A Low-Cost HF Channel Simulator for Testing and Evaluating HF Digital Sytems

Johan B. Forrer, KC7WW

Objective

The incentive and justification for this project was inspired by the author's desire to develop HF digital communications devices that effectively deal with the variable nature of the ionospheric propagation medium. Simulating the behavior of the ionosphere in real time allows for bench testing of HF modems and other communications devices. In the past, these so-called "HF Channel Simulators" used exotic and expensive computing hardware that was not available to the average amateur experimentor.

The simulator presented in this article is based on a low-cost floating-point DSP evaluation kit that accommodates a wide range of simulated conditions, including CCIR 520-1[1]. The simulation model is an implementation of the Watterson, Gaussian-scatter, HF ionospheric channel[2] model which is the *de facto* standard for this kind of work.

The article concludes with a summary of test results for a number of contemporary, forward error-correcting (FEC) HF digital systems tested on this HF channel simulator: PSK31, CBPSK, and MT63.

This simulator is a worthy addition to anyone's array of testing tools for developing DSP modem algorithms, routing or protocol development for HF communication systems.

The Challenge Posed by the Variable Nature of the HF Channel

HF propagation involves several interrelated phenomena that result in a highly variable propagation medium. This variability is a challenge to anyone that needs to design and implement effective high-speed digital communications systems for HF.

The ability to quantitatively evaluate how successful engineering designs carries though to real-world implementations, often makes the difference between success and failure. Experienced, well-equipped engineers use special tools such as channel simulators to shorten development cycles. These are invaluable for example, to verify dynamic range performance, acceptable signal to noise ratio performance, as well as a number of other factors such as adjacent channel interference and frequency/timing tolerances. These are very common real-world problems. Besides the evaluation of these basic factors, protocol performance is of equal importance. This has to do with how efficient frame and character synchronization is, how effective error control works, and how successful protocol adaptation actually is.

Although some of these tests may be done by on the air tests, however, F-layer propagation conditions are almost impossible to repeat thus there is not really a chance for making comparative tests this way. What

[1] CCIR Recommendation 520-1. Use of High Frequency Ionospheric Channel Simulators.
[2] Watterson, C.C., J.R. Juroshek, and W.D. Bensema. 1970. Experimental confirmation of an HF channel model. IEEE Trans. Commun. Technol., vol. COM-18.pp. 792-803, Dec. 1970.

really is needed is a means to create an artificial ionospheric test medium (“ionosphere in a box”) that can be reproduced at will. Only then is it possible to set up norms and milestones for performance evaluation.

Computer simulation is one way to obtain quantitative results. A simulation study based on theoretical concepts can provide the basis for establishing expected performance characteristics, also serve as a guide as to requirements for hardware and software expectations. It can provide an essential justification for continuing development work without the risk.

During test and development phases, real-time testing using a HF channel simulator is essential. The key to developing an effective waveform and protocol suitable for high-speed HF digital communications, is in understanding the behavior of the ionosphere and how it will impact communications.

Ionospheric Reflection Model

HF communication is typically characterized by multipath propagation and fading. Transmitted signals travels over several propagation modes to the receiver via single or multiple reflections from the E and F ionospheric layers. Because of different propagation times over different paths, signals arriving at the receiver may be spread in time by as much as a few milliseconds.

Ionospheric turbulence causes distortion in both signal amplitude and phase, in addition, different ionospheric layers move up or down, which leads to independent Doppler shift on each propagation mode. Ionospheric skywave HF, multipath arises from paths with different number of multiple reflections between earth and the ionosphere (multiple-hop paths) and from paths at multiple elevation angles connecting the same end points (“high” or “low” rays). Natural inhomogenuities of the ionospheric layers and polarization dependent paths because of magnetic-ionic effects also contribute to multipath.

The effect of these natural inhomogenuities in the ionosphere causes multipath spreads of 20 to 40 μs on each path or mode, and the high/low and ordinary/extraordinary rays results in a path spread of about 200 μs. For single hop links (800-2000 km), a maximum multipath spread of 100 μs is common. In this case, all paths are via the same reflection area and thus there is no significant difference in the Doppler spread on different modes. The channel is often a very slow fading channel, with time stabilities of 100 s or more, corresponding to a Doppler spread of 0.01 Hz. Multipath spread in the range of I to 2 ms for HF occur for short ranges (because of near vertical incidence) of under 800 km due to delayed energy arrival via repeated earth-ionosphere reflections or over long paths (2000 to 10000 km) that require two or more hops. On these long skywaves, different spread, controlled by the Doppler shift differences can easily range up to 1 to 2 fades per second.

