A VHF and Up Operator's Discussion of Useful DSP Software and Hardware

By Roger Rehr, W3SZ

Abstract: DSP Signal Processing Software and Hardware have proliferated and progressed tremendously since my presentation on this topic at the 2002 MUD / Eastern VHF/UHF Conference. They have become an integral part of the VHF/UHF/Microwave station for many of us. This talk will review several of the most useful software and hardware platforms including Linrad, Winrad, Power-SDR, Rocky, DttSP, SpectraVue, the WSE hardware series of SM5BSZ, the Flex 5000, the SDR-1000, the SDR-14, the SDR-IQ, the SoftRock, the Time Machine, and the HPSDR/TAPR hardware series.. The goal of the presentation is to give the audience a good understanding of the practical aspects of how to set up an effective station making the best use of these DSP tools for VHF/UHF/Microwave weak signal and contest work, using the W3SZ station as an example.

In 2002 when the first version of this paper was written, there were relatively few useful software DSP packages available, and there was even less "off-the-shelf" hardware designed for use with this software.

Of the software discussed at that time, none of the "First Generation" packages such as DSP-Blaster are still worth discussing. Linrad, a "Second Generation" package, was at that point in a relatively early stage of development. It has blossomed into a full-featured if challenging receive-only package that remains as of August 9, 2007 the standard to beat in terms of performance. It is now available in both Linux and Windows versions. It will be discussed in detail in this document as there is much new to report. The DSP-10 was a very innovative and exciting project, and stirred up a lot of enthusiasm when it was presented in 1999. It still has a small but active users group, but subsequent developments have leapfrogged it and it won't be discussed further here. The WSJT modes designed by Joe Taylor K1JT are a subject worthy of several lectures in themselves and therefore will not be discussed here.

I will also discuss newer software that was not discussed previously including Winrad, a Windows-only, receive-only package by Alberto diBene I2PHD that is similar to Linrad but with a much easier-to-use user interface and PowerSDR, a Windows-only [at present] software offering from Flex-Radio that is available for free download. PowerSDR is specifically designed for both transmitting and receiving with the Flex-Radio SDR-1000 and Flex-5000, but is also usable with a variety of hardware for receive-only. I will also discuss Rocky, a Windows-only receive program by Alex Shovkoplyas, VE3NEA; DttSP, by Frank Brickle AB2KT and Bob McGwier N4HY, which is the "DSP-engine" behind the Flex-Radios and which in addition works well for receive-only with a variety of hardware; and SpectraVue, which is the software by Moe Wheatley AE4JY that is supplied by RFSpace with its SDR-14 and SDR-IQ hardware.

In terms of hardware, I will discuss the Weak Signal Equipment (WSE) hardware of Leif Asbrink, SM5BSZ; the Flex-Radio Flex-5000 and SDR-1000; the SDR-14 and the SDR-IQ of RFSpace, Inc.; the SoftRock of Tony Parks KB9YIG; the Time Machine by Expanded Spectrum Systems; and the HPSDR/TAPR Atlas, Ozymandias, and Janus. I will briefly mention the Universal Software Radio Peripheral [USRP] by Matt Ettus. This unit works with GNU Radio, a free, open-source software framework for the creation of software defined radios.

The purpose of this document will be to discuss where such items may be helpful to VHF/UHF/Microwave operations, and which pieces of software and hardware are best suited to each task in this sphere. I will primarily discuss the use of this 'equipment' for terrestrial, contest-style operations, but will briefly discuss their use for EME, which is the ultimate in weak signal operations. Because Linrad's very steep learning curve keeps some from taking advantage of its extreme usefulness and versatility, I will spend some time discussing how to set up Linrad, and describing some of the details of its user interface.

I. What Do DSP Techniques and Software Defined Radios Bring to VHF/UHF/Microwave Contest Operations?

During the times between contests, weak-signal activity on the VHF/UHF/Microwave bands tends to be sparse. As a result, because most of us have very limited ‘free time’ in which to indulge in Amateur Radio activities, many of us limit our on-the-air activities to contests. This is why contest operation will be the focus of this discussion.

The VHF/UHF/Microwave contest environment is fundamentally different in many ways from the HF contest environment. Six and Two meters are similar in many ways to the HF bands, although less crowded and with fewer bone-crushing signals. On Six and Two, as on HF, a DSP radio’s major advantage over a conventional radio lies in the DSP radio’s extremely effective noise blanking. DSP radios can eliminate deafening pulse-type noise far more effectively than conventional noise blankers. Linrad has been the leader in this respect. The situation changes on the higher bands, where pulse noise is generally not such a problem. Both 222 and 432 MHz have far fewer signals present than the HF bands and the lower VHF bands, and 222 and 432 MHz have a lesser likelihood of random contacts primarily due to the smaller population of signals present as well as the greater directionality of the antenna arrays commonly used on these frequencies and the general lack of multi-hop propagation. As the frequency rises to 902 MHz, 1296 MHz, 2304 MHz, 3.4 GHz, 5.6 GHz, 10 GHz, 24 GHz, and beyond the differences from the HF contest environment become extreme. Due to the increasingly sparse signal population, the increasingly extreme directionality of the antenna arrays used, the increasingly large frequency uncertainty, and the usual lack of forms of propagation permitting very long distance communication, not only do the chances of a random contact approach zero as one moves to the upper ranges of the listed frequencies, but even making a coordinated contact becomes a challenge. It is particularly on these UHF and microwave frequencies where using a DSP radio [SDR, or Software Defined Radio] totally changes the mode of operation and makes what was extremely challenging and tedious a relatively simple and efficient task, easily accomplished in a few seconds, or a couple of minutes at most..

The ability of DSP software such as Linrad to pick out extremely weak signals hidden in a barrage of much stronger noise is well known. This ability is not really needed for most contest work [at least above 144 MHz], but it does help in completing those long-haul very-weak-signal contacts that would otherwise be impossible. At my station, this accounts for less than 1% of all contest QSOs. So why do I always have multiple instances of Linrad running during a contest?

There are at least three reasons to do this. First of all, it is important to know what is going on across the band on all of the lower bands at all times during a contest if one is going to avoid missing a band opening due to enhanced propagation, or a new station possibly in a much-needed grid square popping up on one band while you are operating on another. Having an always-on SDR on each of the four lower bands accomplishes this. You can see what is happening on four bands while running stations on just one of them.

