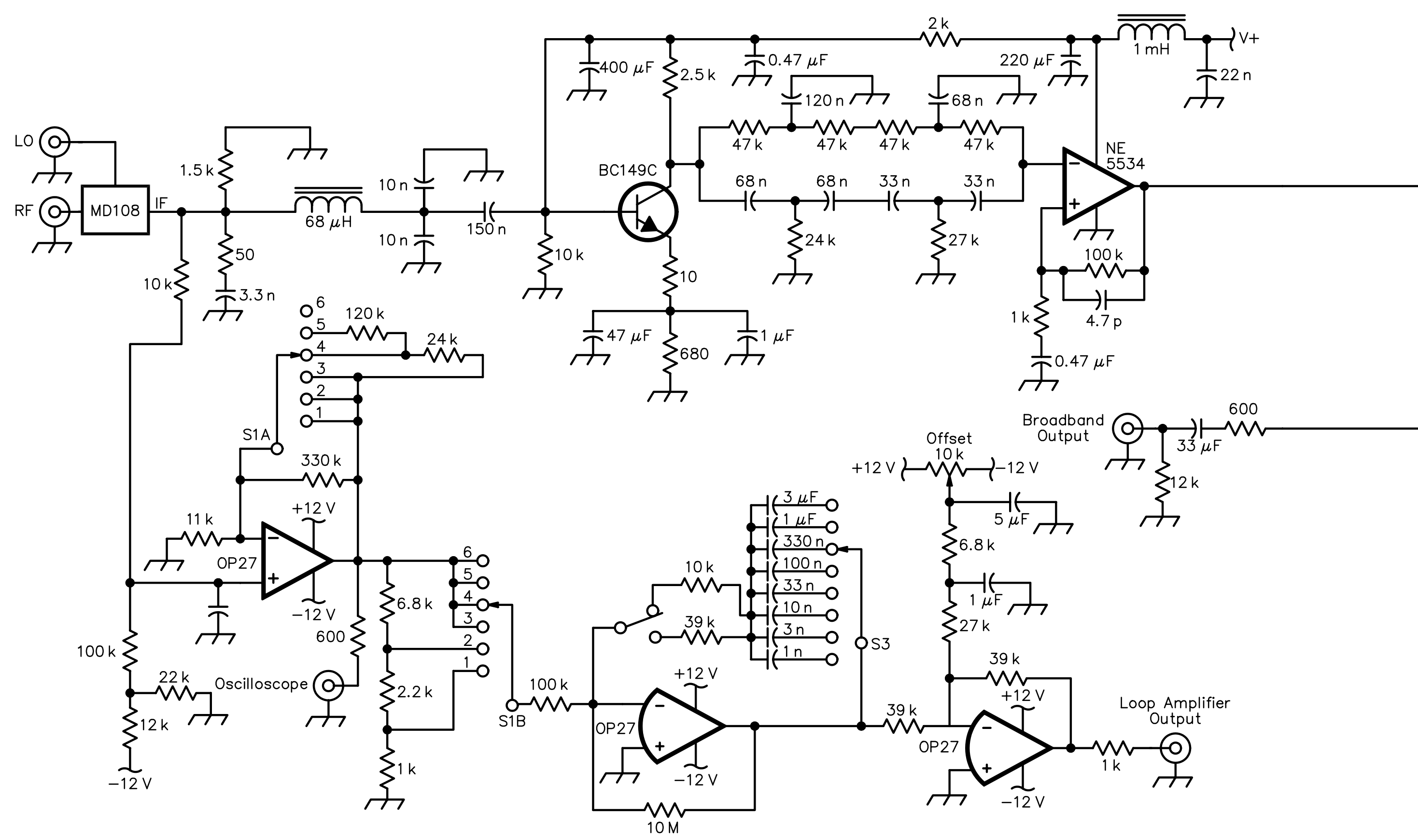


Fig 15—Schematic diagram of the phase-noise test jig.

the setup becomes progressively less. In some cases, however, it can be useful to deliberately use less input power. If the noise sidebands are very strong and overload the preamplifier in the fixture, using less power brings the circuits back in their linear-operating range. The tested oscillator's level must be set by an external attenuator. A word of caution: It is necessary that the two oscillators be well buffered. Otherwise, injection locking is possible via the phase detector, and that is an unwanted situation.

The output of the loop amplifier is connected to a Varicap circuit in one of the oscillators. The oscilloscope is connected to the monitor output, and if there is a second channel, it is connected to the output of the loop amplifier. By adjusting S1, S2 and S3, the oscillators are forced to phase lock. By adjusting potentiometer P1, the operating range of the Varicap diode is set, while the average output of the phase detector is kept at 0 V.

The ac voltmeter can be connected successively to the three outputs to measure the levels of the phase noise. The actual phase-noise levels can be calculated using factors determined during calibration.

The levels determined this way are, in fact, the sum of the composite noise levels of both oscillators. It is therefore important to use a reference oscillator—such as a crystal oscillator or a VCXO—that has much better noise characteristics than the oscillator being characterized. If that is not possible, one cannot find the actual noise levels of the tested oscillator, but one can determine that the tested oscillator is at least as good as the measured level.

Now some words about the "art" of phase noise measurement: The signal levels we are measuring are *very* low, sometimes below the microvolt level. Therefore, it is necessary to keep external influences out of the test setup. Use cables that are as short as possible. Be sure that the coaxial connectors make good contact with the cable, otherwise some of the RF currents induced on the outside of the shield may also flow on the inside, causing interference. This is a well-known electromagnetic-compatibility problem. Keep the setup away from such interference sources as TVs and PC monitors. Do not try to measure unshielded oscillators.

A less-obvious problem is acoustical noise. During some of my measurements of crystal oscillators, I noticed a correlation between variations in the measured phase-noise level at 1 kHz and the speech of a local ham talking on 70 cm! When I investigated, I found that I could easily whistle a tone near the oscillators that would produce phase modulation well above their normal noise levels. This makes you wonder about transceivers with built-in loudspeakers.

One more warning: I have noticed many times during my experiments

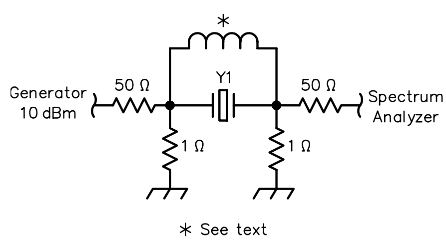
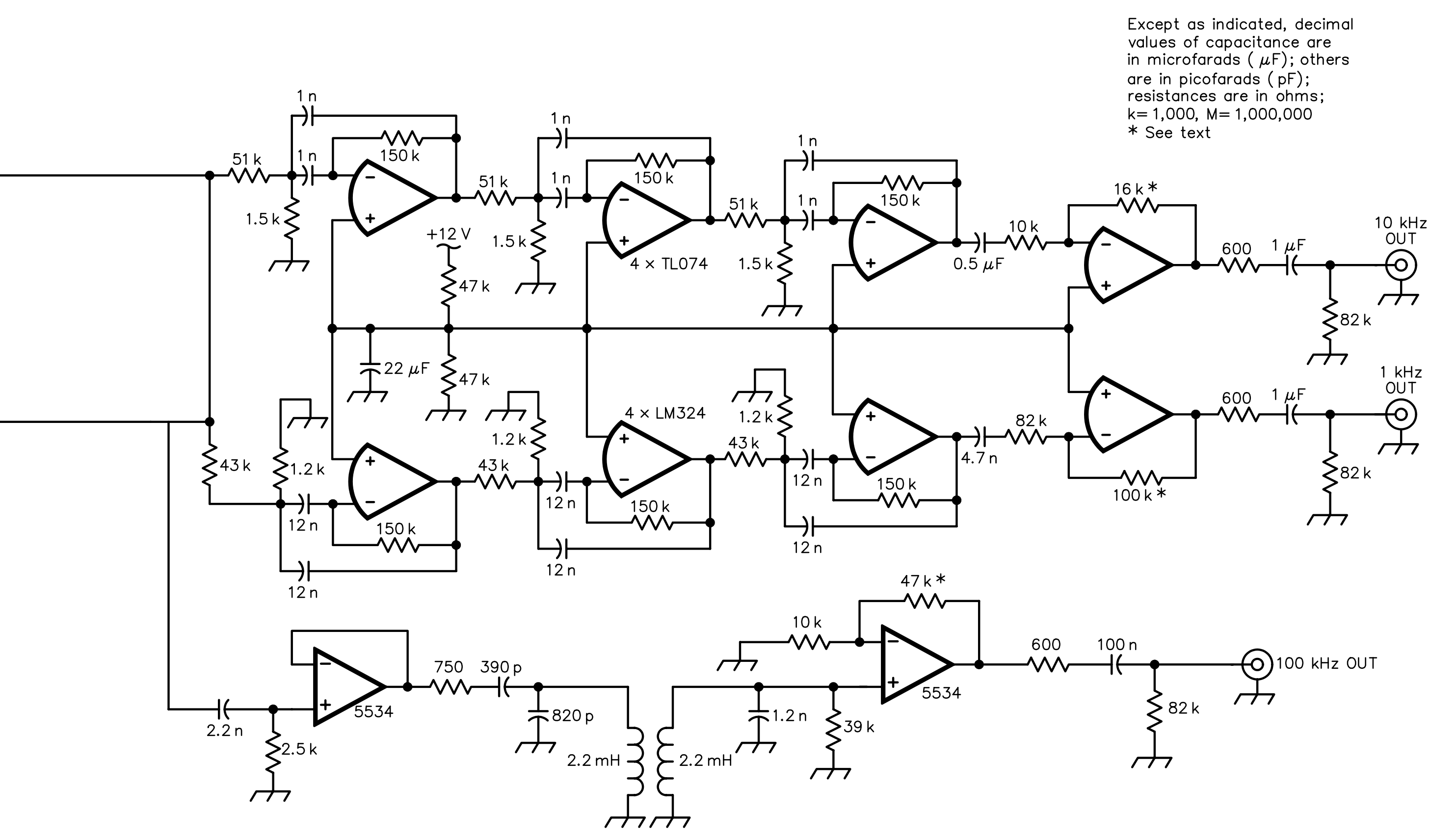


Fig 16—Measurement of the series-resonant Q of a crystal.



that one must not perform phase-noise measurements shortly after having soldered parts in an oscillator. During the first couple of hours after soldering in an oscillator circuit, the phase-noise level shows sudden outbursts, especially at 1 kHz from the carrier. My explanation for this is that minor mechanical tension caused by the temperature changes must relieve itself, and this causes minor electrical changes, which in turn cause these noise bursts. Maybe someone has a better explanation?

Calibration of the Test Fixture

Although relative measurements can be performed without calibration, it is nice to have some idea about the absolute levels. Oscillators with absolutely certain phase-noise behavior cannot be bought, as far as I know. There are two good calibration methods.

The first one uses a crystal oscillator as the reference source, and a phase-modulated signal generator or VCXO as the RF source. The latter source is modulated successively with 1 kHz, 10 kHz and 100 kHz. While observing the measurement on a spectrum analyzer, the modulation level is adjusted until the first sidebands come to –80 dBc. There will be only one pair of sidebands visible; the second pair will be below the analyzer noise level. The power of the reference source must be set to +10 dBm, the power of the RF modulated RF source to 0 dBm. Phase lock is established, then each of the three feedback resistors in the output amplifiers is adjusted to obtain exactly 1 V RMS from each of the outputs. For this first method, one needs a rather good spectrum analyzer. Some of us are quite happy to have such an instrument in our labs, but not everyone is so lucky.

The second method may be somewhat easier to perform. If we take 0 dBm as the RF input calibration level and we want –80 dBc to be the level for 1 V at the outputs, we can perform the calibration using a single –74 dBm signal at an appropriate offset frequency. Since during normal operation, both noise sidebands contribute to the total power, a 6-dB-stronger signal is used here. The calibration procedure is identical to that of the first method. With an LO signal applied at +10 dBm, and the RF signal at –74 dBm placed at 1 kHz, 10 kHz and 100 kHz from the LO frequency, the calibration resis-

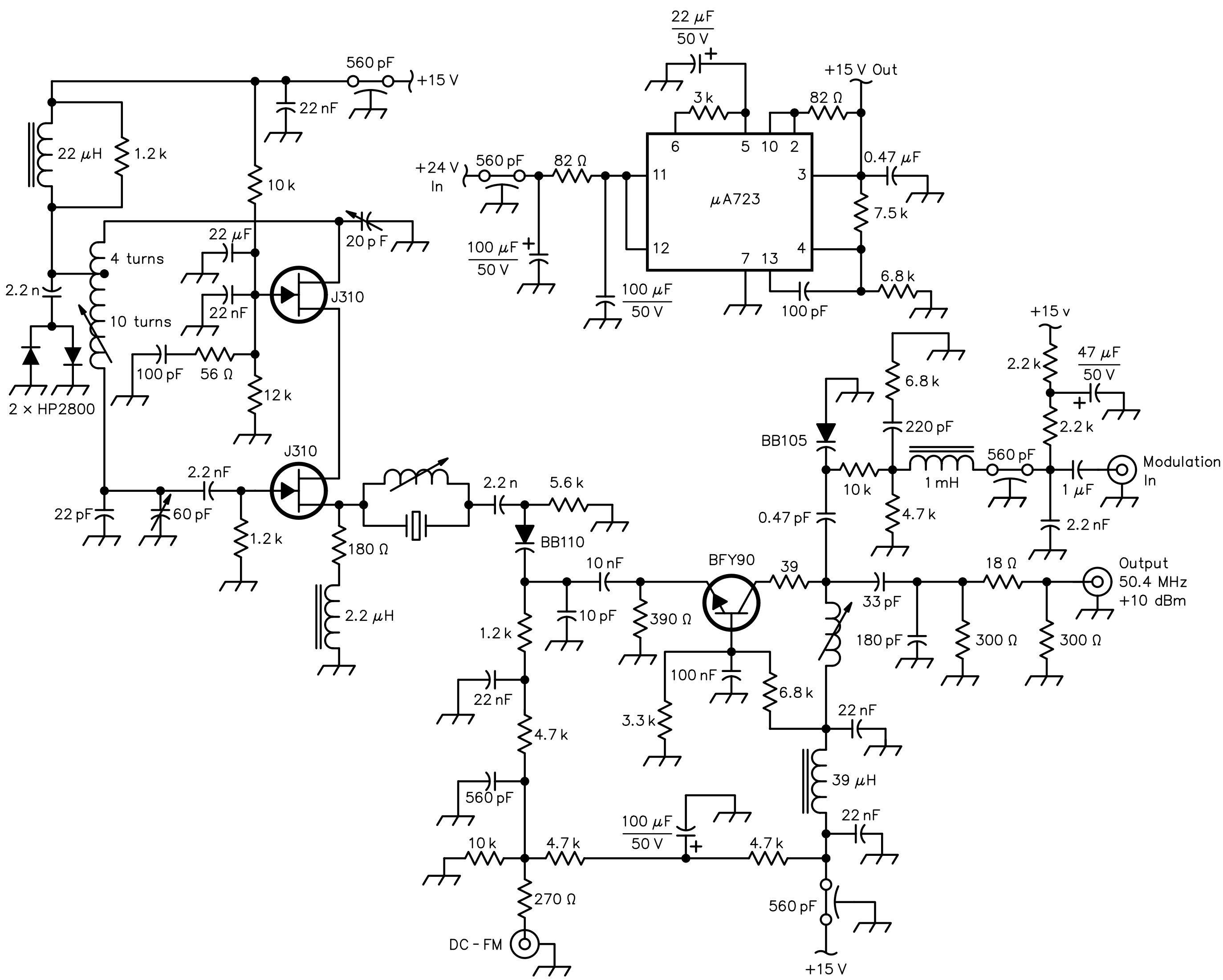


Fig 17—Schematic diagram of the 50 MHz reference crystal oscillator.

tors are adjusted for 1 V output. A signal generator with a calibrated output is probably easier to get than a spectrum analyzer.