Short-term distortion on the HF channel can therefore be described in terms of the parameters that specify the time-spread and frequency-spread characteristics, i.e., differential propagation delay between modes, and the strengths, Doppler spread on each mode.

Figure 1 shows an actual example of these different mechanisms in action *(This illustration provided by courtesy of J.P. Martinez[3].)* Martinez experimentally recorded an event on November 9, 1994 that by saving a digitized audio tone of a remote broadcast station’s carrier on a computer file. The broadcast station’s carrier was located on 7.7 MHz and arrived via the ionosphere; the broadcast station being located on the island of Gibraltar and the receiver located on the South coast of England. Subsequent processing of the recorded digital data revealed frequency-domain behavior over time. For this, the results of 256-point FFTs are presented as pixel intensity values on the Y-axis, with time plotted on the X-axis.

---

[3] Martinez, J.P., G3PLX, High Blakebank Farm, Underbarrow, Kendal, Cumbria LA8 8BN, United Kingdom. The author gratefully acknowledges J.P. Martinez’s permission to reproduce these experimental results.

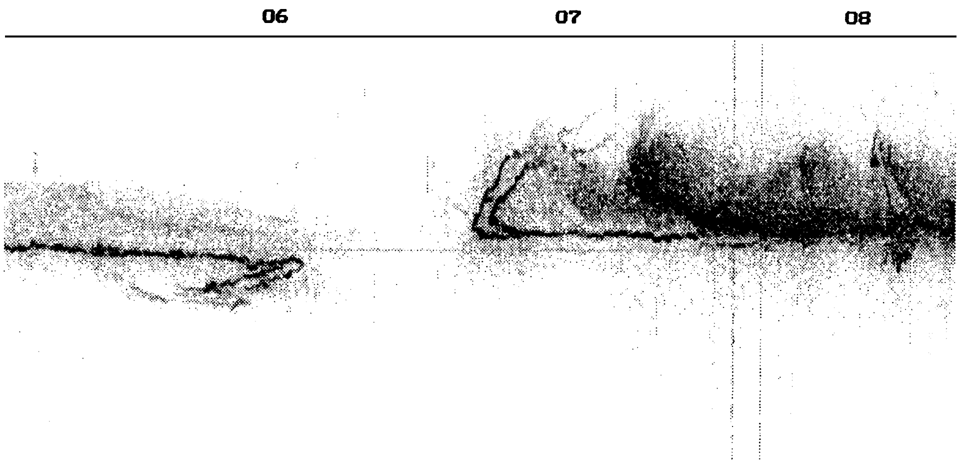

Figure 1. Martinez's Dopplergram illustrating several interesting ionospheric phenomena.

For the graph shown, each pixel point in time represents approximately 20 seconds of signal with UTC hour tic marks shown along the top. The Y-axis represents 0.025 Hz/pixel (256 pixels=6.25Hz). This representation effectively shows the history of a very slowly-changing process, with most of the finer, random events, filtered out to better illustrate the various propagation modes.

Because of the frequency in question (7.7 MHz), we are reasonably sure that the propagation mode is most likely via the F-layer. Note that at about 06:00 UTC the signal penetrates and no signal propagation path to Earth results. Just before this happens, note the high F-layer ray (the so-called, Pedersen ray) appear lower in frequency than the main (low) ray. The high ray itself appears to be split in two parts each with distinct Doppler shifts; the upper image being probably being the opto-ionic, or O-ray, and the lower image being produced by the extra-ordinary, or X-ray. The X-ray undergoes further retardation due to interaction with Earth's magnetic field. Shown is that the high and low rays of the O-trace penetrate first, followed by the X trace. This effect is distinct on this Dopplergram, but only rarely is it identifiable by ear.

If recognized, it appears as regular fading (QSB) that slows down to zero as the particular path fades out. About 06:40 UTC the F-layer comes back in again and the process is seen in reverse, X-trace appearing first and splitting into high and low, followed by the O-ray. Further more diffuse propagation paths open up a few minutes later.

The Watterson Gaussian-Scatter HF Ionospheric Channel Model

Watterson et al, using wide-band HF emissions over a path between Bolder, CO. and Washington, DC., proposed a model for narrow band HF channel. This model forms the basis for most modern HF channel simulation work and often are used for both software and hardware channel simulation.

This model, known as the "Watterson Gaussian-scatter HF ionospheric channel model", assumes that the HF channel is non-stationary in both frequency and time, but considered over small bandwidths (<10 kHz) and sufficiently short times (<10 minutes), most channels can be considered representative by a stationary model.