Secondly, one needs to know what is happening on the band on which one is operating. Have new stations appeared in another part of the band that might provide new multipliers? Is the pattern of activity suddenly ‘different’ than it was moments ago, suggesting that a band opening due to enhanced propagation is starting [or ending]? Is the level of activity declining/changing so that one has reached the point of diminishing returns on that band so that a band change is in order? Having an SDR watch over an extended frequency spectrum on the band on which one is operating will help to answer these questions.

Thirdly, when operating the microwaves, there are a number of variables that need to be examined and managed to efficiently make contact with the station at the other end of the potential QSO. The more efficiently these variables are managed, the more completed QSOs will be made in a given time period, and contacts which might not have been completed can be successfully added to the log. These variables in need of efficient management and coordination are time, frequency, and beam heading. Time can be easily managed by clearly communicating the time constraints to be used, via the liaison channel, before the microwave contact is attempted and by accurate timekeeping at both ends [e.g., "I will transmit for the first 30 seconds of each minute and quit if nothing is heard in the first 5 minutes"]. Even with just beam heading and frequency as variables, establishing contact can be challenging. By using DSP software/hardware to show all frequencies within approximately 150-200 kHz of the intended frequency, it is unlikely that the other station's signal will be missed if it is present. So the only variable left to address is beam heading, and the beam/dish can be rocked back and forth while looking carefully for a signal to appear on the DSP spectrum/waterfall. Using this technique, establishing and completing a contact with a station that is potentially loud enough to be copied is a simple matter. If you can work him, you will see him on the SDR bandscope. And if he is too weak to work, you may still see him on the SDR bandscope.

II. What arrangement do I use at W3SZ to accomplish this?

A brief description of my station is necessary in order to show how I have integrated these techniques into my operations.

My station is set up to be used by a single operator. It covers all bands from 50 MHz through 24 GHz. All band switching is computer controlled via a parallel port by the logging software, RoverLog. The main operating position is an FT1000MP Mk V that is switched to the appropriate transverter for the band in use by RoverLog, which controls a homebrew controller board that was designed by Steve Kerns N3FTI and made as a PackRats project kit several years ago. I modified it slightly so that it would cover up to 24 GHz. This board also takes care of T/R sequencing operations and powering on and off the appropriate receive preamplifiers and transmit amplifiers.

The operator sits directly in front of the FT1000. Immediately on his right is a computer running RoverLog and a separate homebrew antenna controller program that will allow either keyboard-entered heading data or take heading data from RoverLog. Immediately on his left is a computer running Linrad for Windows. Linrad constantly displays a 190 KHz-wide bandscope for the band in operation. Linrad is fed by an RF Space SDR-IQ that is set to operate in the 28 MHz band, and which is fed by the same transverter signal that is feeding the receive portion of the FT1000. The center frequency of the SDR-IQ is controlled by Linrad. Also running on that computer is PowerSDR, which is used to control an SDR1000 that is used as the IF rig for the separate 144 MHz, 222 MHz, and 432 MHz liaison transverters that are used for talkback during microwave contacts. Currently the PowerSDR software and SDR1000 are used with the HPSDR/TAPR Atlas/Ozymandias/Janus PIO and soundcard replacement hardware.

All receive audio is fed to a New Communications Solutions NCS-3230 Multi-Rx so that any/all receivers can be heard through the Heil Pro Set Quiet Phone Headset. Transmit audio is fed to a New Communications Solutions NCS-3240 MultiSwitcher, which is used to select microphone or soundcard input for easy switching between voice and digital modes, and to provide optical isolation in the transmit audio path. An array of two foot pedals is used to control the FT1000 and liaison transceivers. Stepping on the right foot pedal puts the FT1000 and its associated transverters and amplifiers into transmit mode and sends the microphone audio to the FT1000. Stepping on the left foot pedal puts the SDR1000 liaison transceiver and its associated transverters and amplifiers into transmit mode and sends the microphone audio to the SDR1000 [via the Atlas/Ozy/Janus combo].

Mouse-clicking on a signal on the Linrad bandscope and then typing "Q" on the Linrad keyboard will cause the Linrad software to send a signal to the FT1000 that places the FT1000 on the frequency of the signal being received on the Linrad software.

So to make a microwave contact the sequence is generally:

1. A station is worked for example on 144-432 MHz using the FT1000.
2. That station is asked to attempt contacts on 903 MHz and above.
3. The SDR-1000 liaison is moved to the agreed-upon liaison frequency
4. Using the liaison frequency, transmit sequence timing and microwave frequency are agreed upon. The other station is asked to transmit "first".
5. The Linrad bandscope is centered on the agreed-upon frequency, e.g. 903.100
6. The other station starts transmitting and his signal is seen within a second or two on the Linrad bandscope.
7. His signal on the Linrad bandscope is mouse-left-clicked and the FT1000 immediately goes to that frequency.
8. If he is not already doing so, the other station is instructed to begin the exchange.
9. When the other station is done transmitting, he is answered, using the FT1000 which is on his frequency.
10. The other station is instructed, either directly or via the liaison frequency, to move to the next microwave band and the process is repeated until all bands are complete. The time per band required to complete the contact is often 30 seconds or less. No time is wasted 'looking' for the other guy. If he is transmitting, he is seen immediately. Signals which are too weak to be copied are often still easily seen.
11. A single keystroke in RoverLog puts the FT1000 on the next microwave band to be attempted and the process is repeated.

If the other station is not seen on the bandscope on a particular band, he is instructed to keep transmitting and the antenna heading is varied in small increments while the bandscope is inspected for appearance of a signal. If he is still not seen, he is instructed to move his antenna heading by small increments while the bandscope is watched for appearance of his signal. It is rare to have to move the center frequency of the bandscope to find someone because his frequency is outside the frequency range of the bandscope [has happened on 24 GHz with a rover].

This process works efficiently ONLY because the bandscope allows immediate identification of the other station, as soon as he starts transmitting anywhere within a 150-200 kHz window. There is no time wasted because the other station is "off frequency". Incidentally, W3SZ is GPS-locked on 903 MHz and 2.3-24 GHz.