I compared the calibration techniques, and for me, they resulted in less than 2 dB difference. In my opinion, this is adequate for amateur purposes.

If we are using method two and a signal generator, we can also easily determine the exact bandwidths of the three filters. Carefully adjust the frequency of either oscillator to both sides of the frequency where the unit was calibrated, and note the two frequencies were the 1 V output drops to 0.707 V. These are the –3 dB points, and the difference between the two is the 3-dB bandwidth of the filter. What we really need are the so-called noise bandwidths of the filters, but these can be estimated by multiplying the 3-dB bandwidths by 1.1. Having measured the bandwidths of the filters, we can calculate the correction factor for each of the filters:

$$C = 10\log(B) \qquad \text{(Eq 6)}$$

Where C is in decibels and B is in hertz. The conversion factor translates the level measured in the respective bandwidths to the normalized level, expressed in dBc/Hz. For example: if measuring a phase-noise level of 100 mV using the 1 kHz filter, and having determined that the noise bandwidth of that filter is 1100 Hz, the normalized phase noise level is:

$$-80 - 20 - 30.4 = -130.4 \text{ dBc/Hz} \qquad \text{(Eq 7)}$$

Finally, we must determine the noise floor of the whole setup. To do this, we connect a 50-Ω load resistor to the RF port and a +10 dBm LO signal to the reference port. Then we measure the voltages from each of the three filter outputs, and from these values, calculate the noise floor. In my case, it was approximately –162 dBc/Hz. If we are measuring an oscillator, and the phase-noise levels happen to be so good that they come close to the noise floor, we can measure with less accuracy using the formula:

$$L = -80 - 10\log(B) - 20\log\left(\frac{1000}{U_n}\right) + 10\log\left[\left(\frac{U_l^2}{U_n^2}\right) - 1\right] \qquad \text{(Eq 8)}$$

In this formula, to be used with measured phase noise levels less than 10 dB above the noise floor:

L = phase-noise level, in dBc/Hz
B = noise bandwidth of the filter in use
U_n = noise-floor output level, in mV
U_l = measured phase-noise level, in mV

A Reference Crystal Oscillator

The phase-noise test jig was constructed to perform measurements on LC oscillators around 50 MHz. Therefore, I needed a good reference source near this frequency. First, I determined the Qs of a batch of 50.4 MHz crystals that I had. The simple test setup is shown in Fig 16. I determined the –3 dB points of the response, and from that, calculated the series-resonant Q. The values ranged from 20,000 to 30,000. Using the crystals with the highest Qs, two crystal oscillators were built.

The schematic diagram (Fig 17) was derived from an article about VHF crystal oscillators in the German magazine *UKW Berichte*. (This is translated to English as *VHF Communications* magazine.—*Ed.*) The principle is that the lower FET works linearly and that amplitude limiting is done in the upper FET. The crystal acts as the source bypass and determines the oscillator frequency by its series resonance. If the crystal is removed from the circuit and replaced with a capacitor of, say, 1 nF, the circuit can be made to oscillate on approximately the correct frequency. If

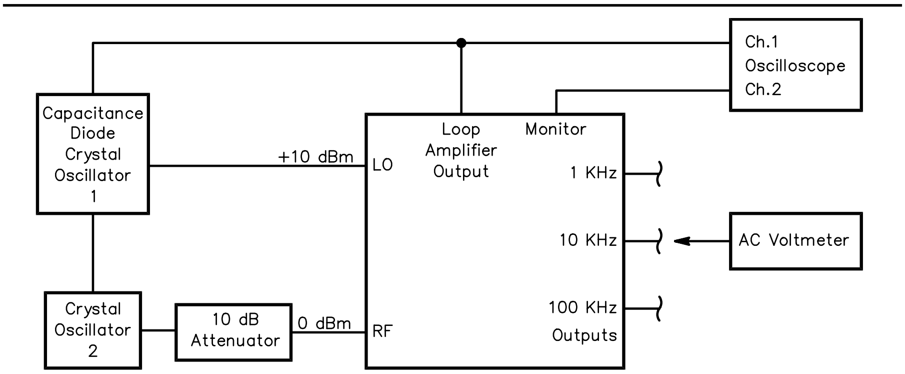


Fig 18—Block diagram of the phase-noise measurement of two crystal oscillators.

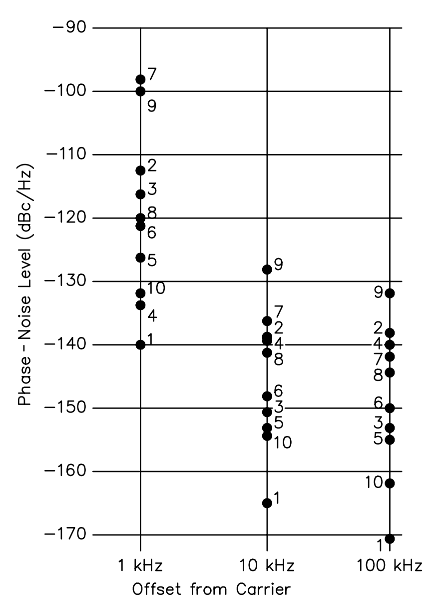


Fig 19—Comparison of the 50 MHz phase-noise levels of several oscillators and signal generators.

Table 1—Test Equipment

1. Crystal oscillator, PA0JOZ
2. ADRET 7200A
3. HP8640B
4. HP8662A
5. HP8642B
6. Rohde and Schwarz SMG
7. Rohde and Schwarz SMDU
8. HP608E
9. HP8657B
10. LC Clapp oscillator, PA0JOZ

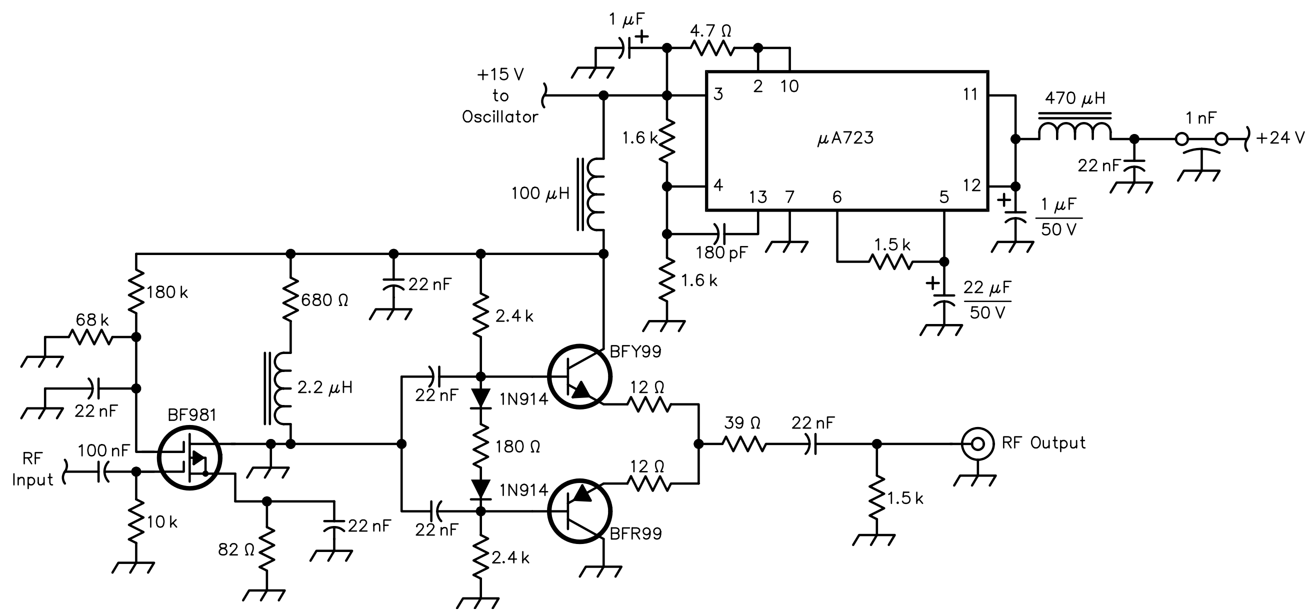


Fig 20—Schematic diagram of a broadband buffer amplifier used in experiments with LC oscillators.

the crystal is then put back in the circuit, it will oscillate at the series resonance. The inductance in parallel with the crystal is there to compensate for the crystal's parallel capacitance. This value can be determined simply, with the crystal and the inductance isolated from the circuit, using a grid-dip meter.

So that the oscillator can be phase locked by the test jig, a Varicap circuit in series with the crystal is used, enabling a frequency variation of a few kilohertz. The output buffer is a common-base transistor stage. The input impedance of this buffer is very low—approximately 3.5 Ω—and therefore does not degrade the Q of the crystal too much. The bandwidth of the buffer is approximately 5 MHz, and it is capable of delivering 32 mW. The power at the output connector is +10 dBm.

The collector's resonant circuit has another Varicap in parallel. It provides us with a phase-modulation input. In my case, a signal between 100 Hz and 100 kHz at a level of 1 V, causes PM sidebands at –60 dBc. This must also be calibrated of course, using a spectrum analyzer, to be useful. The whole circuit is housed in a tinned box that was soldered all around.

This may seem a rather complicated way of making a crystal oscillator, but it really is a good circuit. The oscillator frequency is determined solely by the crystal, and does not depend on the other components as much as in simpler crystal oscillators. It is well buffered and the phase-noise behavior is excellent.

Some Measurement Results

First, I measured the two crystal oscillators with the test setup shown in Fig 18. The levels are plotted in Fig 19 with dots at 1 kHz, 10 kHz and 100 kHz marked "1." In fact, this is not the real phase-noise level of oscillator 1, since in this measurement it is not certain which of the two crystal oscillators is best. If we assume that the two oscillators produce the same amount of phase noise, then the real phase noise level of each of the oscillators is 3 dB lower than the plotted values.

After that, I measured some signal generators available at the laboratory where I was employed at the time, using one of the two crystal oscillators as the reference signal. The results are in Fig 19. In Fig 19, one of my own LC oscillators is plotted, which shows that it is possible to make a good oscillator yourself.

A Broadband Buffer

During my experiments with LC oscillators, I had some trouble with injection locking of the two oscillators connected to the test jig. It became obvious that this was caused by insufficient buffering of the oscillators. I therefore designed a broadband buffer amplifier. It is shown in Fig 20. It is capable of delivering some 100 mW into 50 Ω, has a high input impedance and a voltage gain of three. The current through the complementary output stage must be adjusted to approximately 10 mA by altering the value of the resistor between the two diodes in the base circuit. The bandwidth is about 100 MHz and the isolation is better than 65 dB. With a 50-Ω source, the noise figure is approximately 15 dB.

Using this buffer, no injection-locking problems were experienced anymore. The buffer produces some deterioration in the high-frequency phase-noise levels, but this can only be noticed when using very good oscillators.

Conclusion

I hope that this article gives a good idea of how phase-noise measurements can be performed with rather modest equipment. Realize that my experiments were done in 1990, so some of my circuits are based on components that I had available at the time. If my workload allows, I intend to design another setup using a delay-line discriminator at some time in the future. Such a setup has the advantage that it requires no reference source, although the sensitivity may not be as good as in this setup. If I succeed, I will report my experiences in *QEX*!

References

1. *Reference Data for Engineers* (Indianapolis, Indiana, USA: H. W. Sams) 6th edition, 1975.
2. Ulrich L. Rohde, KA2EUW, *Digital PLL*

Frequency Synthesizers: Theory and Design, (Englewood Cliffs, NJ: Prentice-Hall, 1983).
3. *Spectrum Analysis, Amplitude and Frequency Modulation*, Hewlett-Packard Application Note 150-1, 1989.
4. *Phase Noise Characterization of Microwave Oscillators, Phase Detector Method*, Hewlett-Packard Product Note 11729B-1, 1983.
5. *The "Art" of Phase Noise Measurement*, Hewlett-Packard RF and Microwave Measurement Symposium and Exhibition, Dieter Scherer, May 1983.
6. *Low Phase Noise Applications of the HP8662A and 8663A Synthesized Signal Generators*, Hewlett-Packard Application Note 283-3, 1986.
7. Bernd Neubig, DK1AG, "An Extremely Low-Noise 96 MHz Crystal Oscillator for UHF/SHF Applications," Part 1, *VHF Communications,* 1981/3, pp 135 to 143; Part 2, 1981/4 pp 194 to 203. *VHF Communications* back issues can be ordered from KM Publications, 5 Ware Orchard, Barby, Nr.Rugby, CV23 8UF, UK.
8. Bernd Neubig, DK1AG, "A Modern Professional Look at the Design of Stable, Crystal Oscillators Working at High Frequencies," Part-1, *VHF Communications,* 1991/1, pp 35 to 42; Part 2, 1991/2, pp 74-79. See Reference 7 for contact information.

Jos van der List obtained his Dutch class-A license on his 18th birthday, 25 years ago. He is married and has two adopted children. His chief Amateur Radio interests are home-brew equipment and measurement techniques. He has build equipment for all bands from 1.8 MHz through 10 GHz.

Jos has worked for many years in type-testing laboratories, both in technical and managerial roles. He recently got a new job at the well-known Dutch research laboratory, TNO, where he designs radio front ends (0 to 10 GHz).

A Temperature-Compensated DDS VFO

Use software compensation to stabilize WB2V's popular DDS VFO within 0.5 ppm from 0 to 65° C.

By Curtis Preuss, WB2V

Introduction

One of the challenges in the design of oscillators is to reduce temperature-induced output frequency changes. Frequency drift *versus* temperature can be a particularly onerous problem in VFO designs. Various frequency synthesis techniques can provide a low-drift alternative to the traditional analog VFO by using a more stable crystal oscillator as a reference frequency. However, crystal oscillators are not immune to drift. For example, over a temperature range of 0 to 70°C, oscillators using an AT-cut crystal may vary 20-40 parts per million (ppm) in frequency. On the HF bands, this drift can cause errors of hundreds of hertz in the output frequencies.

5150 Timberidge Ct SE
Rochester, MN 55904
wb2v@aol.com

Many techniques to control oscillator drift have been developed. These techniques include ovens, double ovens, specially cut crystals and a great variety of temperature-compensation techniques. Temperature-compensation techniques can be difficult for amateur experimenters. A tedious process of adjusting component values may be required, and the results can be disappointing.

Lately I've been experimenting with a temperature-compensation technique that employs direct digital synthesis (DDS). In this technique, compensation is accomplished in software rather than hardware. With this compensation technique, I was able to demonstrate a DDS VFO that had 0.5 ppm drift over a temperature range of 0 to 65°C. Fig 1 shows the measured VFO drift with and without compensation.

A Little History

My experiments employed a temperature-compensation system that was invented[1] in 1973. Fig 2 is a block diagram of that system. The system works by adjusting the control inputs of a "presettable" frequency divider. The adjustment is made a function of temperature so that it cancels the drift in the reference oscillator, producing a divider output that is temperature compensated.

It's probably safe to say that in the early '70s, this system required many

[1]Notes appear on page 45.

parts. However, with present-day microcontrollers and single-chip direct-digital synthesizers, the system can be implemented using two off-the-shelf chips, along with a few other components. A microcontroller can be programmed to calculate a frequency control word "W" for a DDS chip, where "W" is a function of both temperature and another external input.