The HF channel is modeled as a tapped delay line, with one tap for each resolvable mode (or path) in time. The delayed signal is modulated in amplitude, and phase, by a complex random tap-gain time-dependent function that is defined by:

$$G_i(t) = G_{ia}(t)\exp\left(j.2\pi.f_{ia}.t\right) + G_{ib}(t)\exp\left(j2\pi.f_{ib}.t\right)$$

Where a and b subscripts denote the i-th element in a time series representation for two magnetoionic path components. In this context, G_{ia} (t) and $G_{ib}(t)$ represents two independent complex bivariate Gaussian ergodic random processes, each with zero mean and independent real and imaginary components with equal RMS values that produce Rayleigh fading. The exponentials provide frequency shifts f_{ia} and f_{ib} for the magnetoionic components in the tap-gain spectrum. Each tap gain has a spectrum $H_i(\lambda)$ that, in general, consists of the sum of two magnetoionic components, each of which is a Gaussian function of frequency, as specified by:

$$H(\lambda) = \frac{1}{\left(A_{ia}.\sqrt{2.\pi.\sigma_{ia}}\right)}.\exp\left(\frac{-\left(\lambda-\lambda_{ia}\right)}{\left(2.\sigma_{ia}^2\right)}\right) + \frac{1}{\left(A_{ib}.\sqrt{2.\pi.\sigma_{ib}}\right)}.\exp\left(\frac{-\left(\lambda-\lambda_{ib}\right)}{\left(2.\sigma_{ia}^2\right)}\right) \quad ,I$$

where A_{ia}I and A_{ib} are component attenuations and the frequency spread on each component is determined by $2s_{ia}$ and $2s_{ib}$.The frequency shift on the two components are given by λ_{ia} and λ_{ib}.

Tap-gain distributions for a two-ray model are shown in Figure 2.

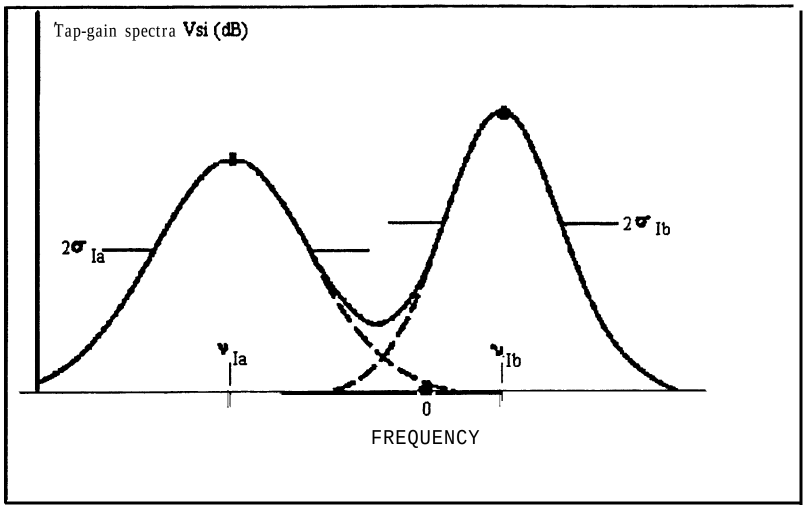

Figure 2. Tap gain distributions for a two-ray model.

Notes:

1. The Watterson model implies the use of equal power (RMS) paths. This effectively is like a deep notch filter sweeping through the passband – at times completely obliterating parts of the signal. This often has devastating implications for some modem algorithms and some end users of this simulator has expressed their concerns as "it not being realistic for typical HF conditions." In order to reduce the depth of the null, it is possible to weigh tap gain functions such that they are never equal, however, this practice should be for in-house developments only and not for publication as such results will include unjustified bias.
2. In attempts to compare performance results of standard equipment against published materials where professional channel simulators have been used (manufactured by Harris Corp. for example,) it has been found that there appears to be some leeway in interpretation of the Watterson model and subsequent discrepancies in results. There has been investigations by researchers on this subject , however, without having access to details on proprietary implementations, these discrepancies remain unresolved.
3. Generally, published specifications or research results often tends to omit weaknesses that are readily shown by such simulators. More often than not, results obtained by this simulator tend to be interpreted as highly critical or erroneous. This is not the intention, rather should be an opportunity that should be exploited to the user's advantage.

CCIR Recommendations for the Use of HF Ionosperic Channel Simulators.

CCIR Recommendation 520-l gives guidelines for practical values for frequency spread and delay times between ray components:

Condition	***Freq. Spread (Hz)***	***Delay (ms)***
Flat Fading	***0.2***	***0***
Flat Fading (extreme)	1.0	0
Good	0.1	0.5
Moderate	0.5	1.0
Poor	1.0	2.0

It is proposed that these parameters be used to validate average and extreme conditions during simulation as well as during actual hardware testing.