The physical layout of W3SZ is such that to the right of the computer screen displaying RoverLog and the antenna rotor control program mentioned above is another computer screen that is divided into 4 quadrants. Each quadrant displays an always-on Linrad bandscope for one of the four lower VHF bands, so that a bandscope for each of 50, 144, 222, and 432 MHz is ALWAYS displayed. That way, the appearance of new stations or a band opening on any of these bands can be immediately recognized. If a station of interest is seen on any of these bandscopes, a mouse-click on that signal will put the audio from that signal into the headphones. If this audio indicates that station is worth contacting, typing "Q" will send the frequency information for the received station over the network to the FT1000 so that it will be put on the proper frequency to contact that station. The station can be quickly worked and then operations on the band previously in use resumed, if desired. Each of these Linrad bandscopes is fed via a different front end. For 50 MHz I feed Linrad with an SDR-14 centered at 28.125 MHz. For 144 MHz I feed Linrad with an SDR-IQ centered at 28.200 MHz. For 222 and 432 MHz I use a Delta44 soundcard [2 bands per card] fed by two simple 28 MHz SoftRock receivers [one for each band, both centered on 28.100]. Each of these four receivers is fed by the appropriate 28 MHz IF signal coming from the transverter for that band.

I use Linrad for the bandscopes because I can run 4 instances of it at the same time, and because each bandscope can put my FT1000 MP on the frequency of interest, if an interesting signal is heard.

I use the SDR-14 and SDR-IQ for the 50 and 144 MHz bandscopes because they have better image rejection and strong signal handling capability than the SoftRocks and The Time Machines, and do not have the problem of LO leakage. Additionally, the SoftRocks for 28 MHz are insensitive, and require a preamplifier. I have used both the DX Engineering RPA-1 preamplifier and KK7B Rick Campbell's R2Pro 28 MHz LNA from Kanga USA for this purpose. I use SoftRocks [with 28 MHz preamplifiers as just mentioned] on 222 and 432 MHz because they are cheap, and because I have not yet figured out how to run 4 SDR-IQs on one computer, and because of speed limitations of the USB bus and computer hardware that would be an issue with more than 2 SDR-IQ's running on one computer.

To get a start experimenting with SDR radios and DSP processing most cheaply and easily, get a SoftRock for the hardware and for software start with PowerSDR, Rocky or Linrad for Windows, using the default parameters for Linrad. Unfortunately, the 28 MHz SoftRock is no longer available so if you take this approach you'll be experimenting on the HF bands...

To get up and running most quickly, effectively, and easily, get a Flex-5000 (or an SDR-1000 if you can find one) and PowerSDR.

For a middle of the road, receive-only approach, get an SDR-IQ and use it with Linrad for Windows or Winrad.

The remainder of this paper will discuss the various software and hardware products in more detail. Please refer to Tables I and II [Software Product Matrix] and Table III [Hardware Product Matrix] as well as the text below.

III. SDR Software (See also Tables I and II)

A. Linrad

For me, Linrad has turned out to be, as of the time that I am writing this [August 8, 3007] the best software solution for weak signal work. But my Flex-5000 is [I hope] soon to arrive, and the combination of that hardware and PowerSDR may prove hard to beat. Linrad provides an excellent bandscope that is bandwidth-limited only by the hardware that you are using with it. When I use Linrad with the WSE boxes for EME, I have a bandscope bandwidth of 96 kHz. When I use Linrad with the SDR-14 or SDR-IQ for terrestrial VHF and microwave work, I have a bandscope bandwidth of 150-190 kHz, depending upon exactly how I have the software configured. I can run Linrad either under Linux or Windows. Currently for contesting I am using it with Windows because I run it on the same computer as I am running PowerSDR. Linrad is reliable, and rarely crashes or locks up. It is a receive-only program, but I have modified it so that it will control the frequency of my FT1000MP, so that I have essentially a Linrad transceiver. When used with the WSE hardware, Linrad does automatic dual-polarity receive, always receiving the signals from both the horizontal and the vertical elements of my X-polarity yagis. Linrad will constantly calculate the optimal receive polarization angle from the incoming H and V signals, and combine the incoming signals so that this optimal angle is the receive polarization angle used. So you don't have to worry about receive polarization angle. You have eliminated a variable from the EME receive equation! For terrestrial work with the SDR-14 or SDR-IQ as the signal source, there is of course only one receive polarity. As noted above, I also use Linrad with SoftRock V7.0, the SDR-14, and the SDR-IQ, for 'always on' bandscopes for the 50, 144, 222, and 432 MHz bands.

Getting started with Linrad can appear quite daunting to the newcomer, but it is really not so hard. Leif has made reasonable choices for the default parameters for Linrad, and so you can really just leave the parameters as he initialized them to get started. You will then just have to select the input source [soundcard or SDR-14 or SDR-IQ] and make a few choices and you can be up and running. Because I have heard repeatedly from many people that they are frightened off when starting Linrad for the first time, I will here go into how to set up Linrad in detail here.

Download the Linrad for Windows zip file from the Linrad Home Page: http://www.sm5bsz.com/linuxdsp/linrad.htm. The correct file will have a name something like:

wlr2-35.zip . Unzip it into a directory [e.g. C:\linrad\]. Then double-click on "linrad.exe" to start it. When you start Linrad for the first time, it will say:

Setup file par_userint missing.

Use W to create a new par_userint file after setup.

Press S for setup routines

Any other key to exit.

Then press enter

You need to type "S", and then hit the ENTER key. You will then see:

Enter font scale (1 to 5, then press Enter):

and you should type a number between 1 and 5. I use "1". Linrad will then ask you for the parallel port address [parport address]:

Parport address (lpt1=888, none=0):

If you are not using the WSE boxes and controlling them with Linrad, you should type "0". Otherwise, type the parallel port address. On some of my machines this is 888; on the 64 bit Core 2 Duos, it is 32768 or 33792 depending on which port I am using. If you specify a parport address, then Linrad will then ask you for the parport read pin:

Parport read pin (ACK=10):

Say "10" if you have wired your WSE hardware status line to pin 10 of the 25 pin d-sub.