A DDS chip is a convenient way to implement the presettable frequency divider portion of the system. An interesting example of using direct digital synthesis for temperature compensation was published[2] in 1989. It combined DDS with a novel temperature-sensing scheme,[3] where the crystal itself was used to sense temperature. This oscillator was reported[4] to have a stability of 0.03 ppm from –55 to 85°C.

The particular implementation for my experiments is shown in Fig 3. This system is very similar to a DDS VFO I've described previously.[5] On the hardware side, the difference is the addition of a resistor/thermistor voltage divider and the substitution of a microcontroller that contains an analog-to-digital converter (ADC). On the software side, the code[6] is changed so that it adjusts the data sent to the DDS chip according to the temperature indicated by the thermistor.

Test Setup

In order to collect frequency-versus-temperature data, I built a small, insulated test chamber. Thermoelectric modules[7] (also known as Peltier-effect modules) were used to pump heat in or out of the chamber. By controlling the current through the thermoelectric modules, the chamber temperature could be varied. My construction techniques limited the temperature range to 0 to 65°C.

The resolution was limited to one-degree steps by the control circuit—not state-of-the-art performance, but it saved many trips to the kitchen, and freed up the refrigerator and oven for normal use.

A crystal oscillator along with a thermistor, buffer amplifier and voltage regulator were placed in the thermal test chamber. The oscillator was a common-base Colpitts using a fifth-overtone AT-cut crystal. A bead thermistor was held in place on the crystal using heat-shrink tubing. The buffer amplifier was a pair of 74AC04 inverters wired in parallel. Two 78L05s were used for voltage regulation, one for the oscillator and one for the buffer amp. The buffer amplifier output was ac coupled to a short run of coax. At the DDS-chip clock pin, the coax was terminated with 100 Ω to ground and 100 Ω to V_{DD}.

The DDS chip and microcontroller were kept outside the thermal chamber to reduce the heat load. A DDS VFO—minus the microcontroller—

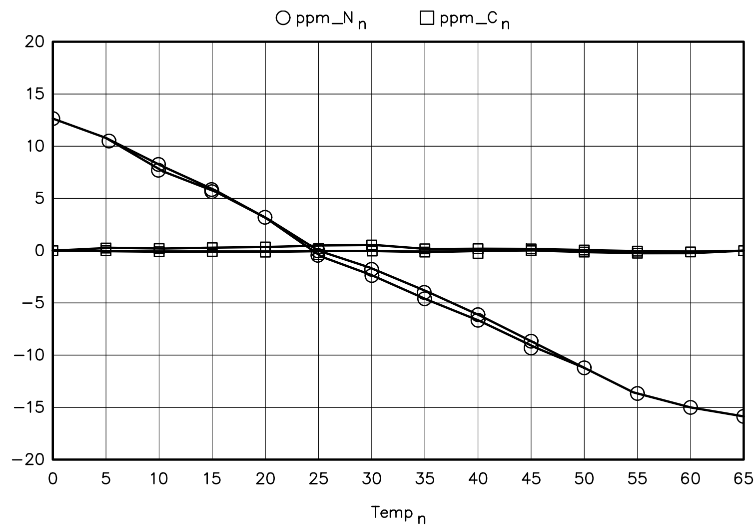


Fig 1—DDS VFO output drift with/without compensation.

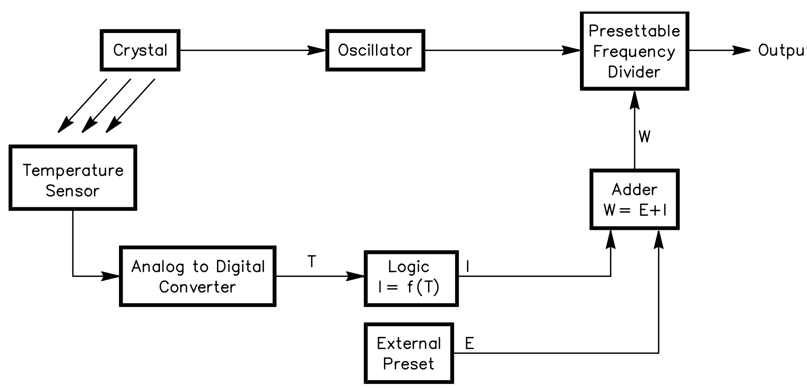


Fig 2—An externally compensated oscillator, circa 1973.

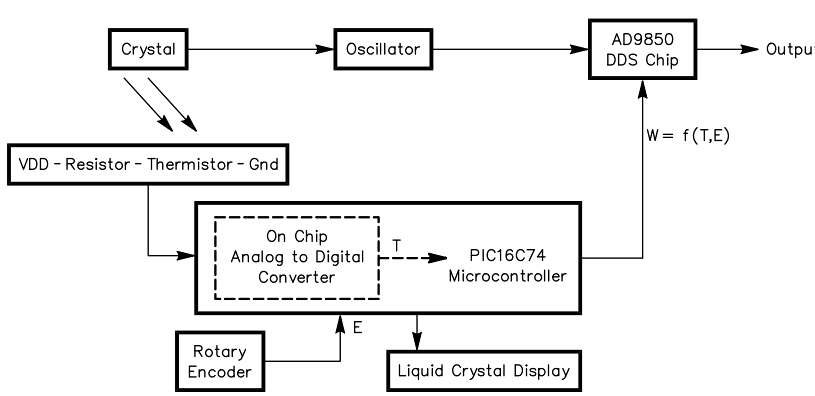


Fig 3—A temperature-compensated DDS VFO.

was mounted on a piece of perforated board along with a PIC16C74 microcontroller, LCD and shaft encoder. These were wire-wrapped together, with the PIC16C74 wired to replace the original PIC16C54 on the DDS VFO. A switch was connected to one of the microcontroller inputs so that compensation could be turned on or off. The complete setup is shown in Fig 4.

Factors Affecting Temperature Compensation

Hysteresis

Hysteresis is a significant problem for any temperature-compensation scheme. Here, hysteresis means that the oscillator frequency at a particular temperature depends on what the previous temperature was. A measurement of this effect can be seen in Fig 5. Data for the plot was taken in 5° steps around the cycle 25°, 65°, 0°, 20°C. After each step, 30 minutes were allowed for everything to reach a stable temperature.

Much of the literature on this subject agrees on one thing: The exact causes of hysteresis are not well understood. Consequently, it is not possible to control hysteresis very well. There are some oft-cited observations[8] about hysteresis:

- The crystal itself exhibits hysteresis.
- Other oscillator components can make the problem worse, often dominating.
- The effect is larger over wider temperature swings.
- The hysteresis of different crystals may have different "signs." That is, the frequency for increasing temperature may be higher or lower than the frequency for decreasing temperatures.
- The magnitude of the hysteresis may vary significantly from unit to unit.

Unit-to-Unit Variations

The frequency-versus-temperature characteristic of crystals may vary greatly from unit to unit. This means that in order to achieve good results, each oscillator must be individually characterized. The compensation data for one oscillator will not work well for another. For example, Fig 6 compares two copies of the oscillator circuit I used.

Numerical Issues

Temperature-compensation techniques rely on a known relationship between an oscillator's frequency and temperature. This relationship is measured, then stored in the microcontroller's memory. The frequency-versus-temperature data can be stored as a table. Then interpolation can be used to calculate points between table entries. Another storage method is to fit the measured data to a polynomial and store only the polynomial coefficients. There are inevitable differences, or *residuals*, between the measured data and the stored representation. These residuals can be reduced by using higher-order polynomials or improved interpolation techniques.[9, 10]

Since my experiments were carried out over a limited temperature range, the problem of residuals was greatly reduced. Crystals with an AT cut have an S-shaped frequency-versus-temperature characteristic. Likewise, a plot of the voltage versus temperature of a resistor-thermistor voltage divider displays a complex curve. Over a limited temperature range, however, the relationship of thermistor voltage to frequency is nearly a straight line. This can be seen in Fig 6.

The resolution of the ADC is in play. The slope of the frequency drift versus thermistor voltage for oscillator B was about 12 ppm/V. Using an 8-bit ADC over a 5-V input range provides a resolution of about 20 mV. This translates into a frequency resolution of about 0.23 ppm. For oscillator A, the limit was about 0.33 ppm.

On the other hand, the resolution of the AD9850 DDS chip was not a significant factor, except at low output frequencies. With a 32-bit accumulator, and the reference oscillator at 120 MHz, the DDS frequency resolution is about 0.028 Hz. This is 0.1 ppm if the VFO output frequency is 280 kHz, but at 5 MHz, the DDS resolution is 0.0056 ppm.

Thermal Time Constants

Thermal time constants can cause more problems. Some time is required for the resonating portion of the crystal to reach the same temperature as a thermistor on the outside of the crys-

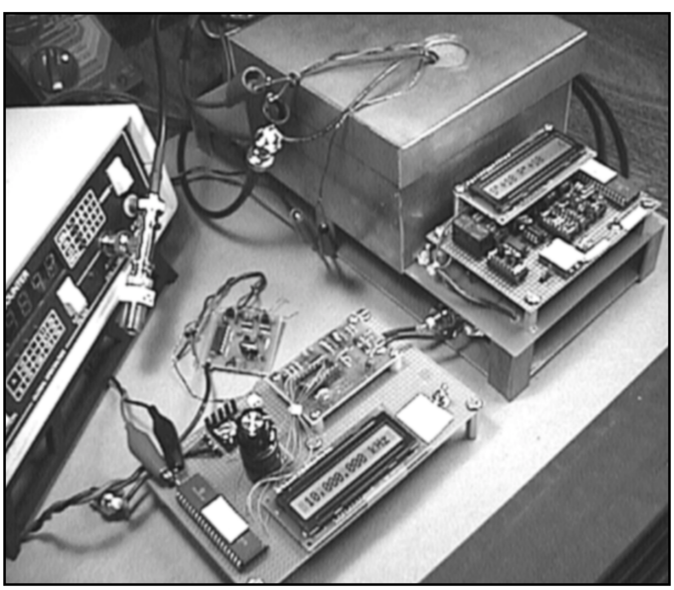


Fig 4—A Photograph of the test setup, showing: thermal chamber, DDS VFO assembly, oscillator assembly and a frequency counter.

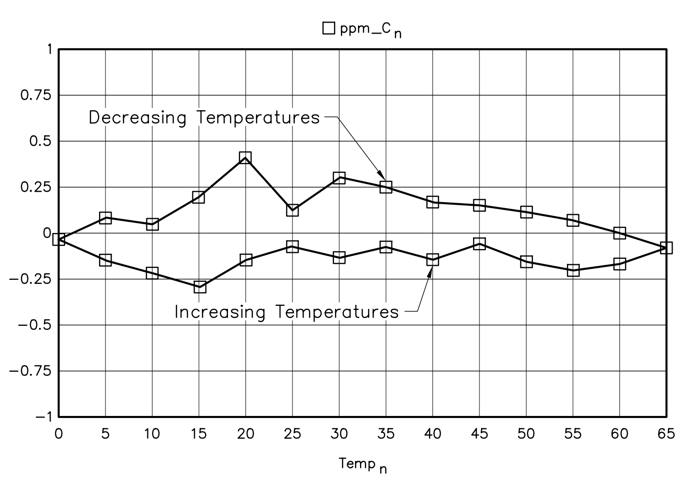


Fig 5—Measured DDS VFO hysteresis.

tal holder. Power-on warm up is one situation where thermal time constants can be noticed; another is during a rapid temperature change.

My frequency counter was too slow to directly measure the effects of thermal time constants, but I was able to observe the warm-up drift by listening to the VFO output on a receiver. The signal could be heard to shift back and forth a few hertz as the temperature changed, but it was difficult to distinguish between the effects of thermal time constants and the limited resolution of the ADC.

Conclusion

These experiments verified the effectiveness of adding temperature compensation to a DDS VFO. A significant reduction in frequency drift was obtained with very little additional hardware. However, time and effort were required to measure the frequency-versus-temperature characteristic of the reference oscillator.

Software-compensation techniques provide a great deal of flexibility. They're able to compensate for complex relationships between temperature and frequency. For the limited temperature swing of 0 to 65°C, the oscillator was characterized quite closely by linear interpolation. For wider temperature swings, a more sophisticated interpolation or curve-fitting technique should be used.

Good results were obtained using an 8-bit ADC. A 10-bit converter may give better results, but at some point, hysteresis and thermal time constants will limit the stability that can be achieved.

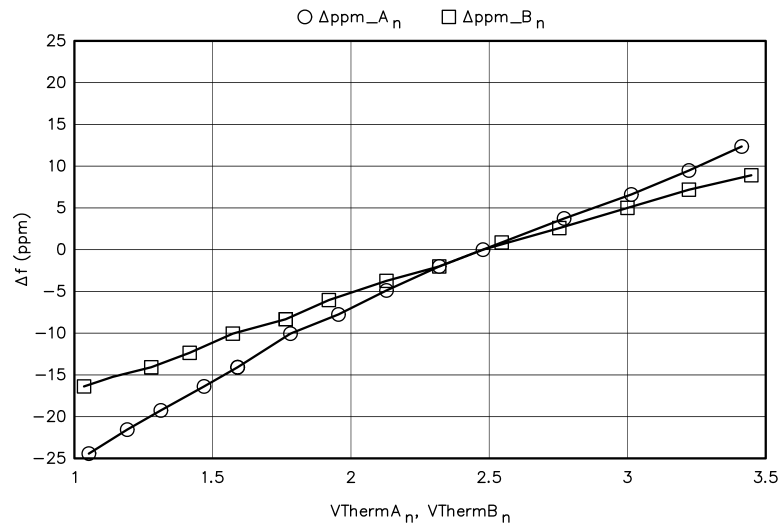


Fig 6—Drift versus thermistor voltage for oscillators A and B.

Notes

[1]US Patent 3,719,838: "Temperature Compensating Digital System for Electromechanical Resonators," Ralph Peduto and Jan Willem L. Prak, issued March 6, 1973, see claims 12-14.

[2]A. Benjaminson and S. Stallings, "A Microcomputer-Compensated Crystal Oscillator Using a Dual-Mode Resonator," *Proceedings of the 43rd Frequency Control Symposium*, 1989.

[3]S. Schodowski, "Resonator Self-Temperature Sensing Using a Dual-Harmonic-Mode Crystal Oscillator," *Proceedings of the 43rd Frequency Control Symposium*, 1989.

[4]A. Benjaminson and B. Rose, "Performance Tests on a MCXO Combining ASIC and Hybrid Construction," *Proceedings of the 45th Annual Symposium on Frequency Control*, 1991.

[5]C. Preuss, WB2V, "Building a Direct Digital Synthesis VFO," *QEX*, Jul 1997, pp 3-7.

[6]You can download the experimental code with the new function from the ARRL Web **http://www.arrl.org/files/**. Look for DDS_VFO.ZIP.

[7]URL **http://www.itiferrotec.com** has a lot of technical information about thermoelectric modules.

[8]R. Filler, "Thermal Hysteresis in Quartz-Crystal Resonators and Oscillators," *Proceedings of the 44th Frequency Control Symposium*, 1990.

[9]M. Frerking, "Crystal Oscillator Design and Temperature Compensation", Van Nostrand Reinhold Company, 1978, Chapter 10.

[10]R. Filler, "Frequency-Temperature Considerations for Digital Temperature Compensation," *Proceedings of the 44th Frequency Control Symposium*, 1990. □□

Practical Application of Wind-Load Standards to Yagi Antennas: Part 1

Is your antenna rated to survive local storms? How much wind load does it put on your mast and tower? Basic physics and a little engineering can provide answers. In Part 1, we learn how wind acts on flat plates and cylinders.