The Development of a Real-Time HF Channel Simulator

Discussions on developing a low-cost HF channel simulator took place on several forums; TAPR HFSIG list, specifically during 1994, 1995 TAPR Annual Meeting in St. Louis, MO., Digital Communications Conferences (DCC), 1995 Arlington TX, and 1996, Seattle, WA.

Early work involving Alexander Kurpiers, DL8AAU, Darmstadt Germany, produced code for a TI 320C26-based DSP implementation. The author ported this for use on the TAPR DSP93 and demonstrated its use at the 1996, DCC meeting in Seattle, WA. This model has seen service in several projects, however has limited performance due to memory and processor limitations.

Several others shown active interest in this project; Barry Buelow, WA0RJT, Jon Bloom, KE3Z, Eric Silbaugh, Glen Worstell, KG0T, Phil Karn, KA9Q, and especially Tom McDermott, N5EG. Tom presented a paper on theoretical aspects of HF channel simulation at the 1996 DCC HFSIG meeting.

The specifics for the implementation of the Watterson Gaussian-scatter HF ionospheric channel model follows. This topic is divided into two sections: the hardware platform and software implementation.

HF Channel Simulator Hardware

The author realized the opportunity when a new floating point DSP evaluation module (EVM) by Analog Devices[4] became available. The EZ-KIT Lite SHARC is a 40 MIPS processor that can produce 150 MFLOP performance in floating point. The SHARC DSP follows modern trends where its instruction set is optimized for use with the C programming language.

The kit was supplied with GNU-based C tools on CDROM that included the usual compiler, linker, and librarian tool chain. The ability to use a high-level language made the implementation of the Watterson-model mathematics much easier. Even time-critical code like interrupt handlers may be written in C, alternately, either in-line assembly or assembly-language modules may be developed. The EVM contains a 48kHz stereo CODEC to handle audio I/O, also a UART chip to handle serial communications with a host. The DSP contains a total of 16K 48-bit words of on-chip memory, part of which is available for user code. The amount of on-chip user memory is adequate for implementing the Watterson-model simulator.

HF Channel Simulator Software

A paper by Ehrman et al.[5] provided basic implementation ideas that was used in this project. Several parallel tasks can be distinguished:

1) Transform and process the baseband input signal such that its phase and amplitude properties can be manipulated in real time,
2) Simulate, independantly, in real time, a pre-defined HF propagation condition,
3) Apply simulated distortion to the processed input signal, and,
4) Apply noise pertubations.

---

[4] Super Harvard Architecture Computer (SHARC) EZ-KIT Lite. Part number: ADDS-2106X-EZLITE. Available from Analog Devices distributors. Street price $179.
http://products.analog.com/products/info.asp?product=21000-HARDWARE

[5] Ehrman, L., L.B. Yates, J.F. Eschile, and J.M. Kates (1982.)
Realtime Software Simulation of the HF Radio Channel. IEEE Trans. on Communications, August 1992, page. 1809.

Figure 3 shows the interaction between a number of parallel tasks. Input is applied at the top left and output produced at the bottom right of the figure.

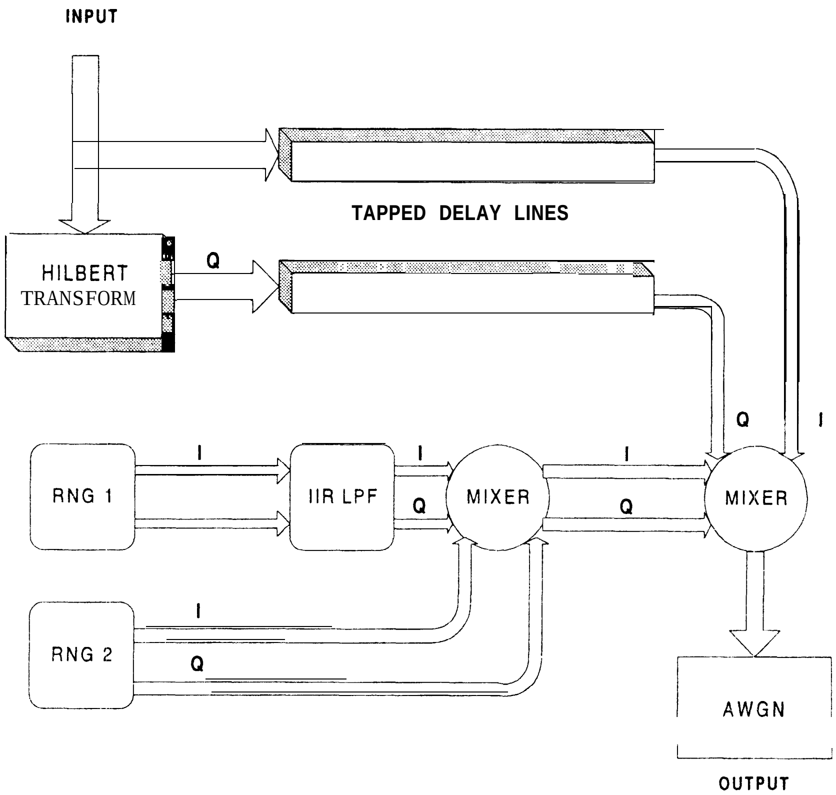

Figure 2. Simulator Process Flow.