You will then be asked to specify the percentage of the screen width to use:

Percentage of screen width to use (33 to 100):

say "100" to get started. You will then be asked to specify the percentage of the screen height to use:

Percentage of screen height to use (33 to 100):

Again, specify "100" to get started.
You will then be presented with the main Linrad menu. Type "w" to save your parameter choices. This is a good time to know that you can exit Linrad by hitting the 'Esc' key. If you later want to change any

of the parameters that you just set, type “S” from this main menu screen. Next you will likely want to set up Linrad to work with your soundcard[s] or SDR-14 input. If you haven’t set this up before, and if you have no parameter files in your Linrad directory, then you can just make a mode selection [e.g. “A” for weak signal CW, or “D” for SSB and Linrad will automatically take you through the necessary setup screens. Or, you can type “U” and enter the necessary A/D and D/A setup routines this way. I would suggest typing “U”. If you have an SDR-14 or SDR-IQ connected via a USB port to your computer and if you have the necessary drivers for it installed, you will be asked if you want to use the SDR-14 for input. If you do, say ‘Y’, and you will be asked how you want to set the SDR-14 / SDR-IQ decimation factors. A reasonable way to set them is as follows:

M_CIC2 [14]
M_CIC5 [8]
M_RCF [4]

This should give you a received bandwidth of about 150 kHz.

If you are asked if you want to use and SDR (-14 or –IQ) and you say ‘N’, you will be presented with the first of several soundcard selection screens. You will see a listing of your soundcards and below that list will be the request “Select (first) device for Rx input by line number:” In my case, I use the M-Audio Delta 44 1 and 2 channels for input and so I select “2”, for “M-Audio Delta 44 1 / 2” and hit “ENTER”. Then I am asked “Do you need more channels from the same soundcard? (Y/N)”. I type “N”. If I were doing dual-polarity receive EME with the Delta44 I would still type “N” because the Delta 44 multi device driver supports 4 channels. Then I am asked to enter “Sampling speed (Hz):” and I type “96000” and hit “ENTER”, and on the same screen I am then asked to enter “No of bits (16/24):” and I type “24” and again hit “ENTER”. The I am asked to “Select a radio interface and given four choices: [1] one rx channel (normal audio), [2] one rx channel I/Q (direct conversion rx), [3] two rx channels (adaptive polarization/phasing), [4] two rx channels, I/Q. I choose [2] for terrestrial work and [4] for EME, so here I choose [2]. The next screen asks me to “Select device for Rx output” and I select [0] Realtek HD Audio output, my motherboard sound, and again hit “ENTER”. I type enter again, and then I am back to the main screen and I type ‘W’ to save my choices. Then I type ‘D’ to go to SSB mode and just hit ‘ENTER’ at each of the parameter screens to accept the default values. Then the receive display screen appears as Linrad enters receive mode. I click on a signal on the main waterfall or spectrum at the top of the screen and sound should begin coming out of the speaker. Note that the setup screens you will see if you run Linrad in Linux are slightly different, particularly when you are setting up the soundcards.

As I write this, Linrad-02.35 is the most recent version, and my comments will apply to this version. Note that Linrad has context-sensitive help screens. If you place the mouse cursor over a graphical control, control box, or text field and press the ‘F1’ key, context-sensitive help will pop up. You can press any key to get back to the Main Receiver screen from the help screen. If you place the mouse cursor over some “empty space” and press ‘F1’ you will see all of the control fields highlighted. You can exit Linrad at any time by pressing the ‘escape’ key. When the discussion below refers to “clicking” or “dragging” with the mouse, a **left** click is implied.

When in receive mode, Linrad has several windows on the screen at all times. This is shown in figure 1; refer to this figure throughout this discussion. This figure is available on the web at http://www.nitehawk.com/w3sz/mud2007-figure1.gif You can resize the individual windows or move each of them by clicking and dragging their edges with the mouse. Below the frequency scale at the top of the screen and running across most of it from left to right is the main waterfall display, showing signal intensity as a function of frequency horizontally, and as a function of time, vertically. Earlier times are nearer the bottom, most recent times at the top. Decimal minutes are displayed along the left vertical axis. Below the main waterfall display, covering the same frequency span and having the same width, is the real-time main spectrum display. Signal strength is the vertical axis and frequency is the horizontal axis, corresponding to the same locations on the waterfall and the frequency calibration at the top of the graph. The little up/down arrows at the bottom left and middle right of this display allow you to adjust the range and center point (baseline), respectively, of the spectrum amplitudes displayed, so that the signals are the right size for best viewing, and centered vertically as you wish on the display. The bandwidth of the main waterfall and spectrum displays can be set by clicking on the little arrows on the top of this display at its right and left ends so you can either view the maximum-possible received bandwidth [96 kHz with the Delta 44 Soundcard and the WSE hardware or up to 190 kHz with the SDR-14], or zoom in to expand a very small frequency interval and examine the details of the signal being received in that small frequency range.

At the bottom-left corner of the main waterfall display, above the main spectrum display, is a box to set (by clicking on the box and then typing in the desired values) the number of averages per line of the main waterfall display. I usually set this to something between 8 and 24, depending on how fast I want the waterfall to scroll. Just below this, at the top-left corner of the main spectrum display, is a box to set (by clicking on the box and then typing in the desired values) the number of [FFT1] averages per displayed point of the main spectrum display. I usually set this at 16. At the bottom-right of the main waterfall display are two more small boxes. The bottom box sets the zero point of the waterfall display and the top box sets the gain of the waterfall display. I usually set the gain to 1 and the zero point at approximately the noise floor of the displayed spectrum [here 20].

Clicking on a signal on either the main waterfall or main spectrum display will cause that signal to be processed and sent to the speaker. It will also cause the signal to be displayed on the high-resolution spectrum display and in the baseband [filter] display, where the frequency scales are expanded. If you are using Linrad just as a bandscope, you can ignore these displays. The high resolution display is easily recognizable because of its red horizontal calibration lines. In figure one it is on the left side of the screen, just below the main spectrum display. As noted above, by clicking with the mouse cursor at any point on the main waterfall or main spectrum display, you cause that portion of the spectrum to be placed in the high resolution spectrum box and DSP-processed. That is, that portion of the spectrum is DSP-filtered, noise-blanked, and converted to audio frequency so that it appears in your headphones or on your speakers. It is point and click receiving. To fine tune, you click on the peak in the high resolution spectrum, if need be, to touch up the tuning. You may also use the wheel on a wheel-mouse to “fine-tune” (a depressed wheel will change the fine-tune step size). On the high resolution display, there is excellent resolution of the signals. If you are using Linrad in dual polarity receive mode, on the high resolution display there will be two signals seen; a larger, green peak and a smaller, purple peak. The larger, green peak represents the selected polarization component of the received signal, and the purple peak represents the smaller, orthogonal polarization component. Optimal polarization matching of the received signal is achieved when the purple signal is minimized and the green signal maximized. At the bottom-left corner of the high resolution display are two small boxes. The yellow one on the left is to control the “dumb” noise blanker, and the light blue one on the right is to control the “smart” noise blanker. By clicking on each of these boxes you can set the mode by which the levels of these digital noise blankers are adjusted. You have your choice of [-] (no noise blanker), [A]utomatic, or [M]anual for each blanker. Automatic means that the blanker level follows the noise floor automatically, but the operator is responsible for setting the level above the noise floor in a way that fits the hardware he is using. Manual means that the blanker level is fixed. The “smart” blanker is only available if you have calibrated Linrad with a pulser, or put the calibration files dsp_ssb_corr, dsp_ssb_iqcorr, etc. available from Leif SM5BSZ’s website into the Linrad directory. The tiny “o” at the right bottom of the high resolution display turns on the oscilloscope function that shows the time domain signals at the inputs to the summation devices just before the second FFT is performed. The signals are presented showing first the real power spectrum of the signal, and then the real and imaginary components of each polarity of both the weak and strong components of the signal, giving a total of 9 different 'oscilloscope' tracings for each time point if you are using dual-polarity receive. With such a display you can really tell what the blankers are doing and gain lots of other useful information. This is explained in detail on Leif's website, e.g. http://nitehawk.com/sm5bsz/linuxdsp/timf2/timf2.htm gives an example of what can be done with this display.