By Stuart E. Bonney, K5PB

Whenever a fierce wind arises, or we start to think about newer, bigger beam-antenna systems, we are reminded that wind plays a large part in antenna-system survival. We need a reasonably simple —but accurate and consistent—way of evaluating wind loads on antennas, masts and towers. We also need a means to determine expected wind velocities in our own local areas. If we plan to use purchased antennas, we need a way to evaluate their wind-related properties reliably. There is a host of other practical issues as well, but these are fundamental.

Electronic Industries Association (EIA) Standard 222 is often used as a basis for antenna wind-load analysis and specifications. Its more recent versions provide improved, more accurate and detailed methods that conform with widely accepted practices. These later releases include data useful when estimating peak wind velocities at a given location and installed antenna height. Although any required permits and approvals are usually governed by local building codes, the process can be simplified if our methods are compatible with those codes as well.

This standard has gone through several incarnations over many years. Probably the best-known version of these is RS-222-C,[1] mainly because of its longevity and wide circulation. Published in 1976, it is still often quoted, or its methods used, although it has been superseded by three later releases: EIA-222-D in 1987, EIA/TIA-222-E in 1991 and TIA/EIA-222-F[2] in 1996. Like an older standard for bridges, RS-222-C expressed wind force-versus-velocity in terms of pressure on a flat plate. This was the source of the widely published figures for wind pressure in pounds per square foot (psf): 70.7 mph = 20 psf; 86.6 mph = 30 psf; 100 mph = 40 psf and so on. As we will see shortly, these numbers are valid only under a specific set of conditions, and not in the broad way that they have often been applied.

Effective with EIA-222-D, this standard was made more consistent with other widely used structural codes,

802 Melrose Dr
Richardson, TX 75080

[1]Notes appear on page 50.

such as the Uniform Building Code, the Standard Building Code and the British Code of Practice (CP3). This release included a basic expression for dynamic wind pressure, as well as explicit drag coefficients for both flat and cylindrical shapes. It also included height and wind-gust factors. Releases 222-E and 222-F also contain more detailed, refined wind-velocity data by state and county from NOAA and National Weather Service sources.

Fluid dynamics and their application to antennas are not exactly commonplace subjects. Our objective here is to facilitate an understanding of the underlying principles and offer practical ways of applying such insight based on the latest standard, EIA-222-F. We will also examine methods and definitions that (we hope) will help overcome some of the confusion that has often resulted from shortcomings in the now obsolete RS-222-C. The material presented here does not address all of the many details relating to Yagi antenna systems, especially for large arrays used in severe environments. That would require much more than just an article or two.

Basics of Wind and Wind Loads

As a prelude to practical applications, and as a means of understanding where the numbers come from, let's review the basic principles of wind loads on structures such as antennas. Wind is air in motion, and air has mass. Therefore, wind has energy that can be quantified. This energy is a function of air mass density, m, and velocity, V, as expressed in Eq 1 below.[3]

We can get the weight of one cubic foot of air at standard temperature and pressure (STP) from handbooks, then divide it by 32 to get its mass and apply a factor to convert velocity in feet per second to miles per hour. This yields an equivalent form expressed in Eq 2:

$$P = 0.5\ m\ V^2 \qquad \text{(Eq 1)}$$

or,

$$P = 0.00256\ V^2 \qquad \text{(Eq 2)}$$

These are expressions for fluid dynamic pressure. Eq 1 is a form of Bernoulli's equation and is related to $E_k = 0.5\ m\ V^2$, the fundamental expression for kinetic energy. Eq 2 can be found in many college-level physics and engineering textbooks, although the symbol q is often used instead of the symbol P. Later versions of EIA-222 have adopted the symbol qz.

The two equations express the kinetic energy and resulting dynamic pressure of a wind stream at a given velocity, but *they do not express the net force on an object in that stream*. When wind encounters an object, both the magnitude of the resulting pressure and its gradient around that object vary with object shape and other factors.[4]

To elaborate, let us consider a flat plate of finite height, but great length—approaching infinite. Imagine this plate is oriented broadside to oncoming wind, as shown in the cross-sectional view of Fig 1A. As the wind encounters this object, the wind stream divides and flows over its top and bottom edges, where the flow then becomes turbulent. Due to the nature of this flow, not only does the wind exert direct positive pressure on the plate's front side, but significant negative pressure also develops on its backside.

Fig 1B shows the approximate pressure distribution around this plate. Pressure on the front side is equal to that expressed by Eq 2, but decreases toward either edge. Negative pressure, or suction, on the backside actually reaches a value greater than the maximum positive pressure on the front. The net effect of these pressures causes a plate of this particular shape to behave as if it were subjected to *twice* the force of the direct wind pressure alone.

The situation for a very long cylinder is similar, but because of its more streamlined shape, the airflow and pressure distribution are different. Both positive and negative pressures can reach peak values similar to those on a flat plate, but pressure gradients around a cylinder are such that average pressures are much lower. Theory and experimental results described in numerous texts indicate the net force to be about 60% of that on a flat plate of equal length and height.

Next, we must consider aspect ratio, which is the ratio of length to height (L/H) for a flat plate, or length to diameter (L/D) for a cylinder. When the aspect ratio is finite, wind can flow around the sides of an object as well as over its top and bottom. This reduces average pressures over the surface. This reduction continues as aspect ratio decreases. For an aspect ratio of 1, the net force falls to a little more than half that for an infinite L/D ratio. The effects of shape and aspect ratio are why equating wind velocity with a fixed pressure or force on objects can be misleading.

Calculating actual wind forces on an object would be greatly complicated by these dependencies were it not for the introduction of another factor. This is referred to by aerodynamicists as *drag coefficient*. You may encounter the term *force coefficient*, but for our purposes, the terms are interchangeable. Table 1 lists values for flat plates as well as cylinders. Drag is also influenced by the relative surface smoothness of an object, wind speed in relation to the speed of sound, and an object's size in relation to wind velocity. Drag coefficients shown are composites from several sources and are applicable to typical beam antennas and the wind conditions under which antennas are used.

The final factor required to express the actual force on an antenna element—or an entire antenna or a mast—is *projected area*. For any object, this is its "shadow area," such as an object would cast on a nearby parallel surface when illuminated by the distant sun. For flat plates, it is simply height times width; for cylinders, it is length times outside diameter.

We can now put all three factors together, resulting in the following equation for objects that are broadside to the wind:

$$F = (P)\ (Cd)\ (A) \qquad \text{(Eq 3)}$$

where

F = force in pounds
P = dynamic wind pressure in pounds per square foot
Cd = drag or force coefficient (dimensionless)
A = projected area in square feet.

By substituting the expression for P from Eq 2 into Eq 3, we can find the actual force on an antenna-system

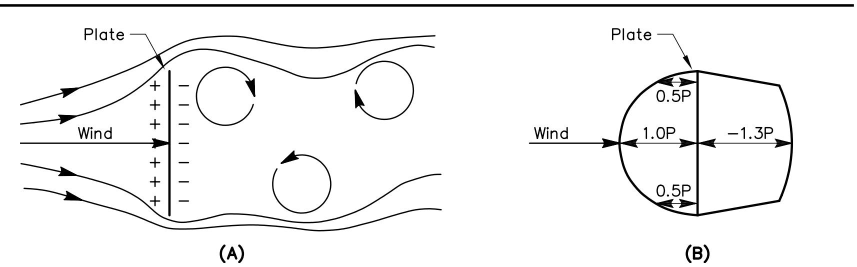


Fig 1—A shows wind flow around a flat plate. B shows pressure distribution around the same plate.

component for any selected wind velocity. Keep in mind that this applies to objects broadside to the wind, subjecting them to maximum force.

$F = 0.00256\,(V^2)\,(Cd)\,(A)$ (Eq 4)

We would not need to go beyond Eq 4, except for one important fact: We don't always know what values to use for *Cd* and *A*, especially for a purchased antenna. Some manufacturers do not specify a value for *A*. Further, if its value—or some equivalent—is specified, but we don't know its basis, or whether it already includes some unstated value for *Cd*, we cannot calculate actual wind loads with any degree of confidence or accuracy. It is a *very* desirable convenience for both manufacturers and purchasers to be able to specify or determine antenna wind-load characteristics simply, accurately and uniformly.

A great deal of this uncertainty is attributable to ambiguity in RS-222-C, which was recognized and corrected in subsequent releases. We now take a brief look at this issue and related problems with this standard. Closer examination suggests that it contained significant lapses. It expresses wind force in psf on a flat plate as $P = K\,V^2$, where $K = 0.004$ and "includes a gust factor and a drag factor for flat surfaces" that are, however, not quantified. With regard to cylinders, 222-C was silent, except for stating, "In all cases, the pressure on cylindrical surfaces shall be computed as being ⅔ of that specified for flat surfaces."

If we use the widely accepted wind pressure coefficient 0.00256, and assume a gust factor of 25%, we can derive drag coefficients from the above equation. They are effectively about 1.25 for flat plates and 0.8 for cylinders, both of which are much lower than generally accepted values. On the other hand, if *K* remains at 0.004 and we assume it contains no gust factor, effective drag coefficients are about 1.6 for flat plates and 1.0 for cylinders. These values more closely reflect actual antenna wind-load properties. Thus, it can be concluded in a practical sense that 222-C does not include a gust factor, despite what it says. Although 222-C has worked fairly well if users did not also depend on an implicit gust factor in determining antenna survivability, it leaves much uncertainty over how wind surface areas are defined.

The Search for Effective Wind Surface Areas

For many years, the term "effective area" or something similar has been used to specify an area that is usable for calculating antenna wind loads.[5] Unfortunately, the absence of accurate, universally understood and applied definitions has rendered such terms all but meaningless. To illustrate, I have an old manual for a popular four-element, 20-meter Yagi, which specifies the antenna's effective area as 3.9 square feet. A more recent catalog lists 7.3 square feet for the very same antenna. This is not a unique instance. Another technical article addressing Yagi antenna wind loads notes that inconsistencies are common and widespread.[6]

Some manufacturers and users of 222-C have interpreted the ⅔ factor for cylinders to mean that effective areas are ⅔ those of equivalent flat plates. Others have apparently applied this factor to the specified value of coefficient *K* and taken effective

Table 1
Composite Drag Coefficients (Cd) For Various Aspect Ratios

Aspect Ratio	Cd *Flat Plate*	Cd *Cylinder*
∝	2.0	1.2
100	1.8	1.1
40	1.6	1.0
10	1.4	0.84
5	1.2	0.72
1	1.16	0.70

Note: See Appendix for discussion.

Table 2
Drag Coefficients from Various Sources

Source: Mechanical Engineering In Radar And Communications[8]

Aspect Ratio	*Correction*
0 to 4	0.6
4 to 8	0.7
8 to 40	0.8
> 40	1.0

Same table (credited to British CP3) appears in A. J. MacDonald, *Wind Loading On Buildings*,[7] who also gives *Cd* of 1.0 for infinite cylinder (Reynolds number 10^3 to 10^5).

Source: Standard Handbook for Mechanical Engineers[9]

Aspect Ratio	*Cd Flat Plate*
1	1.16
4	1.17
8	1.23
12.5	1.34
25	1.57
50	1.76
∝	2.0

Cd = 1.11 listed for discs.

Source: Mechanical Engineering Handbook[10]

Aspect Ratio	*Cd Flat Plate*
1	1.18
5	1.2
10	1.3
20	1.5

Cd of 1.17 listed for discs. Reynolds number = 10^5, source cited: S. F. Hoerner

Source: EIA/TIA-222-E/F for "Appurtenances" (including antennas)[2]

Aspect Ratio	*Cd Flat plate*	*Cd Cylinder*
≤ 7	1.4	0.8
≥25	2.0	1.2

area to be the same as projected area. By assuming a gust-factor value, one can also derive otherwise unspecified drag coefficients. However, each of these approaches produces a different result for effective area. We very rapidly reach a point where it is impossible to determine where we stand without a detailed description of the method and assumptions used, which usually are not readily available.

This discussion would be pointless if not for the fact that 222-C's dubious methodology often remains in use and continues to be a source of confusion. However, this is not a case of who's right or wrong; it is regrettable if anyone chooses to view it in that light. The problems are within the document, and it seems well past time to retire it, especially since its successors provide a basis for more consistent results.

Practical Solutions and Their Results

Confusion and inconsistency concerning effective area could be easily ended by broad adoption of a specific definition. The following, consistent with current standards, is proposed:

Wind surface area (WSA) is defined as projected area (A) of an object of interest, multiplied by the drag or force coefficient (Cd) appropriate to the basic shape and aspect ratio of that object.

Stated as an equation:

$$WSA = Cd\,(A) \qquad \text{(Eq 5)}$$

Appendix: Practical Drag Coefficient Considerations

Whenever we design or evaluate an antenna element or boom for survivability, we want to know how much wind it will tolerate and how much wind load it will accumulate at a given wind velocity. Actually, the two are interrelated, because increasing diameters for increased strength also means more wind load, resulting in more stress. Drag, of course, is an integral part of all this, so we need to know as accurately as we can just what drag coefficient to use. This is a different viewpoint from that of the structural engineer, who is concerned with the practical aspects of meeting applicable codes and signing off on them. In that case, application of a particular drag coefficient is usually mandated. The two viewpoints do not necessarily conflict, but they can result in different approaches.

Among the numerous technical sources I've researched, there is general agreement on basic *Cd* values of 2.0 for a flat plate of infinite aspect ratio, and 1.2 for a corresponding cylinder. There is less agreement on values for aspect ratios of finite dimensions. Part of the reason may be the many variables affecting aerodynamic drag; they make analysis difficult, and experimental setups are vulnerable to measurement variations. Table 2 shows several examples of *Cd* values for various aspect ratios. Some sources list aspect ratios in groups, and include correction factors to be applied to basic *Cd* values.[7, 8] This is clearly a shortcut, since the relationship between *Cd* and aspect ratio is not a step function. Other sources also put aspect ratios in groups, but list discrete *Cd* values for each, noting that intermediate values can be interpolated.[9, 10]

As shown in the table, the parts of 222-E/F dealing with attachments, such as antennas, list only two *Cd* values, each, for flat and cylindrical surfaces. One is for aspect ratios of seven or less, the second for ratios of 25 or more. Values for ratios between these steps are to be interpolated. We should keep in mind that these standards were developed primarily for towers, where failures can result in total collapse. Safety and legal considerations, and a desire for substantial overload margins, justify a quite conservative approach, which seems evident here.

Our goal of accuracy in selecting drag coefficients for antennas, especially in a design sense, is primarily to achieve the best balance between wind loading characteristics and survivability. Certainly, it is wise to allow reasonable margins for unexpected overloads, which we can do by designing to a sufficiently high peak wind velocity. At the same time, we do not want to over-design or over-specify. That can lead to larger, heavier, more costly construction and consequent higher loads on masts, rotators and towers, resulting in a cascade effect on total system costs.

While much wind-tunnel test data exists for various shapes, I've found none that deal with step-tapered cylinders. We know that the typical tapered element has at least a partial path for wind to flow around the end of a segment where it tapers down to the next smaller segment. When swages or inserts are used for greater step-down in tubing diameter, this end-flow path becomes larger. Although this effect remains to be quantified and may be relatively small, we can be reasonably sure it exists.

The *Cd* values listed earlier in Table 1 are composites of data extracted from several sources, including those discussed above. Care was exercised to ensure their validity for conditions normally applying to antennas, ie, Reynolds numbers from 10^3 to 10^5. (A note regarding Reynolds number: It relates object dimensions, wind velocity, and air viscosity.) These values were then plotted graphically to aid in visualizing spreads and to determine the best overall fit. Based on this data, the foregoing considerations and practical experience, it appears that a reasonable value of *Cd* for cylindrical elements and booms lies between 1.0 and 1.1. A value of 1.1 would be slightly conservative without being excessive. Again, if you must satisfy requirements of a building or structural code, it is best to use whatever figure that code specifies.