The Watterson model only deals with the effects of the ionosphere and the distortion that it introduces -- it does not attempt to simulate HF noise pertubations. CCIR 520-l also does not specify any kind of noise source, however alludes to including a noise source in simulation.

These processing steps are now analyzed in further detail:

Input Signal Processing

The input signal is a real signal. Fading and Doppler effects will be introduced to this signal by a process of signal mixers. These mixers, however, are complex devices requiring in-phase (I) and quadrature (Q) components, thus requiring that the input signal be an analytic signal. This conversion of the input signal is achieved by using a Hilbert transform.

To simulate multiple rays passing through the ionosphere, dual tapped delay lines are used; one for the I component, another for the Q component. The analytic input signal is then extracted from the appropriate points in the delay lines -- the position in the delay line is a function of the input sample rate (typically 9600 SPS) and the required path delay (varies between approximately 0.1 mS to 10 mS, or 1 to 96 delay line taps).

Computing Channel Effects: Doppler Shift and Fading

Watterson et al. showed that the desired fading and Doppler shift can be introduced by the product of two Gaussian functions, i.e., a Rayleigh distribution. Since this multiplication process of the two Gaussian functions are commutative, it does not matter what gets generated first; the fading function or the Doppler shift.

Assuming the fading function, that gets produced from a random number generator with Gaussian distribution output. This stream of numbers are then passed through an infinite impulse response filter (IIR) designed for appropriate bandwidth, i.e., that determines the fading bandwidth. Actually it controls the statistical spread for this Gaussian function, like that shown in Figure 2.

Doppler shift is produced on the fading function using a similar method, except that no filter is used. After performing the *complex* mixing of the fading and Doppler functions, the resultant signal now has a Rayleigh distribution. That is the desired tap-gain function, or modulation function to be applied to the delayed analytic input signal. The final outcome is to take only the real part of this last mixing step.

As an option, noise perturbations with the correct amplitude are then added to set the noise background for the desired signal to noise (SNR) level.

The computation of the noise background requires further consideration.

Computing Channel Noise Effects and SNR

Gaussian noise models are commonly used in VHF, UHF, and microwave work, however, HF noise behavior is more complex and sometimes described in terms of Markov models, rather than stochastic models, in the literature. For purposes of this paper, only Gaussian noise is considered – this simplifies matters, however, does not accurately represent HF channel noise.

The exact channel measurements that typically are used for comparing systems should be carefully considered. Classical reference books use bandwidth-normalized SNR measurements. This reflects a unit of "bits per second per Watt per Herz" instead of a simple signal to noise ratio values. When dealing with real-world communications systems, however, this kind of measurement is difficult as power measurements need to accurately known at exact bit timings in order to compute the actual energy per bit. Coding schemes and ARQ protocol issues further complicate this measurement. It often is more convenient to determine throughput rate instead, but there would be difficulty to relate this to Eb/No as used in reference materials.

In this regard, Leeland's[6] discussion on methods to determine bit-error rates (BER) is of interest. It is suggested that BER should be this the basis for evaluating modem performance – if it doesn't meet BER specifications, it doesn't work as expected. That may imply that defensive actions like dynamic protocol adaptation and/or tracking algorithms are failing to assess channel properties correctly. BER also allows one to compose the classic "waterfall" BER vs. SNR curves. These sets of curves allows one to check measured performance against theoretical (expected) performance, but also to compare your work against other published work.

Allowing remote requests through the modem's host control port can retrieve performance measurements can assist algorithms doing a better job; Raw BER, corrected BER, and Eb/No comprise the standard suite of measurements. Raw BER is the actual count of erroneous data bits detected and corrected by the

[6] Leeland, Steven. Digital Signal Processing in Satellite Modem Design. Communication Systems Design, June 1998.

decoder. Corrected BER is the estimated BER after the decoder has reconstructed the original data stream. Eb/No is, of course, the signal-to-noise (SNR) ratio.

Historically, raw BER has been measured within the decoder circuitry by counting the number of detected/corrected bits over fixed time durations. The error count register formed the address to PROM based LUTs to supply the actual raw BER in $x.y*10^{-z}$ format to the control processor. When the BER gets as high as the 10^{-3} region, or some other arbitrary value, the decoder is usually ready to give up the ghost and declare loss of lock.