Immediately to the right of the high resolution display, beneath the main spectrum display, is the baseband display. This box is easily recognized by its green horizontal calibration lines and the yellow "inverted U" at its center. The baseband display has its own waterfall on top and spectrum display below. The relative sizes of these two displays can be adjusted by pulling their common border one way or the other with the mouse. Your selected signal should be nicely centered in the baseband display. If not, click on it again in the high resolution spectrum to center it. The yellow line and 'hump' or inverted 'U' on the baseband display show the filter center frequency, bandwidth, and shape factor in graphical form. If you want a different filter bandwidth or shape factor, you just take the mouse over to the baseband display, and drag the yellow filter curve wider or narrower, and the filter adjusts graphically. There are several additional controls in the baseband window. There is a red vertical bar at the left of the window, superimposed over a vertical column of digits starting at 0 at the bottom and increasing to typically 70 or so. This is the level or volume control and it is adjusted by clicking it or dragging the red vertical bar with the mouse. Just to the left of this vertical red bar is either a white or a very bright red 'dot' (actually a short, horizontal line) that indicates the received signal level. If the receive gain has been set too high, this dot will be red and may be 'pinned' at the top of the scale; if this is the case, click on the red volume control bar to reduce the signal level. The fact that this dot is red indicates that the audio channel has saturated with the selected audio gain level. Reducing the gain will cause this dot to become white in color and to fall below the top of the graph. There are three red vertical lines superimposed on the signal in the baseband spectrum window. These are the BFO frequency control bars. You can change the frequency of the BFO and change the received sideband without taking the received signal out of the filter pass band or moving it in the display by dragging one of these three bars. The upper bar represents the true BFO frequency. With a narrow baseband displayed frequency scale, the BFO frequency may be outside the visible window and then this bar cannot be used to set the desired pitch. The lower bars have the frequency scale of the BFO frequency offset contracted by 10 and 100 times respectively, so that at least one control should always remain in the window and be available to set the BFO frequency. The baseband display has similar controls for setting the number of averages per waterfall display line, waterfall bandwidth, waterfall gain, and zero point of the waterfall as does the main waterfall display; refer to the discussion in the main waterfall display section above for further description. However, there is one important difference between the main waterfall display and the baseband display in the operation of these controls. In the baseband display, the small arrows on the top-left of this waterfall expand or contract the displayed frequency bandwidth just as they do for the main waterfall display, but the small arrows on the top-right, while appearing to work in the same manner, actually double or halve [left or right arrow, respectively] the FFT3 size. The baseband spectrum display also has similar controls as does the main spectrum display for controlling its center point, gain, and number of averages per spectrum point. Again, see the discussion in the main spectrum display section above for details.

There are other controls at the bottom of the baseband display for turning on either an amplitude limiter or expander, for choosing the coherent processing mode and adjusting coherent mode receive parameters. The leftmost of these controls sets the audio compression/expansion mode: "Off" for normal mode, "Exp" for amplitude expander, "Lim" for Simple amplitude clipping. The next box to the right gives a choice of 4 coherent processing modes: normal [off]; binaural CW (one ear delayed) [coh1]; coherent with signal (I signal) in one ear and noise (Q signal) in the other ear [coh2]; and signal (I) to both ears, and noise discarded [coh3]. If the signal is not quite stable enough for coh3, then using coh2 instead may be of some help. For EME, I find that running with [Exp] and [coh3] works very well. The next box [Rat 3] sets the ratio between the baseband filter width and the width of a subfilter used to extract carrier information for the coherent processing. The fourth box from the left [off] or [del] toggles on or off the signal delay between the ears. The delay can be activated only when 'coh' is not selected. The fifth box [8] is the value of the delay if selected in the previous box. The last [most rightward] box allows one to set the value of the delay controlled by the 4th box. For terrestrial work, I don't mess with any of these box controls on the baseband display.

Just to the right of the main waterfall and spectrum displays is the S meter display. It can be set to indicate either S units or dBm. It has several adjustable parameters that will not be discussed here. Place the mouse over each control and hit "F1" to learn what that control does.

Just to the right of the S meter are 4 small boxes. The top box is the <u>coherence graph and signal amplitude box</u>. This is primarily useful for EME work. The coherence graph shows the quality of the received signal's phase coherence. The digits represent from top to bottom: peak signal amplitude, current signal amplitude, and RMS amplitude value for everything that passes the filter. The colored dots in the upper half of this box show the statistics of the complex amplitude of the baseband signal, using the same color scale as the waterfall display. The distance from the center of the crosshairs is proportional to the signal amplitude. Zero phase angle is to the left. I is along the X axis, and Q is along the Y axis. If the operator has selected a large coherence ratio and there is no signal (white noise), then the points will scatter evenly in all quadrants since there is no correlation between the phase of the carrier and the instantaneous phase of the total signal in the filter passband. As a result a round area at the crosshair center will be formed. If a perfect signal with no QSB is present, the round area will move a distance along the x-axis that corresponds to the signal amplitude. A perfect CW signal in white noise will produce two circular areas, one at the center corresponding to key up, and one displaced to the right at a distance corresponding to the signal amplitude. If a signal has some chirp and constant amplitude, then the signal will form an arc with constant radius, with the phase drifting symmetrically around zero during the key down period. The horizontal bar below the crosshairs box shows the time duration of CW dashes in relation to the duration that would be optimum for the selected baseband filter. The dots should be near the center of the display if the filter width is set optimally.