The values in Table 1 are not presented as absolutes; they should be considered a workable basis only until better data becomes available. In passing, it is interesting to note that the calculated *Cd* for cylinders from RS-222-C is 1.04 if no gust factor is assumed. This explains why 222-C has worked reasonably well despite its large potential for confusion. Applying the 2/3 pressure factor to *K* as specified, 0.004 (2/3) = 0.00267. Dividing this by the basic dynamic wind pressure constant 0.00256 yields 1.04, the effective *Cd*. The more usual form $F = 0.00256\ V^2\ (Cd)\ (A)$ then emerges, but the standard certainly is not clear as written.

Finally, there are a few points worth mentioning about the mathematical model described here for wind loads. For simplicity of calculation without impairing basic accuracy, this model assumes that objects, such as antenna elements, always remain fully broadside to the wind. In the real world, elements deflect, and in high winds can shed wind loads amounting to several percentage points of calculated values, depending on deflection angles. These angles, which vary between root and tip, depend in turn on wind velocity and element construction. This effect contributes to wind survival margins for elements but to a much smaller extent for booms, which are usually stiffer. Thus, the model is essentially conservative.

As discussed in the Appendix, a drag coefficient of 1.1 appears to be—in a purely technical sense—accurate for cylindrical elements and booms of typical beam antennas, but this is not the only consideration. Building and structural codes often mandate a value of 1.2, and for manufactured antennas, this is the best figure to use. Substituting Eq 5 into Eq 4, we have a practical and uniform basis for determining wind loads on antenna elements, booms, complete antennas or masts:

$$F = 0.00256\ V^2\ (WSA) \qquad \text{(Eq 6)}$$

or, alternatively:

$$F = V^2\ (WSA)\ /\ 390 \qquad \text{(Eq 7)}$$

where:

F = force in pounds
V = wind velocity in miles per hour
WSA = wind surface area in square feet.

Note that this method results in effective areas that are larger than we may be accustomed to seeing. This is a byproduct of doing away with problematic flat-plate-equivalent methods. However, this newer method has the distinct advantage that all adjustments needed to account for different shapes and aspect ratios are incorporated into a single, net figure: *WSA*. The same dynamic-wind-pressure constant is used at all times, removing all ambi-guity.

Despite the increase in areas, net broadside-wind-load forces computed in this way will not be greatly different in many cases than with the old methods. Among the reasons is that the pure dynamic wind pressure values calculated with 222-D/E/F are only about 65% of values commonly associated with 222-C for various wind velocities. Further, based on the cross-flow principles described by Weber (see Ref. 6), wind loads for complete antennas, when rotated off the wind at angles of 30 to 50°, will be lower. How to calculate wind loads for typical antennas will be described with examples in Part 2 of this series.

We now arrive at the specifications antenna manufacturers should provide to users, or that you need to determine if building your own. First is a combined *WSA* figure for the elements. Second is a separate *WSA* figure for the boom. The reason you need both is that the antenna configuration influences which case will represent the worst wind load. Normally, for HF beams, the element load is the greater. For multielement VHF beams on long booms, the boom often presents the greatest wind load. If you plan to mount both types on one mast, maximum net wind loads on the mast and tower may be produced by wind that is either broadside to the elements of both antennas or broadside to the booms. You need to be able to evaluate both conditions.

The third item required is antenna wind-survival rating expressed in miles per hour of peak wind velocity. Survival rating refers to the wind velocity that, if exceeded, will result in permanent, measurable deformation to a physical member of the antenna. Some manufacturers already provide this information. Those who do not should begin. Actual computation of wind-survival capability is a fairly complex process of stress analysis, best done with a computer. It is beyond the intended scope of this series.

With *WSA* and wind-survival ratings in hand, you have the basic information you need to evaluate the physical suitability of an antenna for your location and installed height. You can then proceed with installation planning. Using the rated *WSA* and Eq 6 or 7, the force that the mast and tower must handle can be easily found, once the local wind conditions are known.

The next hurdle you will normally face in planning a new or upgraded antenna installation is site engineering. By this we refer to the general process of planning the installation, not to the services of professional engineers who, if needed, usually will have requirements of their own. Included are the applications of antenna height and wind gust factors, determination of peak design wind velocity appropriate to your locale and site topography and practical selection of supporting masts. Part 2 will explain how to do all of this based on 222-D/E/F criteria.

Conclusions

Methods used to analyze wind loads on antenna structures and to specify wind load areas still are often based on, or derived from, the old EIA standard RS-222-C. This standard and methods based on it have been shown to have significant problems, primarily a consequence of ambiguity and absent specifics in the document itself, along with the resulting lack of uniformity in application.

Here, we have briefly reviewed the basic physical principles on which wind-load analysis is based, as an aid to reader understanding and application. As we have seen, these principles and methods have been more consistently and comprehensively implemented in EIA-222-D/E/F. There are several practical advantages to be gained by broad adoption of these improved methods:

1. They provide antenna manufacturers with a uniform, consistent, yet simple method for specifying the wind-load properties of their products. This, in turn, helps to produce a level playing field for all.
2. They provide users of manufactured antennas with a simple means of evaluating wind-load characteristics of these products, and of determining their suitability for a particular installation.
3. While not a substitute for applicable building codes, these methods are consistent with many of these codes, and can simplify the initial stages of application for local antenna permits and approvals.

References

1. EIA RS-222-C, *Structural Standards for Steel Antenna Towers and Antenna Supporting Structures* (Washington, DC: Electronic Industries Association, March 1976).
2. TIA/EIA-222-F, *Structural Standards for Steel Antenna Towers and Antenna Supporting Structures* (Washington, DC: Electronic Industries Association, June 1996).
3. A. L. Sanford and J. M. Tanner (Georgia Institute of Technology), *Physics for Students of Science and Engineering* (New York: Academic Press, 1985).
4. J. Bertin and M. Smith, *Aerodynamics For Engineers*, Second Edition (New Jersey: Prentice Hall, 1989) p 85.
5. E. Fuller, W2FZJ, "The Application of Stress Analysis to Antenna Systems." *Ham Radio*, Oct 1971, pp 23-31.
6. D. Weber, K5IU, "Determination of Yagi Wind Loads Using the 'Cross Flow Principle'," *Communications Quarterly,* Spring 1993.
7. A. J. MacDonald (University of Edinbergh), *Wind Loading on Buildings* (Toronto: Halstead Wiley, 1975) p 81.
8. C. J. Richards, *Mechanical Engineering in Radar and Communications.* (London: Van Nostrand Reinhold, 1969) p 125.
9. Marks' *Standard Handbook for Mechanical Engineers,* 9th Edition (New York: McGraw Hill, 1987) Section 11, p 78.
10. M. Kutz, *Mechanical Engineering Handbook* (New York: Wiley, 1986) p 1525.

Stuart E. Bonney, K5PB, has been licensed since 1952 and formerly held the calls W8JUV, W0DMR and W5PAQ. He is a retired engineer and manager from Rockwell Collins, and is founder and president of a communications and computer services company. He is author of the book "Amateur Single Sideband," published by Collins, two other books on computer software and several articles.

A Handy Coil Winder

Before ferrites came on the scene, every ham wanted a coil winder. Some applications still require solenoid coils with many turns. Here's a winder small and inexpensive enough to rest on a shelf until it's needed.

By Bob Dildine, 7J1AFR/W6SFH

This small coil winder can be built in a few evenings with nothing more than simple hand tools. Constructed from readily available parts, it features variable speed control and a built-in turns counter.

For a recent project, I needed to wind a transformer for a small, high-voltage switching power supply. The secondary consisted of about 500 turns of #34 wire, wound on a small pot-core bobbin. I first attempted to wind this coil with a variable-speed electric hand drill. A simple turn counter was rigged up using a small light bulb, a photocell and the lid from a cat-food can. A small hole punched near the rim of the lid triggered a photocell connected to my frequency counter in its accumulator mode.

It was difficult to securely mount the drill to the worktable, the speed control was troublesome and it was tricky to simultaneously watch the turns counter and the flow of wire onto the bobbin. Although I managed to wind the coil, the results were less than satisfactory. About this time, I saw a simple coil winder based on a toy motor, described in one of the ham magazines here in Japan.[1] That article was the inspiration for this coil winder.

Apt 203, 5-2-2 Denenchofu
Ota-Ku, Tokyo 145-0071, Japan
bob_dildine@HP-Japan-om2.om.jpn.hp.com

Objectives

I set out to build a coil winder that could be used for small coils such as pot cores, transformers and air coils with many turns of fine wire. The following were the objectives for this project:

- Inexpensive
- Simple to build using readily-available parts
- Easy to control winding speed
- Direction reversible
- Built-in turn counter
- Ruggedness

[1] *CQ Ham Radio* (a Japanese periodical), April 1998

The Motor

After investigating the toy motor mentioned above, I decided to use a more substantial unit. At a local motor shop, I found a 400-RPM gear motor with a 6-mm shaft (about 1/4 inch) that could be powered from 12 V dc. A low-voltage dc motor has the advantage that its speed can be easily controlled by varying the supply voltage. Its direction can also be reversed using a DPDT switch or relay.

Speed Control

The speed control is shown in Fig 1. It consists of a simple potentiometer connected as a voltage divider, which feeds an NPN Darlington transistor in an emitter-follower configuration. This provides plenty of current to power the motor and allows the use of a small potentiometer for the control. The motor direction is governed by a DPDT toggle switch.

The Turn Counter

I originally planned to use my frequency counter in the accumulator mode as the turn counter, rather than build a dedicated counter circuit. However, I found a little electromechanical counter at a local surplus shop that turned out to be just right for this application. A disk about 5 cm (2 inches) in diameter was cut from a piece of thin copper-clad circuit board material, and a hole was punched near the rim. I soldered the disk to a 6 mm brass insert from an old plastic knob and fastened the insert to the motor shaft using its setscrew. A Sharp 1A53 photo interrupter is mounted so that it triggers each time the hole in the disk passed by. Each 1A53 contains a Schmitt-trigger inverter that causes the output to go low when the light beam is interrupted. Suitable substitutes can be obtained from Jameco (part number 114091 or 114104) or Digi-Key (various). These substitute parts may have only an open-collector transistor output, which conducts when the light beam is *not* interrupted. If such a part is used, omit the 2N2222 inverter that drives the 555 pulse stretcher.

Refer to Fig 2. The output from the photo interrupter drives a pulse stretcher made from a 555 timer. The mechanical counter is rated for 10 counts-per-second, so the pulse width was set to 50 ms. This gives a square wave at the highest counter frequency. Without the pulse stretcher,

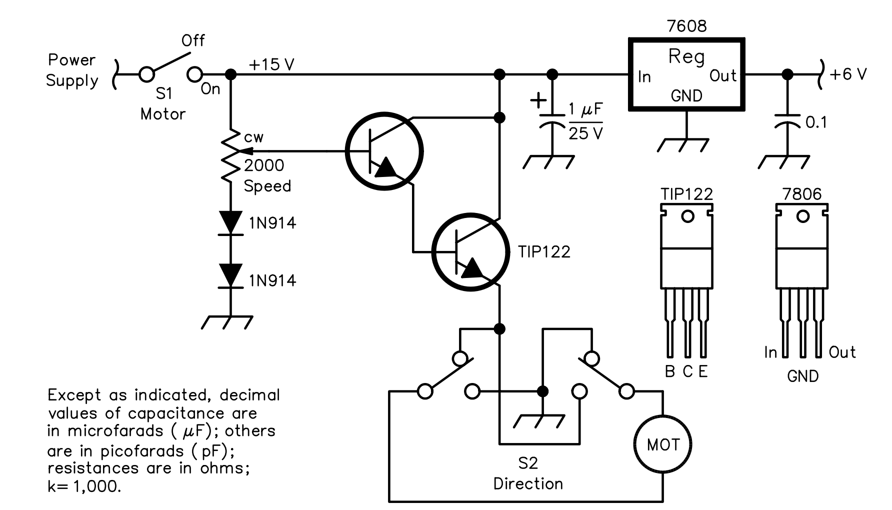


Fig 1—Speed-control circuit.

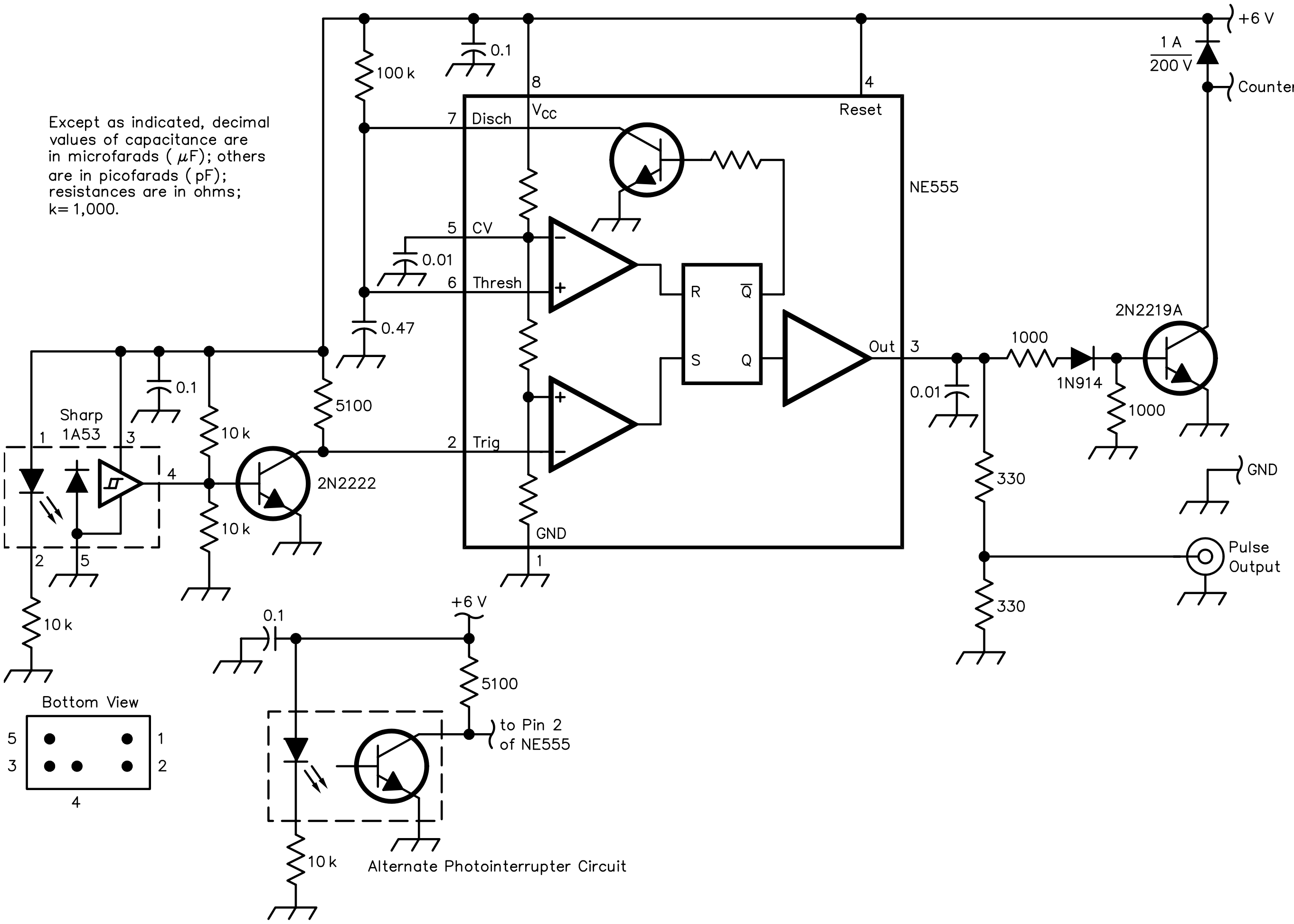


Fig 2—Turn-counter circuit.

the pulses from the photo interrupter were too narrow to trigger the counter at all but the lowest motor speeds. An auxiliary pulse output was connected to my frequency counter—set to its accumulator mode—and the motor was run at full speed long enough to accumulate several thousand counts. The mechanical counter and the electronic counter agreed precisely.