Through a set of arcane heuristic algorithms, the same lookup PROM generates the estimated corrected BER and Eb/No. Due to resolution, there is an upper limit to how well a BER can be measured with this technique. When these limits are exceeded, the results are reported as less than $1E^{-4}$ for raw BER, less than $1E^{-9}$ for corrected BER, and greater than 9.9 dB for Eb/No.

Modern modems use calculated Eb/No methods for BER estimation. The Eb/No is calculated from the measured SNR using symbol data. The SNR is computed from the mean, Mx, and the variance, Sx, of the data as follows:

$$S/N = (Mx)^2 / (Sx)^2$$

where

$$Mx = \sum (Xi) / N,$$

and

$$(Sx)^2 = \sum ((Xi - Mx)^2 / (N - 1))$$

For BPSK and QPSK, Xi is the absolute value of the I-Channel data. For 8-PSK, Xi is *Sqrt*(I*I+Q*Q). The sample size, N, should be as large as is feasible. In order to maintain a report rate of I sec at say 200 symbol rate, the sample size is constrained to 200.

For some modem implementation, there are three problems with this scenario. I and Q data are digitized on both the falling and rising edge of the symbol clock. Only one edge will be correct after the Costas loops are locked. The problem is that the digital Costas loop circuitry knows which edge is correct, but the DSP does not. Another problem can be gleaned from the form of the equations given. The variance equation requires knowing the mean of the entire sample set before calculating each term in the summation.

This requires storing the entire sample set in DSP memory. Internal DSP memory is insufficient for the task, and external memory is an undesirable expense in both cost and, more importantly, board real estate.

The third problem is the square root operation required for 8-PSK:
It is not trivial to find the kind of bit edges that produce high levels SNR. For example, how does the algorithm know which edge to use, falling or rising, for the I and Q data measurements? Of course, this kind of algorithms often comes at a price – it will consume additional DSP execution time resources.

A C-code snippet shown in Listing 1 shows one approach to computing SNR. It is shown that it is no longer necessary to first compute the mean of the entire sample set. Instead, the algorithm only computes the sum of the samples and the ***sum of the samples squared.***

```
Float snr(int sum, int sum2, int samples)
{
```

```
float mx, mx2, mi2, sx2;
        mx = ((float)sum)/samples;
        mx2=mx*mx;
        mi2=((float)sum2)/samples;
        sx2 = mi2 - mx2;
        return mx2/sx2;
}
```

Listing 1. Implementation of SNR calculation in C.

The 8-PSK samples require a square root operation from the specifications given in Listing 1. This is a very undesirable operation for the DSP to perform on each sample in the set of data. It consumes valuable DSP execution time resources. It requires finding and testing a square root routine.

To resolve this dilemma, I and Q data are first absolute valued. This essentially folds the eight points of the 8PSK constellation into two points in the first quadrant. To fold these two points into one, the I and Q data are compared. The larger value is used as the sample value. This is the same as comparing I and Q. If Q is greater than I, then swap I and Q. Finally, use the I value as the sample, the same way as in BPSK or QPSK.

The Eb/No is calculated from the SNR as follows:

$$Eb/No = 10*\mathrm{Log}((1/2)(S/N)(1/c)(1/p)) - M$$

where *c* is the code rate, *p* is the symbol packing rate, and *M* is the modem loss (nominally 0.5 dB).

The packing rate is 1 for BPSK, 3 for 8-PSK, and, normally, 2 for QPSK. However, because we are using only I data, *p* is also 1 for QPSK.

If Reed Solomon decoding is installed and enabled, then:

$$Eb/No = Eb/No + 10*\mathrm{Log}\,(N/K)$$

where N and K are the Reed Solomon encoding factors.

Finally,

$$Eb/No = \mathrm{Fudge}\,(Eb/No)$$

where "Fudge" is a function that accounts for differences between theoretical versus real-world modem situations.

Test Results

Simulator tests were performed on three FEC communications modes: PSK31, CBPSK, and MT63 as examples. In this example, the test condition used was CCIR POOR, which comprises the use of two equal-power rays with 2ms differential path delay, 1 Hz Doppler frequency spread. The SNR level was set at -10dB SNR. This represents a 3kHz bandwidth AWGN channel. This test condition represents marginal HF conditions, that probably are close or at the practical limit for reliable HF communications.

Results are shown in Appendix 1.

Acknowledgements

This work was made possible by generous contributions made by participants of the TAPR HFSIG list and discussions at various DCC meetings. Not only did these forums stimulate the development of this HF channel simulator, also new HF digital communications modes like PSK3 1 and MT63.