Beneath the coherence graph and signal amplitude box in Figure 1, and ordinarily not visible unless dual polarity receive is being used, is the <u>adaptive polarization control</u>. By rotating with the mouse the line in the top portion of this control, you can select any desired receive polarization angle. Or, you can leave this set to automatic or 'adaptive' mode [as in this figure] and then the software constantly optimizes the polarization angle and phase. Moving the green line on the horizontal bar at the bottom of this control changes the polarization from linear to elliptical to circular. I usually leave the polarization control set to 'adapt' and let the computer do the work. The little "receive polarization" digit display that is a part of this control shows the received polarization angle, here 116 degrees as just noted.

Below the adaptive polarization control in Figure 1 is the <u>EME Window</u>. Once the EME Window is set up by typing "M" from the main menu window and the database files dir.skd, eme.dta, and allcalls.dta are placed in the C:\linrad_data directory, Linrad will provide data on DX EME stations. KB8RQ has been typed into one of the boxes, and the EME window gives his data in green. The display shows that the terrestrial azimuth from W3SZ to KB8RQ is 269 degrees and the terrestrial distance to KB8RQ's grid of EM79sv is 723 km. KB8RQ's azimuth to the moon is 218.3 degrees, and his elevation 75.2 degrees. W3SZ's azimuth to the moon is 237.1 degrees, his the elevation is 70.0 degrees. Given that the receive polarization angle of KB8RQ was 139 degrees, the calculated optimal transmit polarization angle for W3SZ is 111 degrees.

The fourth box is an add-on I made to Linrad to allow Linrad to control the frequency of my FT1000MP, so that I have in essence a "Linrad transceiver". This box allows me to account for the LO frequency of my transverter so that, for example, if Linrad is receiving on 144.045 MHz and my transmit converter has an LO if 116 MHz with an error of 500 Hz, I set the LO frequency in this box to 116.0 [MHz] and the SH [shift] to 500 [Hz]. Having done this, when I type "Q" Linrad will sent a CAT control to the FT1000MP to put it on the frequency of the received signal. I have written code to do this both from instances of Linrad running under Linux, and also under Windows. The Windows version allows instances of Linrad running on other computers on the network to control the FT1000MP remotely. Leif SM5BSZ plans to add transmit capability to Linrad in the future.

Below the baseband display and in the bottom right hand corner of the screen is the automatic frequency control box. This displays time along the horizontal axis and the received frequency vertically. The yellow trace is the signal-to-noise ratio of the received signal. The two traces representing the frequency of the received signal, in green, and the frequency of the DSP LO in white, are not visible, as the signal frequency of 144.045.580.2 is below the bottom of the display window, due to the fact that the signal has drifted considerably [~160 Hz] since the AFC was activated. The boxes at the bottom of this display allow you to set the averaging parameters for the AFC circuit.

The hardware control box is present if you have either the WSE hardware or an SDR-14 or SDR-IQ in use by Linrad. It allows you to set the center frequency of the hardware, and to control the RF attenuators of these devices. Because figure 1 was made using recorded data, this box did not appear on the screen.

Joe Taylor, K1JT, has placed an excellent Adobe Reader [pdf] file summary of Linrad's on-screen controls at: http://physics.princeton.edu/pulsar/K1JT/Linrad_On-Screen_Controls.pdf

Setting the Linrad parameters for each mode [SSB, weak cw, FM, etc] to values other than their default values is beyond the scope of this article. Leif has some additional setup information at the following URLs:

http://www.sm5bsz.com/linuxdsp/usage/examples.htm

http://www.sm5bsz.com/linuxdsp/install/uiparm.htm
http://www.sm5bsz.com/linuxdsp/linrad.htm

Finally, there is a Linrad users' group. You can sign up from the Linrad URL given immediately above.

B. Winrad

Winrad is the excellent SDR program for Windows by Alberto DiBene, I2PHD. It runs under Windows and will accept spectrum widths of up to 192 kHz. It works both with soundcards and with the SDR-14 and the SDR-IQ. There is an excellent Manual on Alberto's website at http://www.weaksignals.com , and you can download the software from there as well. Winrad requires a display resolution equal or greater than 1024x768, with 1280x1024 as ideal. If your display is set for a lower resolution, Winrad will not run. Winrad has a denoiser, a noise blanker, and graphical filters as well as a CW Peak function and a birdie zapper. It also has AFC capability. Winrad's user interface is intuitive and very easy to learn, even without the manual. There is a Winrad users' group on Yahoo.

C. PowerSDR

PowerSDR is the excellent open source Windows software package primarily intended for use with the

Flex-Radio transceivers SDR-1000 and Flex-5000. PowerSDR can also be used for receive with the SoftRock hardware, or any I/Q hardware. PowerSDR is extremely powerful, and yet very easy to use. It has a huge user base and an extremely active user group constantly improving the software. As I type this the current release version is 1.8.0, the beta version is 1.9.1, and the SVN version is 1410. PowerSDR will display up to 192 KHz of spectrum. It has graphically adjustable filters and when used with the Flex-5000 offers full-duplex performance due to its multi-threaded architecture, which uses separate threads for transmit and receive. The newest version has implemented an impulse noise subtraction algorithm in addition to more traditional DSP noise reduction and multi-tone automatic notch filters. When used with the Flex-Radios, it covers 160 through 6 meters, and in addition has built-in support for 14 additional bands to support transverters.

I use PowerSDR with my SDR-1000 and Ozy/Janus PIO and soundcard replacement hardware from TAPR as my liaison radio for microwave work. You can find out more about PowerSDR at http://www.flex-radio.com/Products.aspx?topic=powersdr1x . There is also a Flex users' group. To sign up for that group go to http://mail.flex-radio.biz/mailman/listinfo/flexradio_flex-radio.biz .

D. Rocky

Rocky is a very nice, simple to operate piece of SDR Software from Alex VE3NEA. It is mostly used by owners of the SoftRock series of radios and is a good way to get started exploring SDR radios. The URL for it is http://www.dxatlas.com/rocky/ Rocky is supported by the Yahoo SoftRock40 users group.