The Power Supply

The motor and counter circuits are powered from a small switching power supply, found at one of the local surplus shops. This supply was an open-frame model, and it's mounted so that it's difficult to accidentally contact any high-voltage points. Because I expected that the coil winder would only be used occasionally, it has no primary power switch. A switch was installed, however, in the low-voltage circuit, to remove power from the motor during set up. A bright LED connected to one of the power-supply outputs shows when the winder is plugged in!

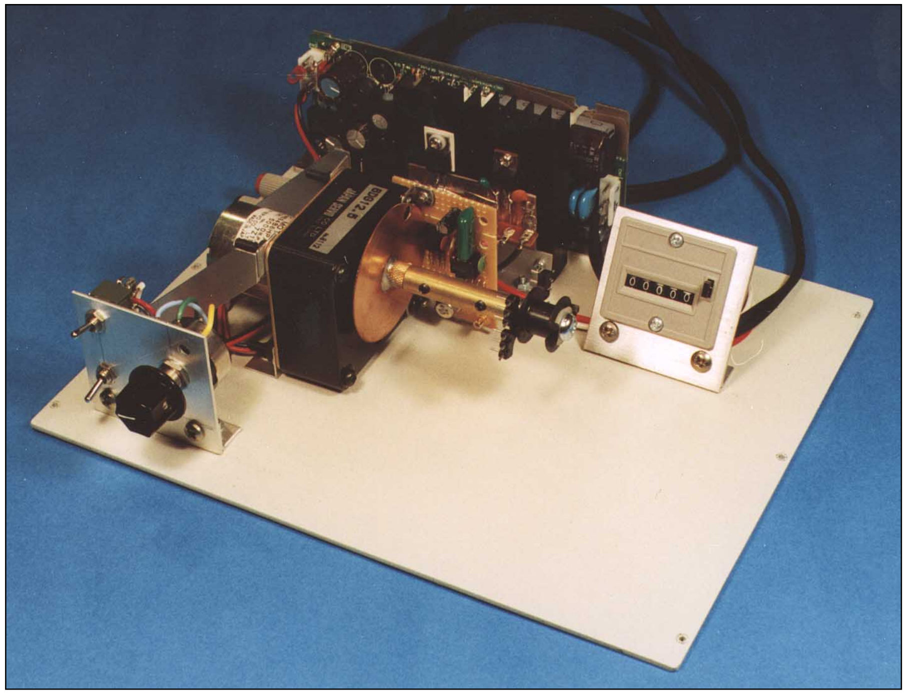


Fig 3—The complete coil winder.

Mechanical Construction

The winder was built on a 26×21 cm ($10^1/_4$×$8^1/_4$ inches) flat aluminum plate, about 3 mm ($^1/_8$ inch) thick. The motor is securely mounted using angle brackets and braces, so that the shaft points to the right. The turn counter mounts on an angle bracket just behind the motor shaft, so it is possible to see the counter without taking your eyes off the coil. The motor speed-control circuitry is built on a small heat sink. The turn-counter circuit is on piece of perforated board, and mounted to the side of the motor. The speed control and reversing switch are mounted on a small bracket on the left-front corner of the coil-winder base. This allows speed control with the left hand, while feeding wire with the right. "Southpaw" builders may wish to reverse the orientation of the motor and control panel.

A 6-mm shaft coupling joins the motor shaft to a 6-mm bolt that holds the bobbin or coil being wound. It might be better to machine an adapter that connects the motor shaft to any of various machine screws, which could be used to hold coil forms of differing sizes. A pair of cones—drilled along their axes—would allow the coil form to be easily centered on the machine screw. I don't have access to the necessary machine tools to make these accessories, so I just carefully center the bobbin, and hold it in place with nuts and washers.

Using the Winder

The coil winder is easy to use. Fasten the bobbin or coil form to the shaft, making sure that it is centered. You can check this by running the winder a few turns while watching for eccentricity. It's best to wind coils in the direction that allows the wire to feed onto the top of the coil (rotating away from you) so you can watch turns go on the form. Don't forget to zero the turns counter before you start! Make the first few turns of any winding by hand, to be sure the wire starts properly. Then slowly increase the motor speed until you have a comfortable build going. When the counter approaches the desired number of turns, slowly decrease the motor speed while keeping tension on the wire. Stop when the final turn count is reached. It's best to hold the wire 30 cm or so (about a foot) back from the coil. This allows the winding to accumulate in even rows. Fast-moving wire can burn your fingers, so use a glove or other protection.

Place the wire spool on a table or other support about a meter behind you. I usually hang the spool on a screwdriver that is clamped in a portable vise. It's important that the spool can rotate freely, especially if you are using fine-gauge wire.

Some Improvements

After using the coil winder a few times, I thought of several improvements that would make it more useful:

My turn counter increments regardless of the winding direction. If I reverse the direction to take off a few turns, the counter reads incorrectly. This could be solved by replacing the simple mechanical counter with an up-down counter—either mechanical or electronic—and by using an appropriate direction-sensing circuit. This circuit could simply be another photo interrupter, offset from the first one, including logic circuitry to sense the direction. (Some photo interrupters are made for just this purpose.—*Ed.*)

A set of different-sized machine screw shafts for various coil sizes, along with centering cones and the associated adapters to the motor shaft would be useful.

A foot-pedal speed control would keep both hands free to apply the winding. This could be nothing more than a pedal attached to a speed-control potentiometer.

Summary

The coil winder has given good service. It enabled me to wind a precise number of turns on the high-voltage transformer for the switching power supply. The convenience of a good coil winder makes it easy to rewind coils to change the number of turns, wire size or other parameters. □□

The Cheap Sweep

Here's an example of functional test equipment built from components in the junk box. Perhaps you have similar instruments hiding in yours!

By C. A. Hoover, K0VXM

Having acquired a pair of beautiful six-pole adjustable filters from an old cellular telephone, I set myself to the task of using them on 902 MHz. I didn't really want to abuse my solid-state final by pouring power into an unknown load. Yet, I had neither a signal generator nor a sweep generator of suitable characteristics. So, I came up with a sweeper using junk-box parts and equipment on hand.

The sweeper consists of a VCO, a continuous rotation pot, a stepper motor and a stepper motor driver (See Photo A). A diode detector, a suitable 12 V power supply and an oscilloscope with X-Y display complete the setup.

1945 E Phillips Ct
Merritt Island, FL 32952

Fig 1 is a block diagram of the system. Photo B shows the running system and Photo C shows the trace of the filter under test. Photo D is a full view of the entire test setup.

The VCO (again pulled from an old cellular telephone) tunes from 890 to 950 MHz and drives the device under test directly. Since the tune current is less than 1 mA, a pot of any value from 100 Ω to 100 kΩ will suffice. I used a 1 k pot.

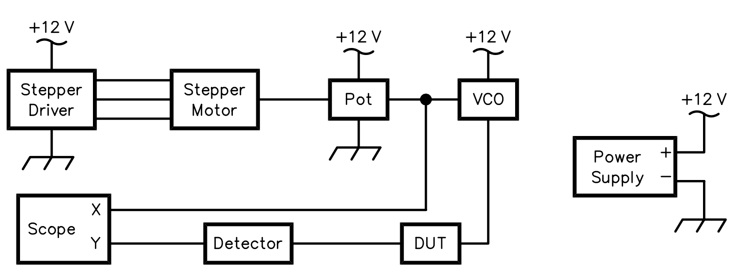


Fig 1—A block diagram of the test setup.

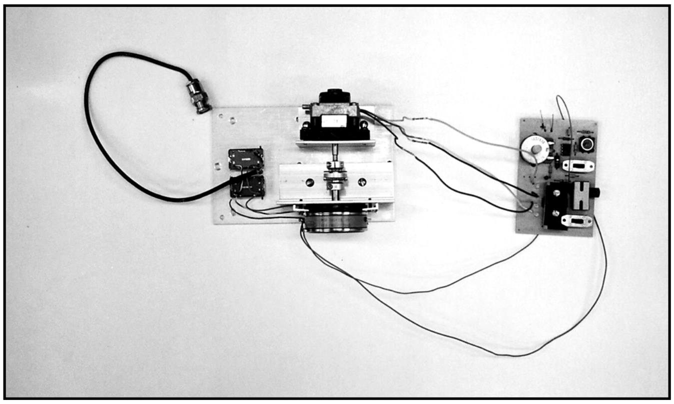


Photo A—The sweeper consists of a VCO, a continuous rotation pot, a stepper motor and a stepper motor driver.

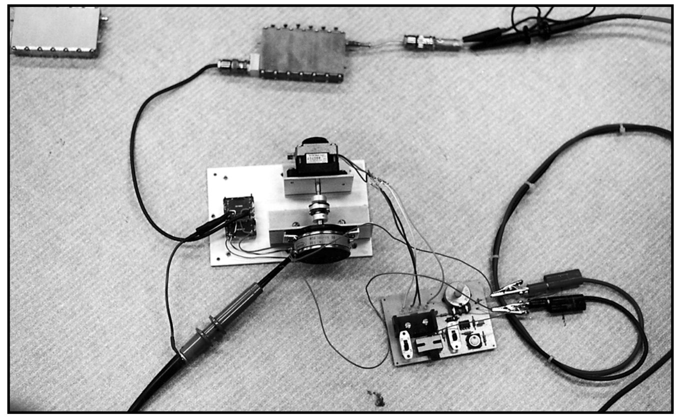


Photo B—The running system.

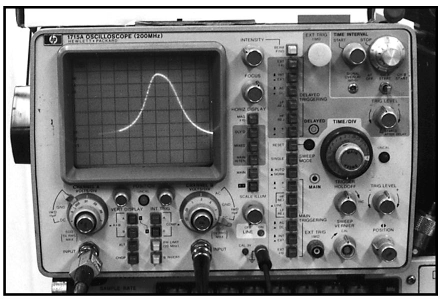


Photo C—A trace of the filter under test.

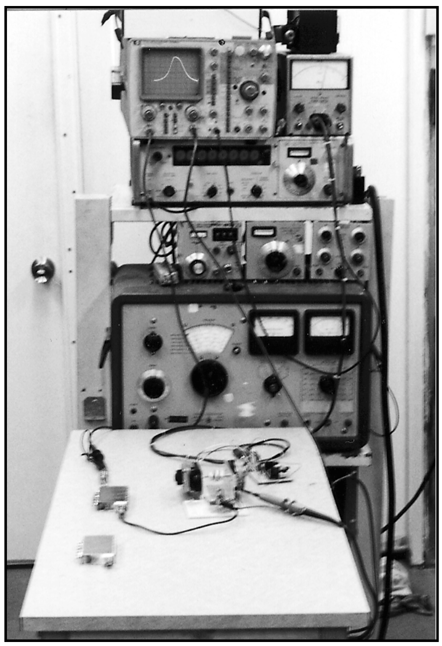


Photo D—A full view of the entire test setup.

The stepper-motor driver is a kit purchased from All Electronics.[1] Other suppliers sell similar kits; they are not hard to find. The stepper motor can be any motor that will drive the pot and mounts easily. The motor supplied with the driver kit is fine. When coupling the stepper motor and the pot together, be sure that the shafts rotate without binding.

The diode detector is homebrewed from *The ARRL Handbook*.[2] Keep the leads as short as possible. My detector was fabricated with composition resistors and disc capacitors; it is usable (barely) to 2 GHz.

The 12 V power supply should handle about 2 A and be well regulated. Be sure the voltage does not change when disconnecting or stopping the stepper motor. Any change in voltage will cause the VCO frequency to change and the trace on the 'scope to move.

The 'scope does not need to be at all fast. Any 'scope that has X-Y capability will do. Connect the output of the diode detector to the Y input and the wiper of the pot (which also goes to the tune pin of the VCO) to the X input. Once the trace is found, adjust the X and Y positions and the motor speed for best viewing. A motor speed of about 200 RPM seems to be optimum in my setup.

At this point one begins questioning the use of a stepper motor as opposed to other possible methods of swinging the VCO. Aside from having many stepper motors and a couple of stepper motor drivers on hand, I wanted the ability to run the system manually through its range. This makes for easy, relatively accurate frequency spotting.

I don't expect that anyone will duplicate my system exactly. However, my purpose is to demonstrate what can be done with a well-stocked junk box and a little ingenuity.

Notes

[1] All Electronics Corp, 14928 Oxnard St, PO Box 567, Van Nuys, CA 91411; tel 800-826-5432, 818-904-0524, fax 818-781-2653; e-mail **allcorp@allcorp.com**; URL **http://www.allcorp.com/**. Catalog #SMKIT-2.

[2] R. Dean Straw, N6BV, Ed. *The ARRL Handbook* (Newington: ARRL, 1998), Order No. 1816. ARRL publications are available from your local ARRL dealer or directly from the ARRL. Mail orders to Pub Sales Dept, ARRL, 225 Main St, Newington, CT 06111-1494. You can call us toll-free at tel 888-277-5289; fax your order to 860-594-0303; or send e-mail to **pubsales@arrl.org**. Check out the full ARRL publications line on the World Wide Web at **http://www.arrl.org/catalog**.

No-Tune Transverter-Spur Identification

No-tune transverters have encouraged many newcomers to the UHF and microwave bands. Yet a newcomer's receive system may be plagued by whistles and squeals. Here's how to find the culprit and a few hints to silence those birds.

By Michael McKay, W4AZR

This article shows how to identify spurs generated in a receiver lineup with one of the no-tune transverters that are now popular. Although there is nothing very new here, the analysis approach may help people just getting their feet wet in the UHF/Microwave game. The specific example is from my own 33 cm station.

I use a Down East Microwave SHF 902k transverter with a DEM 33LNA located at the 18-element loop Yagi. The 902k feeds a DEM 144-28DC, which in turn feeds my TS-940 HF transceiver via the transverter socket on its rear panel. The arrangement is shown in Fig 1. Notice that the TS-940 has been modified to transmit outside the ham bands. This permits its use as a 25 to 30-MHz tunable IF. The 902k uses an oscillator injection frequency of 759.000 MHz (derived from a 94.875 MHz crystal oscillator) and the DEM 144-28DC injection is at 118.000. The bottom line is that 25 MHz relates to both 143 MHz (118 + 25 = 143) and 902 MHz (759 + 118 + 25 = 902). When the TS-940 is at 30 MHz, the receive frequency is 907 MHz; in an ideal receiver one would hear nothing but white noise when tuning this range in absence of a signal on the antenna.

With a 50 Ω termination on the 902k input, I observed a loud carrier at a frequency of 25.500 MHz, another at 27.874 MHz and still another at 27.900 MHz. All had clean notes when tuned for "zero beat." The corresponding frequencies at the 2-meter level are 143.500 MHz, 145.874 MHz, and 145.900 MHz, respectively. I'll refer to these as spurs 1, 2 and 3. To understand the origins of these spurs, begin by listing the harmonics of the 2-meter oscillator and the base oscillator of the 902k as in Table 1 (to the 10th order).