The author gratefully acknowledges the contribution of TAPR in this respect and wish to thank those that participated in the multitude of interesting and educational postings on the HFSIG list.

The contributions of Peter Marinez's, G3PLX, ionospheric "Dopplergrams" as well as work on PSK3 1 is gratefully acknowledged.

A special word of appreciation to Pawel Jalocha, SP9VRC who brought us SLOWBPSK, the granddaddy of PSK31 and MT63.

Free demonstration simulator code is available for downloading[7] from the author's world-wide web site.

[7] http://www.peak.org/~forrerj

Appendix 1

Simulator tests results performed using PSK31, CBPSK, and MT63 under CCIR POOR conditions (two equal-power rays with 2ms differential path delay, 1 Hz Doppler frequency spread) at -10dB SNR, 3kHz Bandwidth AWGN.

The contents of the test message is the "TUNER program" as shown. The results after passing the test message through the simulated channel using the selected HF communications mode are shown.

Notes:
1. Due to decoding errors, some unprintable control characters were encountered that caused the word processor to make substitutions, more often than not, line feed characters.
2. The last test for the 2kHz bandwidth MT63 used –5dB SNR.

The "test" message:

The TUNER program - TUNER.COM

1. This is a tuning aid to help get a received tone exactly on 800.0 Hz.
It should accept COM2, COM3, COM4 command line parameters (default is COM1)
and report CLIPPING (audio signal too strong for the sigma-delta circuit).

2. Unfortunately it takes too many computing cycles to incorporate this
in COHERENT, so run TUNER first if necessary, using an 800 Hz sinewave
with no modulation on it (a steady carrier in other words).
It may be slightly useful on a carrier that is phase-modulated, but
the indicator will jump around trying to follow the modulation, and in
any event the useful frequency range would be limited.

3. The idea is to get the little yellow line centered between the 2 green
lines, and staying within the green lines at all times. The nominal
frequency is 800.0 Hz.

4. The range of this tuning indicator is 800 Hz plus or minus 20 Hz.
If your signal is not ALREADY tuned to within better than 20 Hz, this
indicator will be useless and quite likely confusing as hell!

5. There will be some rejection of other signals outside this range, but
if the signal you want is weak and the interfering signals are strong there
will no doubt be problems.

6. If you can hear the tone, there is no substitute for zero-beating it
with a good crystal-derived 800 Hz sinewave sidetone.

7. TUNERC.COM is for anyone who still uses CGA graphics - I slowed down
the update rate to accommodate sluggish LCD displays.

VE2IQ - November '95.

Simulator Results; PSK31 with Varicode

The UNERO on ramD TðER.ROO
-r- tiDi---iDe -- ------- ul----

1. Tt=s is a tuning ai t tfheð ge a repotvedi/e I tc&a ð 800.0 Hz.
It trould a oc?t Cr M2o COOððr MM cemmand line farao etera
default is ttO01)
nnd report CLIPðð Maudl sign&oo stroog for ðe oigma-deiit circðe0ð
2. Unftrtunately 7 takes too many coeeputing cycleðo
oraorate this
tn CO ERENet, so run AU1 ERKirst f ne eessarð using8aeD Hz sinewað
with vtmosul ti/ on it(a stead t carri=n otrer wordo).
At o aybe slightly uð ul on a carier that s pha8-ðdulat d, but
the iodiahto ai oa juee]arount trying-f how the modllation, and in
any even th u eeul frequ ncy r a ae woðdbe liðte .

6 Tme ic a is to get t& littliyellow line cen Ved betweefta2tireen
ones, and staying wtthiihhe yreei lines at alðimesEii he nominal
frequenc io 80gbte 6.
14. The ron ne of this tuningyodicator is 800 rðplus oa ðnur 2ðHz. ebf yoeer signae es ntt yieREðY tu ld o
wVain betoer taan $g Heret this
indicncr hill be us$ess Ld puite likelm honfus_gas helle

5. Theri ailLb
eome re:eðioa of othetðgnals tutsi T this raegð but
im hlsia nal you wtnhis geak and the interferinte signalalrst tng therqwill no doubt be problems.

a. _f yol can hea e the eone, thite iDno substTðe foi >ero tbeatie?wð a good crð alder ved 800 r z si(vave
si tetooer

C. TUPE eC.COM ð dð aoaone i6o stilTZ es c(graphics u I ðoe e6down
hup tat . te to adcommo rte sluggish PCD disðays.

VE2IQ - Gog]'r '9$.