E. DttSP

DttSP is an open source software project by Frank Brickle AB2KT and Bob McGwier N4HY. It is the software core that implements the basic modulation, demodulation, signal conditioning, and synchronization processes required to operate a high-performance transceiver using DSP for signal detection, processing, and synthesis. It can be used from the Linux operating system independently of PowerSDR. There is a Yahoo users' group for it called "dttsp-linux". There is a webpage with some information on how to get started with it at http://www.nitehawk.com/w3sz/dttspw3sz.htm .

F. SpectraVue

SpectraVue is proprietary software provided by RF Space for use with the SDR-14 and the SDR-IQ. It is supplied with the SDR hardware by RF Space. It provides demodulation but does not have noise blanking or noise reduction capability. It can be downloaded from http://www.moetronix.com/spectravue.htm .

IV. SDR Hardware (See Also Table III)

A. WSE [Weak Signal Equipment] by SM5BSZ

The Weak Signal Equipment series of equipment by Leif Asbrink, SM5BSZ is high performance dedicated hardware for the experimenter, made especially to be used with Linrad. Obtaining maximal performance from an SDR was the key design factor, with no attempt being made to minimize size or power consumption. The hardware is designed to be used with the M-Audio Delta44 soundcard, with a 96 kHz sampling rate. 144 MHz EME was the primary motivation for this project, although in addition to the 144 MHz front end, Leif has made available a front end covering 500 MHz segments of the five Amateur Bands from 1.8 through 14 MHz (excluding the 60 meter band). The units can be completely controlled from within Linrad or by simple external control programs for either Linux or Windows. The phase noise limited dynamic range of the WSE hardware is reported to be -145 dBc/Hz or -148 dBc/Hz depending on whether one or two channels contain the offending signal. The IP3 on the HF bands with 10 dB preamplification is reported as +27 dBm. Under those conditions, the noise figure for the entire system is 11 dB. For 144 MHz use, the IP3 is also +27 dBm, and the noise figure for the entire system is 12 dB. All of these values were obtained with the Delta44 running at minimum gain.

Detailed discussion, schematics, performance figures, tuning and testing procedures are available at:

http://nitehawk.com/sm5bsz/linuxdsp/optrx.htm

B. Flex-5000

The Flex-5000 by Flex-Radio is just being shipped as I type this and there are as of yet no reports of the performance of production units by independent users. Based on reports by Flex-Radio however, the performance of this hardware when coupled with the PowerSDR software is expected to be superb. The Flex-5000 will have full duplex operation. It has dedicated separate transmit and receive transverter ports. It has 14 ‘extra’ bands to be assigned to transverters, and will have a second receiver available as an option later this year. Its bandscope will “see” 192 kHz of spectrum at a time, and has a +33 dBm IP3 as well as 105 dB two tone third order dynamic range, both with 2 kHz spacing. It does not require a soundcard, and connects to the computer via 1394 FireWire. It has optimized 11th order filters for all ham bands 160 meters through 6 meters, inclusive. I suspect that this radio will become the backbone of my computer-controlled VHF/UHF/microwave station. You can learn all about the Flex-5000 at http://www.flex-radio.com/

C. SDR-1000

The SDR-1000 by Flex-Radio has been discontinued, but many of these units are in service and with the arrival of the Flex-5000 they should gradually start to become available at attractive prices. It serves as an excellent IF radio for microwave work, and I use mine as IF rig for my 144-432 MHz liaison transverters. It has a reported IP3 of +26 dBm, and a 99 dB two tone third order dynamic range. Like the Flex-5000, it covers 160 through 6 meters. Because it uses the same PowerSDR software as does the Flex-5000, it shares many of the very nice features of that radio. With this radio one can set up a transverter matrix in software and then interface the SDR-1000 with a Universal Controller Board [UCB] to handle transverter switching and control. This is a very powerful feature set to have available. Mike King, KM0T has an excellent webpage at

http://www.km0t.com/pages/sdr.htm

where he outlines his progress and techniques for applying the SDR-1000 to microwave contesting. Using this setup he placed 1st in the ARRL August UHF contest for 2004, 2005, and 2006. There are more SDR-1000-related webpages available from his main URL at http://www.km0t.com .

Although the SDR-1000 is no longer being produced by Flex-Radio, there is a derivative of this radio advertised on the internet. This is the KDG-SR100. The manufacturer is DJ3UW of Kneisner + Doering Elektronik GmbH . I have heard no reports from users. There is a review of this radio in German at http://rw3ps.qrz.ru/Download/Messungen_KDG-SR100.pdf which indicates that the performance is not as good as that of the original SDR-1000, with IP3 being 5-10 dBm lower, MDS being 2-4 dBm higher, and IMDDR being approximately 10 dB lower than the SDR-1000.

D. SDR-14

The **SDR-14** is a 14 bit software defined radio receiver. It has two inputs. The first has a high-frequency amplified front end with 0-30 MHz receive capability. The second is a 'Direct' input port to the analog to digital converter, and using this port one can undersample and receive signals up to 230 MHz directly, with a maximum spectral display width of 30 MHz. For realtime display, however, the USB port limits the displayed bandwidth to approximately 150-190 kHz. Leif Asbrink, SM5BSZ, has reviewed the difficulties of using our standard methods to assess dynamic range in digital receivers such as the SDR-14. See Nov/Dec 2006 QEX (IMD in Digital Receivers, pp 18-22).

The SDR-14 by **RF Space** has functioned as the front end for my main Linrad 'bandscope' for several years. It is controlled by Linrad and has provided trouble-free operation via its USB connection to my Linrad computer. Before using it with or without Linrad one needs of course to install its drivers. The Windows drivers are supplied by RF Space with the unit. You can also download a zip file **ftd2xx.zip** containing the driver from Leif's webpage **http://www.nitehawk.com/sm5bsz/linuxdsp/linrad.htm** . Go to the section titled "Installing software under Microsoft Windows" to find the file. The Linux driver is available for download from:

http://sourceforge.net/projects/ftdi-usb-ft245 .

Leif SM5 BSZ has some pages discussing using the SDR-14 with Linrad:

http://www.sm5bsz.com/digdynam/practical.htm

http://www.nitehawk.com/sm5bsz/linuxdsp/usage/sdr14-w/sdr14-vw.htm

http://www.nitehawk.com/sm5bsz/linuxdsp/usage/millihz.htm

There is also a Yahoo users' group for the SDR-14.