Now calculate the sum and difference frequencies of the 2 meter frequency of spur 1 with each of the harmonics of the 2 meter oscillator. For example, 236.000 MHz plus and minus 143.500 MHz gives 379.500 MHz and 92.500 MHz. Then check to see if one of the results lies on or very near a frequency in the base-oscillator column. In this case, the sum frequency is the fourth harmonic of the base oscillator. This identifies the source of spur 1. In

2265 Windsor Dr
Merritt Island, FL 32952-5508

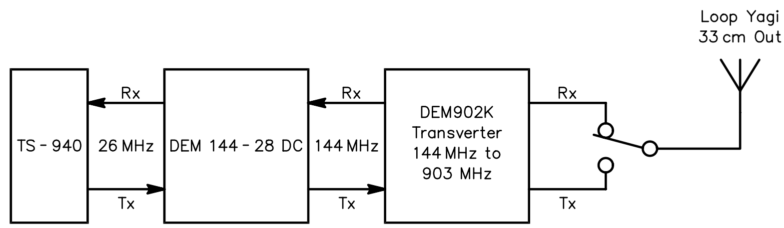
like fashion, spur 2 is caused by the sixth harmonic of the 2-meter oscillator mixing with the ninth harmonic of the base oscillator. In numbers, we have: 853.875 MHz equals 708.000 MHz plus 145.874 MHz.

Spur 3 is a bit different from the other two. It is the image response of the TS-940. Notice that the spur 3 frequency is 27.900 MHz. The first IF in the TS-940 is 45.05 MHz and the oscillator is on the high side of the signal. Thus, the image response is at 118.000 MHz (27.900 + 90.100 = 118.00). Although the signal is weak, this means the TS-940 hears the DEM 144-28DC oscillator. Because there is no crowding on 33 cm at this time, this is no problem, particularly since I (and the locals hereabout) operate on 903.100 MHz. If you're in the neighborhood, look for us Tuesday evenings at 7:30 local time.

So there will probably be spurs in your nice new shiny UHF setup—is this bad? The answer is no, because the spur signal can serve as a quick check of your receiver. At one time, I verified that the 94.875 MHz oscillator had started by listening to its fundamental on an ordinary home-entertainment FM receiver. Now I simply tune down to spur 1 at 26.500 MHz (902.500 MHz on 33 cm) and verify that all is well.

It's interesting that—after an hour and a half—the oscillator in the DEM 144-28DC sometimes develops a slight frequency drift at about a 1 Hz rate. This is not detectable when working FM, but it's a disaster on SSB. Someday I shall have to fix it, but the point is the problem was found by listening to the built-in spur.

The levels of spurs 1 and 2 can be reduced an S-unit or so by simply grasping the +13.8 V dc feed line to the 144-28DC tightly in hand. Since all the modules share an Astron RS-20M supply, it is probable that better decoupling within the modules would improve the matter. Although I have not tried it yet, I intend to use ferrite bead chokes on the internal +13.8 V wiring in conjunction with surface-mount bypass capacitors. I've found that MMIC spur levels wildly vary for the first five minutes, until the MMICs achieve thermal equilibrium, at which point measurements are generally reproducible.



Notes: 1. TS - 940 Input/Output on Back Panel
2. LNA at Antenna Not Shown

Fig 1

Table 1—Oscillator Harmonics

Order	*2 Meter Oscillator Harmonics (MHz)*	*Base Oscillator Harmonics (MHz)*
1	118.000	94.875
2	236.000	189.750
3	354.000	284.625
4	472.000	379.500
5	590.000	474.375
6	708.000	569.250
7	826.000	664.125
8	944.000	759.000
9	1062.000	853.875
10	1180.000	948.750

Yes, the identification of spurs can be automated by use of a computer and such programs are available. For newcomers, however, I believe the approach just described produces greater understanding. It's simple, you need only basic arithmetic and a scratchpad to get the answers.

After discharge from the WW2 Navy, Mike went back to college, received a BA (1947) and an MA (1949), both in physics. He then spent two years in TV-receiver circuit design. During the Korean War, Mike switched to defense electronics (for the next 38 years) working mostly on microwave radar guidance of weapon systems. Mike was named on 13 US patents in this period. First licensed in 1947 he has held W1QVV, W8ERL, W2GRS and now W4AZR. Current interests are weak-signal gear for UHF and above, a modest amount of RTTY and the 6 and 12-meter bands. □□

RF

By Zack Lau, W1VT

Building Solid-State Microwave Power Amplifiers

It's been said that there are two different approaches to building microwave projects. In Europe, they build everything from scratch, while in the US many people re-engineer and convert the somewhat plentiful supply of surplus equipment. I'm going to discuss an intermediate approach, taking surplus gear and using the parts to build gear essentially from scratch. The idea is to combine the best of both worlds: Get the performance and practicality of gear built and designed with amateur techniques *and* the low cost of surplus parts.

The reason for using surplus parts is obvious—microwave parts can be extremely expensive, particularly devices like power GaAs FETs. Some of the more-exotic parts, like 3 W, 10 GHz FETs, cost hundreds dollars—a bit too rich for most amateurs. Others, like absorptive rubber, may not cost a lot of money, but are a real hassle to buy because manufacturers aren't set up for $10 orders from individuals. Thus, disassembling a surplus amplifier can provide little parts that are a hassle to get, in addition to the obviously expensive pieces.

It isn't always practical to modify surplus equipment, however. The jump to the nearest amateur band may just be too far for some circuits, such as power splitters and combiners. On the other hand, the equipment may use tiny wires and single-layer capacitors that are easily destroyed when handled without specialized equipment. I've heard of circuit boards with plated traces that vaporize and become unsolderable.[1]

Those combiners, even when they don't work, can be quite useful. Often, they indicate the equipment's original operating frequency. The power-supply bypassing may be helpful too. Typically, $\lambda/4$ stubs or radials are used. You can gauge frequency by the size of the circuitry; as frequency increases, components get smaller.

Markings on transistors can also be useful. Many manufacturers use a four-digit number that signifies the frequency range. Thus, 5964 corresponds to 5.9 to 6.4 GHz. Often, there is a suffix that corresponds to the output power, in watts. Thus, 5964-3 is a 3-W device. These are impedance-matched transistors optimized for a narrow frequency range. The internal networks have a low-pass characteristic—they can often

225 Main St
Newington, CT 06111-1494
zlau@arrl.org

[1]Notes appear on page 60.

be retuned lower, but not higher, in frequency. Transistors with regular model numbers don't have specific frequency ranges, although they yield less gain at higher frequencies. In addition, gain and other characteristics seem to be less predictable at higher frequencies.

Of course, the best way to identify devices is with a data sheet from the manufacturer, but this isn't always practical. If you have a working amplifier, you can often get the device currents and voltages by making some measurements. Actually, the datasheet is just a starting point if you want to get as much power as possible out of the device. Typically, people run devices at 50% of the measured zero-gate-voltage drain current (I_{DSS}). Often, this current varies by 50% or more, so you need to select devices with the highest current for the output stage. Be careful when measuring I_{Dss}—the drain voltage must be reduced to avoid destroying the device. Otherwise, the transistor will dissipate far too much heat and burn up.

I strongly recommend you use a current-limited supply for each transistor you are worried about destroying. With proper current and voltage regulation you can protect a transistor from common accidents, such as shorting out the gate-bias supply. Active biasing protects transistors indirectly—with low-current transistors, the current-sensing resistor can act as a current-limiting device. With a power transistor, there often isn't enough voltage overhead to allow much current limiting. In addition, big power resistors are very inefficient. You may have 100 Ω of resistance in an LNA, while 1 Ω is more typical in a power application.

Contrary to popular belief, you don't always want a lot of ground plane on the topside of a double-sided circuit board. At microwave frequencies, it is quite possible for the top and bottom "ground planes" to differ. Fig 1 shows an excellent example using a high-gain PF0011 33-cm hybrid module. The top ground plane provided a feedback path resulting in an unstable amplifier. The cure was to add a couple of screws to block the feedback path, as shown in Fig 2. While a grounding zealot may argue that there wasn't enough grounding, a second version (shown in Fig 3) with no top ground foil between the input and output worked just was well and required less hardware. Admittedly, the problem is partially due to the high gain of the amplifier—I've measured 46 dB of gain! I've seen modules encased in a tin plated shield—undoubtedly to reduce unwanted feedback paths.

Instead, you really ought to design the ground paths—how, exactly, is the ground current going to go where you want it? There are three critical paths: to the transistor, the coax connections and the matching elements.

The first item is what often makes RF transistors so expensive. The package may be as expensive as the chip inside it! High-power transistor packages are usually fastened to the ground with screws. Mitsubishi warns against using thermal grease—instead specifying that you use a very flat surface (a surface finish of 32 µ

Fig 1—A microwave amplifier with complete topside foil—oscillates.

Fig 2—Four bolts added near the lower board center connect the ground planes—no oscillation.

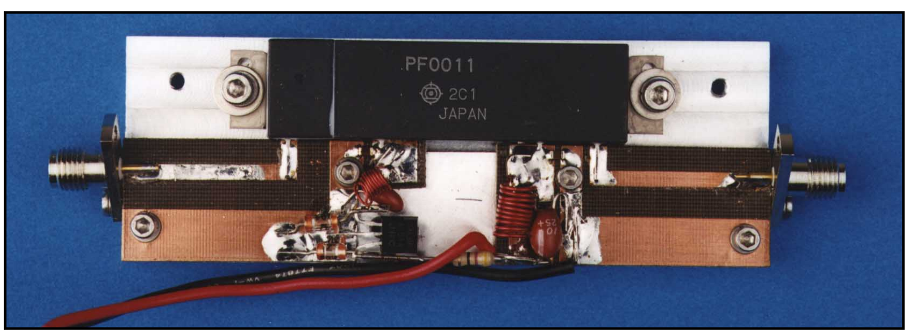


Fig 3—Without the four bolts, but with the top foil removed between input and output connections—no oscillation.

Table 1

Device	*Screw*	*Torque*
MGF K25M/K30M/K33M	#0 pan head	2.0-2.5 kg-cm
MGF-0905A	#2 pan head	2.5-3.0 kg-cm
MGF-C39V	#4 pan head	5.0-6.0 kg-cm

inch maximum with a camber of 0.0003 inch maximum).[2] They also recommend using a torque wrench to mount the device. Table 1 has some suggested torque settings.

If you study the current path at the transistor, a weak point may be the path between the grounded flange of the transistor and the ground plane of a circuit board. If you have a double-sided circuit board, what holds the ground plane against the heat-sink surface? The input and output leads of most transistors really aren't meant for that purpose; many are rather delicate strips of metal. Thus, some people use conductive epoxy to attach the circuit board to milled plates. This approach is very expensive because conductive epoxy has a limited shelf life. Everyone I've talked with recommends using all the epoxy at once. Its limited shelf life is even shorter once opened. We can more cheaply use screws to hold down the circuit board at key points; I've gotten this to work with 3 W, 10 GHz amplifiers. The best approach is to use an aluminum-backed circuit board, so the mounting surface *is* the circuit ground plane.

I'm most familiar with the Rogers version of this exotic circuit board, having purchased a sheet at a flea market nearly a decade ago. I only figured out how to use it just recently. The easiest problem is etching the material without damaging the aluminum. (A sodium persulfate etchant won't attack aluminum unless the proper catalyst is available, so it may not be necessary to mask off the aluminum.) According to a document on the Kepro Web page,[3] ordinary table salt can be used as a catalyst. Thus, handling the aluminum with sweaty hands may not be a good idea. The worst case may be a poor masking job—where a little etchant is trapped against the aluminum. The concentration of ions may be enough to start a reaction displacing copper from used etchant. Very good masking will be required with ferric chloride etchant—its reaction with aluminum is quite vigorous.

A tougher problem is 6061 aluminum alloy, which has a tensile strength of just 20 kpsi. While 6061-T6 with a 35 kpsi tensile strength is easy to mill, 20 kpsi 6061 is much softer and machines poorly. I ended up using a Sherline miniature milling machine and keeping the machining simple. I opted for slots across the full width circuit board, instead of the fancy pockets people normally cut.

When machining metal, it is essential to fasten the work securely, while minimizing distortion from the fasteners. Instead of using a vice, I make threaded mounting holes an important part of the design. Not only do these holes make the project easy to install in the final transmitter or transverter, but also they can be used during the machining process. As a bonus, they more closely duplicate the stresses found in the assembled unit, so unwanted warping is less likely.

A similar technique is to solder a machined brass block to the circuit board. This works well if the circuit board is just the right thickness—you might just need to cut the block to size and add tapped mounting holes. It's a good idea to mount the block to the chassis as well. This will allow heat transfer from the block to the chassis; there is usually enough thermal conductivity for small and medium power devices, up to a watt or two. Larger devices may need a more-direct connection to a heat sink, however.

The next problem areas are the coaxial connectors. How do you get good electrical contact from the ground plane of a circuit board to the connector? Again, soldering works well. I've used brass plates that are soldered to the ground plane. The connectors are then attached to the plates with screws. If the transistor mounting plate is a close fit, it may be necessary to bevel it to accommodate the solder fillet. For testing purposes, one often needs just the front and back plates—the side plates aren't needed until after the amplifier is working. They are useful for shielding the amplifier. With thick metal-backed circuit board, you need only bolt the connectors to the ground plane.

Matching elements can also be difficult to ground. Often, you don't know exactly where to place them, though most designs do end up with a lot of capacitance near the input and output of the active device. Appropriately placed screws normally work well, unless you guess wrong. A better solution is to use matching techniques that don't require actual ground connections, such as bits of copper foil. This works well at 10 GHz with 2.2 or 2.5 ε_r board, but the foil gets too big at 2.3 GHz. One solution is a high-dielectric-constant board, such as Rogers 6010, with a constant of 10. Then, small bits of copper foil (just 50 or 100 mils on a side) have a significant effect on the tuning. I've successfully tuned 3 W, 6 GHz IMFETs down to 2304 MHz on 30-mil-thick Rogers 6010 board.

Another approach is to make a thin metal plate that fits between the circuit board and the transistor mounting plate. This eliminates the need for an expensive milling machine. Sheet aluminum comes in a variety of sizes—if you can't find the exact thickness you need try combining two sheets. There may be a complication when you try to attach connectors with mounting screws—solid pieces of metal are generally required for drilling and tapping holes. Thus, it may be necessary to rotate the connector to obtain better connection points.

Notes

[1]Keith Erickson, K0KE, "Tuning of Microwave Stripline Amplifiers," *Proceedings of the Microwave Update '87*.

[2]Mitsubishi Technical Bulletin *Recommendations for Mounting High-Power GaAs FET Package Devices*, p 7.2.1.

[3]http://kepro.com/

Letters to the Editor

Of Fields, Near and Far, in "A Test for Ambient Noise"

◊ Several times, I have started to write a letter to *QEX* regarding Appendix 1 of the Sep/Oct '98 article "A Test for Ambient Noise." I am surprised this Appendix was published "as is," although I guess there is much misconception over what is meant by the far and near fields of an antenna. Some noted authors, including those in Jasik's *Antenna Handbook*, sure have not helped the matter.

Basically, the technique given in Appendix 1 is invalid. The distance $(2\,d^{\,2})/\lambda$ has nothing to do with the nature of the fields around a basic dipole or loop antenna. It is an arbitrary boundary (from ray optics) between the Fresnel and Fraunhofer regions. That is where the rays from an aperture are nearly focused—if I recall correctly, the phase error is less than $\lambda/8$.

In the case of phased arrays and reflector antennas, this distance is about where the main beam gain of the antenna is pretty close to its gain at an infinite distance (the far-field gain). The main beam gain of an aperture at less than the $(2\,d^{\,2})/\lambda$ distance is less than the far-field value. Hence, to compute the power density within the main beam (based on simple spreading loss) at distances less than $(2\,d^{\,2})\,/\lambda$, one has to reduce the far-field gain of the antenna.