Simulator Results; C-BPSK, ET-2

Thn TUNER program -4TUNER.COM

---------------E-------------

1. Thiy is a tuning aid to help get a
Ered+i8nd tone exactly on 800.0 Hz.
It should accept COM2, COM3, COM4 c
ommand line paramKtebs (de@aul' is COM1)
and report CLIPPING (audfo sig
nal too strong for the sigmp-delta circuit).t

. Unfortunately it tak
es tooEmant computing cycles to incorporate thsm
n COHERENT, to rcn TUN
ER first if necessary, using an 800 Hz sinewave
citp no modulation on i
t (a steady carrie1 in other words).
It may be slightly useful on a car
rier that
phase-modulated, but
t_e indicator will jump_around tryidg
4to follow the modulation, Gnd in
any event the usedul frequency range
would be limited.
e
3t The idea is to ge0 the little yel.ow line cente
red between the 2%green
lines,__nd sta-ing within the green lines;at al
l times. The nominal
frequency is 800.0 Hz.

4. The range of this t
uning indicator is 80f Hz plus or minos 23 Hz.
If your signal is not AL
READY tmnOd to within better than 20 Hz, this
indicator will be uaeless
and quite likely confusing as hell!
}
5. There will be some rejection
OofEoeher signals outside this range, but
if thesi9nal you4want is weak
and the int_rfering signals are strong there
gwill no doube be probleds

6. If yod can hear the tone, there is no s#bstitute for zero-beati
ng it
with a geod+crystal-derived 800BHc:sinewave sidetone.

<. mUHE
RC.CE\q5;T7wS_zunwg1sW_2kT=aRh
_es CGA graphics - I slowed Town
the up

datb rate to aTco l modate sluggish LC4 displays.

1VE2IQ - Novemeer ‘95.

Simulator Results; MT63 - 2kHz, double interleave factor.

The TUNER program - TUNER.COM

----

1. This is a tuning aId to help geT a received tOne exactly on 800.0 HZ
It should accept COM2, COM3, COM4 command line parameters (default is CoM1
and report CLIPPInG (aUdio signal too stRong for the sigma-delta circuit).

AG5q 2. Unfortunately it takes too many computing cycles to incorPoratE this
/kin COHERENT, so run TUNER first if necessary, usiNg an 800 Hz sinewave
with no modulatIon on it (a steady carrIer in oTher words).
I _ It may be slightly useful oN a carrierthat is phase-modulated,but
the indicator will jump around trying to follow the modulation, and in
* bany event the useful frequency range would be limiteD.
= ?jq3. The idea is to get the little yellXw line centered between the2GReen
lines, and staying within the green lines at all times. Thenominal
dfrequency is 800.0 Hz-
* x __ 44. The range of this tuning indicator is 800 Hz plus or minus 20 Hz.
9x m If your signAl is not ALREADY tuned to within better than 20 Hz, thIs
x - m~lindicator will be useless aNd quite likely confusing as hell!
iQe 5. There will be some rejection oFOtHer signAls outside this range, but
if tHe siGnal you wAnT is weak and the interfering signals are strong there
LB ut5Jl1 1_11 no doubt be problems.

6. If you can Hear the tone, there is No substitute for zero-beatingit
u with a. goOd crystal-derived 8_0 Hz sInewave sidetone.-

6 d2 7. tUNERC.cOM is for anyOnEwho still uses CGA
* the update rate to accommodate slUggisH LCD disPlaYS.
iP __VE2IQ - November ‘95.

Simulator Results; MT63 - 2kHz, double interleave factor
(Test at -5dB SNR, 3kHz Bandwidth AWGN.)

The TUNER program - TUNER.COM

----

1. This is a tuning aid to help get a received tone exactly on 800.0 Hz.
It should accept COM2, COM3, COM4 command line parameters (default is COM1) and report CLIPPING (audio signal too strong for the sigma-delta circuit).

2. Unfortunately it takes too many computing cycles to incorporate this in COHERENT, so run TUNER first if necessary, using an 800 Hz sinewave with no modulation on it (a steady carrier in other words).
It may be slightly useful on a carrier that is phase-modulated, but the indicator will jump around trying to follow the modulation, and in any event the useful frequency range would be limited.

3. The idea is to get the little yellow line centered between the 2 green lines, and staying within the green lines at all times. The nominal frequency is 800.0 Hz.

4. The range of this tuning indicator is 800 Hz plus or minus 20 Hz.
If your signal is not ALREADY tuned to within better than 20 Hz, this indicator will be useless and quite likely confusing as hell!

5. There will be some rejection of other signals outside this range, but if the signal you want is weak and the interfering signals are strong there will no doubt be problems.

6. If you can hear the tone, there is no substitute for zero-beating it with a good crystal-derived 800 Hz sinewave sidetone.

7. TUNERC.COM is for anyone who still uses CGA graphics - I slowed down the update rate to accommodate sluggish LCD displays.

VE2IQ - November '95.