E. SDR-IQ

The SDR-IQ by RF Space is a software defined radio receiver covering 500 kHz through 30 MHz. It is the little brother of the SDR-14. Unlike the SDR-14, it is powered via the USB port and its maximum sampling bandwidth, independent of USB constraints, is limited to 190 kHz. Its price is less than half that of the SDR-14 and for use as a microwave bandscope and receiver it performs just as well at W3SZ. It uses the same drivers as the SDR-14. It is possible to run two simultaneous instances of Linrad for Windows on one computer, one using the SDR-IQ as input and the other using the SDR-14 as input, as long as the SDR-14 is not powered up until after the instance of Linrad using the SDR-IQ has started. The Linux driver for the SDR-IQ is the same as that for the SDR-14 and can be downloaded from the URL listed above. There is a Yahoo users' group for the SDR-IQ.

F. SoftRock

Tony Parks KB8YIG designed the SoftRock as a $24 project for the AMQRP Club and after they sold out the 400 units for which they had contracted, he took the SoftRock on as a personal project and has shipped thousands around the world. A very limited number of 28 MHz SoftRock V7.0 kits were shipped for use as transverter IF radios. I use two of these as bandscopes, for 222 and 432 MHz. They have limited sensitivity, and so I use them with either a DX Engineering PRA-1 Preamp or a Kanga USA KK7B R2 Pro 28 MHz LNA. In addition, their image rejection is not as good as the other hardware discussed in this paper, their dynamic range is also inferior to the other products discussed, and they leak a good deal of LO frequency RF from the input port. But at a cost of approximately $20-36 each, their "Value" is superb! Dave Robinson, WW2R, has used the SoftRock V7.0 as part of his receive chain for 1296 MHz EME.

Bill Tracey, KD5TFD has put some performance data on the SoftRock V7.0 on the web at:

http://ewjt.com/kd5tfd/sdr1k-notebook/sr40/sr7/index.html . The MDS he measured was -112 dBm, consistent with my impression that the receiver is insensitive.

The Yahoo users' group for the SoftRock is called "softrock40".

G. The Time Machine

The Time Machine from Expanded Spectrum Systems is an I/Q receiver that covers the HF Amateur Bands and can thus be used as a 28 MHz IF receiver. Its output is fed into a soundcard such as the M-Audio Delta44. Its performance in terms of image rejection and strong signal response is subjectively somewhat better than the SoftRock series, but inferior to the other products discussed. ESS also sells a daughter board that makes it easier to use an external LO with the Time Machine.

H. HPSDR/TAPR Ozymandias/Janus/Atlas

HPSDR is the acronym for High Performance Software Defined Radio. It is an informal group of hams interested in SDR, many of whom are SDR-1000 owners. Its goal is to produce a "next generation software defined radio for use by radio amateurs and short wave listeners". As of the time of this writing, it has 14 project boards in various stages of development. The first four of these are in production and are being sold by TAPR: Atlas, Pinocchio, Janus, and Ozymandias.

The Atlas/Ozymandias/Janus combination was made to function with the SDR-1000 as a PIO Adapter providing communications between the computer and the SDR-1000 and a soundcard replacement, providing cost-effective A/D and D/A conversion. Atlas is merely the backplane on which Ozymandias and Janus sit. It provides power and data connections. Ozymandias is an interface controller board, providing control functions to Janus and the SDR-1000 and USB connection to the computer. Janus is the A/D and D/A module that eliminates the need to use a soundcard with the SDR-1000. It samples at up to192 kHz, twice what the Delta44 can do, and provides a very high level of performance.

Pinocchio is merely an extender board to facilitate testing of boards that need to be plugged into Atlas to function.

I use this combination with my SDR-1000 to give me a bandscope that is twice the width that I would have if I used an M-Audio Delta44, and to eliminate the need to hook the SDR-1000 up to a parallel port on the computer. PowerSDR has been updated to work with this set of boards.

V. Experimenter's Corner

Gnu Radio is an open-source software defined radio project spearheaded by Eric Blossom. It is written in C++ with graphics done in Python and is up and running on Linux, Windows, Mac OS X, and NetBSD systems. Eric has written an excellent introduction to Gnu Radio available here. Gnu Radio does not have at this time the same visibility and awareness among the Amateur Radio fraternity at large as do most of the other offerings discussed here, but there is an active Gnu Radio internet mailing list that generally has 10-20 new postings on a given day. Importantly, Matt Ettus has made available hardware that works with Gnu Radio in the form of a Universal Software Radio Peripheral [USRP] that connects to a computer via a USB port, and which accepts a variety of RF daughterboards which he has also made available. The USRP currently costs $700 and the individual daughterboards range from $75 to $275. The daughterboards available include $75 [each] DC-30 MHz transmitter and receiver boards, a $100 50-870 MHz receiver board, and a family of 5 low power [20-200 mW transceivers covering 400-500 MHz, 800-1000 MHz, 1150-1450 MHz, 1.5-2.1 GHz, and 2.3-2.9 GHz. See the URL referenced above for a complete list of hardware that Matt has made available. This project is definitely worth a look if you are interested about experimenting with SDR software and hardware in a serious way.

VI. Conclusion

Software Defined Radios provide important benefits to the VHF/UHF/Microwave operator:

1. A bandscope showing all stations appearing within a wide swath of spectrum

2. Noise Blanking and Noise Reduction capability allowing one to hear otherwise inaudible stations

3. When combined with computer control of station functions, the opportunity to have a powerful central console for display and control of all important station data and function

To get a start experimenting with SDR radios and DSP processing most cheaply and easily, get a SoftRock for the hardware and for software start with PowerSDR, Rocky or Linrad for Windows, using the default parameters for Linrad. Each of these software programs can be downloaded from the internet at no cost. Unfortunately, the 28 MHz SoftRock is no longer available so if you take this approach you'll be experimenting on the HF bands.

To get up a top-notch SDR up and running most quickly, effectively, and easily, get a Flex-5000 and PowerSDR.

For a less costly approach, try to find a used SDR-1000 to use with PowerSDR.

For an even less expensive, receive-only approach, get an SDR-IQ and use it as a transverter IF receiver with Linrad for Windows or Winrad.

This document was written with embedded html links. If you are reading a text version, go to http://www.nitehawk.com/w3sz/w3szdsp04.doc to get the version with the html links.

Tables I, II, and III should follow immediately after this text. They are printed in Landscape mode.

Figure 1 should follow after the tables. It is printed in Landscape mode.