In both the Fresnel and Fraunhofer regions, the field is generally a plane wave. In the Appendix, the author was referring to the case where there is *not* a plane wave, which is a totally different concept. Here, we are referring to the case where there is more than one E field and/or one H field around the antenna. In the case of a short dipole, there are two E fields—an E field normal to the axis of the dipole, an E field parallel to the axis of the dipole and an H field concentric about the dipole axis. The cross product of these E fields and H field is taken to form the Poynting vector, representing the propagation of energy. The E field parallel to the axis of the dipole contributes to a Poynting vector directed radially outward from the dipole axis; thus the power is radiated outward. The other E field will produce Poynting vectors that are opposite (in phase) on each side of the dipole, directed upward or downward along the axis of the dipole, canceling each other out. These fields do not contribute to the outward radiation of energy, and they are sometimes called "reactive fields."

Apparently, the intent of the Appendix was to outline a method of determining at what point the normal E field around a dipole or monopole would become insignificant compared to the E field parallel to the axis. Obviously, what is insignificant depends on your purpose—a level 10 or 20 dB down might be satisfactory. The best way I know to determine this distance is with a method-of-moments-based model, such as is used in *NEC*, *MININEC* or similar codes.

By the way, the distance used in the Fresnel-Fraunhofer boundary equation is the actual dimension of the aperture, not the effective area computed from Eq 7 in the Appendix. The effective area is often only 40% to 60% of the physical area for most reflector-type antennas. I wonder if *QEX* needs a tutorial on this subject and its impact on RADHAZ calculations?—*Lee Garlock, KD4RE, 3163 Plantation Pkwy, Fairfax, VA 22030*

◊ I think you've just given us one! Nevertheless, could not reality be described more simply? Try the following: When current flows in a wire, electric and magnetic fields are created around the wire. When the current ceases or reverses direction, the fields tend to collapse back onto the wire. Some of the fields' energy, propagating away from the wire at velocity *c*, escapes into free space.

It's interesting to note that as the operating frequency decreases, it becomes easier for the fields to collapse before they can radiate away. At frequencies in the audio range, for example, almost all field energy is returned to the antenna, making radiation efficiency very poor. Even so, the near-field intensity can be strong, hence the concern regarding human exposure to fields from antennas and power lines. The following is the author's response.—*Doug Smith, KF6DX,* QEX *Editor;* **dsmith@arrl.org**

◊ I am grateful for your discussion of near and far fields. Because most of us know an approximate value for our antenna gain, I supplied a method to estimate the edge of the far field, based on antenna gain. In the December 1987 issue of *QEX*, H. Paul Shuch, N6TX, published a different approach in his paper "Far-Field Fallacy." He saw the edge of the far field as the distance where the path loss is 14 dB greater than the gain of two antennas looking at each other. I will take two antenna examples and calculate the edge of the far field using my method and that of Mr. Shuch.

1a. Two half-wave dipoles are operating at 14 MHz. The edge of the far field from Appendix 1 is:

The wavelength (λ) at 14 MHz = 21.43 meters, the dipole gain is 2.15 dB over isotropic, or G = 1.64 as a power ratio. This gives the apparent antenna area of:

$$A = [G\,\lambda^2]\,/\,(4\pi)$$
$$= 1.64\;(\,21.43^2\,/\,(4 \times 3.14)$$
$$= 59.9\ \text{m}^2$$

The apparent antenna diameter squared (D^2) is:

$$D^2 = 4\,A\,/\,\pi$$
$$= 4 \times 59.9\,/\,3.14$$
$$= 76.3\ \text{m}^2$$

The edge of the far field is:

$$R = 2\,D^{\,2}\,/\,\lambda$$
$$= 2 \times 76.3\,/\,21.43$$
$$= 7.12\ \text{m or } 23.4\ \text{ft}$$

1b. The edge of far field from *R* equals the distance where path loss is 14 dB greater than the gain of two antennas:

The combined gain of two dipoles is 2 × 2.15 dB = 4.30 dB. The specified path loss is 4.30 + 14 = 18.30 dB. The path loss is:

$-27.56 + 20\log(f) + 20\log(R)$, or

$$20\log(R) = \text{path loss} + 27.53 - 20\log(f)$$
$$= 18.30 + 27.53 - 22.92$$
$$= 22.91\ \text{dB}$$

$R = 13.98$ m or 45.86 ft

Where

f = frequency, in megahertz

R = radius, in meters

2a. Basic Yagi with a gain of 7 dB above a dipole and a simple dipole at 14 MHz. The edge of the far field from Appendix 1:

The wavelength (λ) at 14 MHz = 21.43 meters. The Yagi gain is 9.15 dB over isotropic, or $G = 8.22$ as a power ratio. This gives the apparent antenna area of:

$$A = [G\,\lambda^2]\,/\,(4\pi)$$
$$= 8.22 \times 21.43^2\,/\,(4 \times 3.14)$$
$$= 300.4\ \text{m}^2$$

The apparent antenna diameter squared is:

$D^2 = 4\ A / \pi$
$= 4 \times 59.9 / 3.14$
$= 382.5\ m^2$

The edge of the far field is:

$R = 2\ D^2 / \lambda$
$= 2 \times 76.3 / 21.43$
$= 35.70$ m or 117.1 ft

2b. The edge of far field from *R* equals the distance where path loss is 14 dB greater than the gain of two antennas:

The combined gain of two antennas is 2.15 + 9.15 = 11.30 dB. The specified path loss is:

11.30 + 14 = 25.30 dB

path loss = –27.56 + 20 log (*f*) + 20 log (*R*), or

20 log (*R*) = path loss + 27.53 -20 log (*f*)
= 25.30 + 27.53 -22.92
=29.91 dB;

or R = 31.30 m or 102.7 ft.

The definition of "far field" in this application is somewhat academic. The goal is to find a generic method to calculate the distance between antennas where the experimental noise measurement results represent antennas a great distance from each other. One could test the experimental results by first measuring at one distance and then measuring again at twice the distance. Mr. Shuch suggested that we take his distance and double it for antenna gain measurements.—*Pete Lefferson, K4POB, 6101 7th Ave N, Saint Petersburg, FL 33710-7015;* **elecleff@cftnet.com**

Signals, Samples and Stuff: A DSP Tutorial

◊ I was just reading Part 4 of Doug Smith's DSP series in the Sep/Oct '98 *QEX*. I could not follow a step in the equations.

In the DFFT analysis, I think there is a typographical mistake in the second line of Eq 5. My calculations show that there should be a *k* term in the first exponent, and there is an *e* missing after the multiplication dot.

I did reach the same result, since the first *e* term reduces to 1, and the second term is W_n^{-k}.

Thanks for putting this series together. DSP articles in ham radio magazines are rare. We need more! *—Pete Wyckoff, KA3WCA, 1080 Taylorsville Rd, Washington Crossing, PA 18977, Member, Technical Staff, Personal Earth Station Hardware Engineering, Hughes Network Systems,*

◊ You are correct! That one slipped by us at the last moment, although the result of the derivation is accurate. Thanks for taking the time to write.—*Doug Smith, KF7DX,* QEX *Editor*

Educational Activity Award Nominations

◊ The ARRL offers four awards to ham radio instructors and recruiters:

- Herb S. Brier Instructor of the Year (a nonpaid volunteer instructor)
- Professional Instructor of the Year (a paid, non-state licensed, instructor)
- Professional Educator of the Year (a professional teacher)
- Excellence in Recruiting (hams who exemplify outstanding recruiting enthusiasm and technique).

Does someone you know (maybe you) deserve this recognition? Start the nomination process today. Access our Web page at **http://www.arrl.org/ead/award/** for information about the awards. You can download nomination forms at **http://www.arrl.org/ead/award/application.html** or call Jean at 860-594-0219 for a nomination form and information on requirements and deadlines. Send completed nomination forms to your Section Manager by January 31. A job well done should be rewarded.

The Tetrode Boards

◊ My article about tetrode power supplies ("Power and Protection for Modern Tetrodes," *QEX*, Oct 1997, pp 15-26) has led to a commercial product called THE TETRODE BOARDS. This product is a new solution for the control and protection of tetrode power amplifiers. The circuits will work with any transmitting tetrode for amateur power levels, in any power-supply grounding arrangement. Two small PC boards include regulated power supplies for the screen and control grids, screen and grid current protection, T/R sequencing, ALC and relay supplies—that's almost everything except the high-voltage supply and the tetrode!

The kit includes all the components for the PC boards, a comprehensive 32-page manual and full support from the designer. Experienced constructors can buy the bare boards and manual. A special mains transformer is also available, that connects directly to the boards and provides everything except the anode voltage.

For details, see **http://www.ifwtech.demon.co.uk/g3sek** or contact Down East Microwave Inc, tel 908-996-3584. Outside the USA, contact Ian White, G3SEK at the *Callbook* address, or e-mail **tetrode-boards@ifwtech.demon.co.uk**—*Ian White, G3SEK,* **ian@ifwtech .demon.co.uk**

Multiple-Octave Bidirectional Wire Antennas

◊ I received about 10 e-mails regarding my Jul/Aug '98 *QEX* article before I left IBM and cancelled my e-mail address there. Readers can now reach me at **W7SX@aol.com**.—*Bob Zavrel, W7SX/0*

Practical Hot-Guy-Wire Antennas for Ham Radio

◊ I've noticed an error in my Nov/Dec *QEX* article. In Figure 13 B (page 55) the vertical dipole strung alongside the tower shows the bottom of the dipole touching ground. While there is no connection dot there, I want to make it clear that the dipole end should be insulated from ground. In paragraph 2 of page 53, "1 /2" should be "λ/2."—*Grant Bingeman, KM5KG, 1908 Paris Ave, Plano, TX 75025;* **DrBingo@compuserve.com** □□

Next Issue in QEX

Among other features in the next issue of *QEX*, Rudy Severns, N6LF returns (!) as an author with some new and interesting ways of examining the properties of full-size vertical antennas with counterpoise radials. Rudy has put his real estate and timber to excellent use—wait 'til you see how he erected his 160-meter vertical using only a "small" amount of base loading. Modeling and practice converge neatly in this very engaging article.

Ken Beals, WK6F, puts the serial control port of his HF transceiver to good use over a 10 GHz remote-control link. Ken takes us over the system at the block diagram level, lightly, then provides some useful information about how to get it all into a box or two. Along the way, he discusses the general requirements of remotely controlled systems and illuminates the many decisions facing designers. Complete schematics and PC board layouts are available for readers serious about circumventing antenna restrictions and RF-exposure problems. □□

TAPR

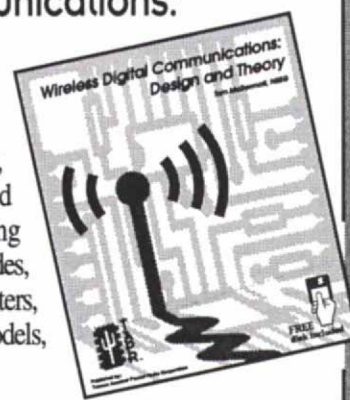
***Join* the effort in developing Spread Spectrum Communications for the amateur radio service.** *Join* TAPR and become part of the largest packet radio group in the world. TAPR is a non-profit amateur radio organization that develops new communications technology, provides useful/affordable kits, and promotes the advancement of the amateur art through publications, meetings, and standards. Membership includes a subscription to the *TAPR Packet Status Register* quarterly newsletter, which provides up-to-date news and user/technical information. Annual membership US/Canada/Mexico $20, and outside North America $25.

TAPR CD-ROM

Over 600 Megs of Data in ISO 9660 format. TAPR Software Library: 40 megs of software on BBSs, Satellites, Switches, TNCs, Terminals, TCP/IP, and more! 150Megs of APRS Software and Maps. RealAudio Files. Quicktime Movies. Mail Archives from TAPR's SIGs, and much, much more!

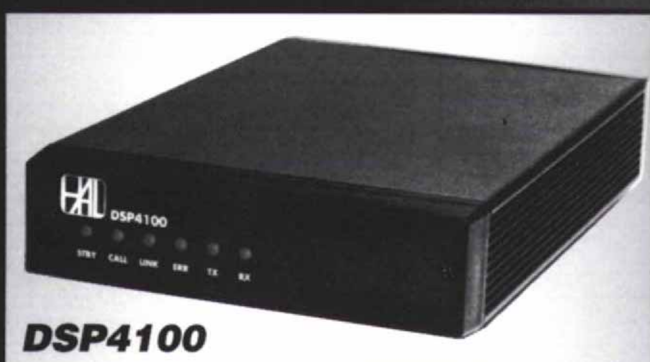
Wireless Digital Communications: Design and Theory

Finally a book covering a broad spectrum of wireless digital subjects in one place, written by Tom McDermott, N5EG. Topics include: DSP-based modem filters, forward-error-correcting codes, carrier transmission types, data codes, data slicers, clock recovery, matched filters, carrier recovery, propagation channel models, and much more! Includes a disk!



Tucson Amateur Packet Radio
8987-309 E. Tanque Verde Rd #337 • Tucson, Arizona • 85749-9399
Office: (940) 383-0000 • Fax: (940) 566-2544 • Internet: tapr@tapr.org www.tapr.org
Non-Profit Research and Development Corporation

DSP4100



HF Radio DSP Modem

STAND-ALONE

COMPACT

PORTABLE

HIGH PERFORMANCE

-- all the features you've been asking for in a CLOVER DSP Modem --

At long last, the proven HAL DSP Modem architecture, modes, and software are available for applications that cannot use plug-in PC cards. While the DSP4100 closely follows the concepts of the PCI-4000, now you get CLOVER-II and high-performance TOR, Pactor, and RTTY in a stand-alone DSP modem. Requiring only 0.25A from a 12V battery, the 2.75 lb DSP4100 will go anywhere you can take your LAP-TOP PC and transceiver. Software changes are easily made in the field. Just pick-up new software from HAL and upload it to the DSP4100 via the serial port for storage in non-volatile *FLASH* RAM. A 2nd RS-232 port is included for customized systems. Call HAL now for complete details.

HAL Communications Corp.
P.O. Box 365, Urbana, IL 61801 USA
Phone (217) 367-7373 Fax (217) 367-1701 E-Mail halcomm@prairienet.org

MasterCard VISA

New from Alpha Delta. **WOW!**

Why didn't someone think of this before?

ALPHA DELTA Models DH-1 and DH-2 Hydraulic Dampened Fold-Over Mast Fixtures for HF Verticals and Small VHF/UHF Ground Planes

Now one person can raise and lower an HF vertical in a simple 30 second operation!

- Easy antenna adjustments and maintenance.
- The vertical can be lowered out of view when not in use.
- Small VHF/UHF ground planes and discones can easily be raised.
- The Model DH-1 fold-over fixture includes a powder coated steel pedestal--$249.95 ea.
- The Model DH-2 excludes the pedestal and mounts on a 4x4 wood post or sturdy wood fence post-$229.95 ea.
- Stainless steel hardware, aircraft grade aluminum and "post-hole" type mounting with premix concrete. Designed for verticals up to 29 ft. and 25 lbs. weight. DH series fold-over fixtures are pre-assembled and include a hydraulic dampener for safe operation. User supplied mast tubes shown in photos.

Now You See It—

Now You Don't!

Model DH-1 in action!

Toll free order line (888) 302-8777 ***(Add shipping & handling, exports quoted)***

ALPHA DELTA COMMUNICATIONS, INC.

P.O. Box 620, Manchester, KY 40962 • (606) 598-2029 • fax (606) 598-4413

Alpha Delta —*Compelling You Into the 21st Century*

MasterCard VISA