EME 2000: Applying Modern Communications Technologies to Weak Signal Amateur Operations.

Tom Clark, W3IWI (clark@tomcat.gsfc.nasa.gov)
Phil Karn, KA9Q (karn@qualcomm.com)

July, 1996

1. Introduction:

The availability of several new communications technologies permit significant advances in amateur weak-signal operation. In this paper we will outline some ideas which could achieve ~20 dB improvements over conventional practice. We will focus our discussions in the Moonbounce (EME) arena at VHF/UHF/Microwave frequencies, but the ideas are applicable to other activities.

As an idea of the kind of gain that can be achieved, we look back in history to 1964/5 when Gordon Pettingill and Rolf Dyce (*Nature* 206, 1240, 1965) used the Arecibo 1000' antenna at 430 MHz to determine that Venus had a rotation period 2/3 of its orbital period. Comparing their achievement with a modern amateur EME QSO at 70cm between a "big" and "small" station:

Item:	RADAR ASTRONOMY	AMATEUR EME	dB
Transmitter Power	250 kW	1.5 kW	22.2
Transmitting Antenna	250 Meter dish	9 Meter dish	27.9
Distance	67,500,000 kilometers	350,000 kilometers	-91.4
Receiving Antenna	250 Meter dish	20 dBd Yagis	34.8
Receiver Tsys	250 K	80 K	-4.9
	NET dB DIFFERENCE =		**-11.4**

Notes and assumptions:

- The radius of Venus's orbit around the sun is 0.72 AU, so the closest it gets to earth is 0.38 AU. The 1965 radar observations were made around closest approach, so a distance of 0.45 AU is taken for these calculations. For the moon, we have taken a distance near perigee.
- Although the Arecibo dish is 1000' (305m) in diameter, its spherical shape leads to inefficiencies so the effective diameter is taken as ~250m for this calculation and an aperture illumination efficiency ~ 40% has been assumed (57 dBi = 54.8dBd gain).
- The amateur transmitting station is "state-of-the-art": a 9m (30') dish with 50% aperture efficiency (29.2 dBi = 27 dBd gain) and a 1.5 kW transmitter.
- The amateur receiving station is running a "small" Yagi array with 20 dBd (22.2 dBi) gain.
- Modern FET/HEMT LNA technology gives the amateur better Tsys receiving performance than was available in 1965 (~250K in 1965 vs. ~80K easy to obtain now).
- Even in 1965, The Arecibo radar performance with the 1000' dish was so good that EME echoes could be obtained with a Model-80 signal generator, and "EMEME" (i.e. double-hop reflections, with a net R^8 path loss) were observed with the big transmitter! With the same radar system, Arecibo was able to obtain the first maps of the surface of cloud-covered Venus.

30 years ago, the professional community could do valuable scientific research with link margins more than 10 dB poorer than we have today. They did it by clever signal processing. Can amateurs with the technology available now obtain similar performance enhancements? We think that the answer is a resounding YES! For a target, let's look at a comparison between the amateur EME example used above and a pair of typical OSCAR satellite stations shows:

Item:	AMATEUR EME	OSCAR Station	dB
Transmitter Power	1.5 kW	150 W	10.0
Transmitting Antenna	9 Meter dish	17 dBd Yagis	10.0
Receiving Antenna	20 dBd Yagis	17 dBd Yagis	3.0
Receiver Tsys	80 K	100 K	1.0
		NET dB DIFFERENCE =	**24.0**

Our goal in this paper is to examine ways that amateurs can “buy” performance improvements of 10-15 dB better than Pettingill and Dyce used 30 years ago for the purpose of making valid QSOs. The “purchase” must use technology that is inexpensive, readily available and easy to replicate. The key elements that the amateur will need are:

1. A Pentium-class PC, which is augmented with a SoundBlaster card to provide A/D and D/A conversion at audio frequencies.
2. Millisecond accuracy timing clock (a small GPS receiver -- W3IWI’s “Totally Accurate Clock” will work fine. See Appendix A)
3. A low-speed digital logic plus audio frequency analog adapter “widget” to “glue” the computer, clock and radios together.
4. An OSCAR-class (or better) satellite station.

2. What’s a QSO?

The initial questions/comments from the skeptics when advanced signal processing schemes are discussed always seem to start with the age-old premise:

YOU’VE GOT TO HEAR ‘EM TO WORK ‘EM !!

And then follow with the dictum:

A QSO ISN’T VALID UNLESS IT’S IN REAL-TIME !!

The schemes we discuss here are not “earball” based, and the S/N ratios will be very low, so we have to dismiss the first point as irrelevant. As a *reducto ad absurdo* analogy, consider the case of two deaf amateurs who succeed in making an EME QSO with RTTY. Would even the most hardened “you’ve got to hear..” advocate deny the validity of the RTTY QSO?

The “real-time” issue stems from people’s pre-conceived notions that a QSO shouldn’t take hours of off-line data analysis. To answer this issue, we extend the deaf amateur analogy to the use of packet radio instead of RTTY. The sending amateur types in an entire line of text and hits the return key. The computer in the TNC assembles the text into a standardized packet and appends control information and a check-sum which is sent as a single burst. The receiving station disassembles the packet and if the check-sum is verified presents the entire line at one time a few seconds later. Is this real-time? Would the deaf amateurs be able to count a packet QSO?

The scheme we are proposing can be envisioned as a glorifed (albeit slow) form of packet radio. We envision that the information needed for a QSO takes about a minute to transmit and that the information will be presented a few seconds later at the other end. So we regard the “real-time” arguments as moot.

The requirements for a “legal” QSO read something like:

- Each station must copy the call of the other station and know that the transmission is directed to him.
- Some unknown information must be exchanged in both directions. This is usually a signal report but might also include a Grid Square.
- Each station must receive an acknowledgment that the other station has copied the required information.

3. Raw Bits!

Clearly, we are thinking in terms of true digital communications. On our computers we all use utilities like PKZIP to compress data into its most compact form. A program like PKZIP uses a well-defined mathematical algorithm that looks for redundancies in the entire block of data and represents repeated material as a much shorter "token". It keeps a dictionary of the tokens it finds and appends it to the tokenized data. A partial PKZIPped file is useless since only the complete file can be restored.

Since the information needed to complete a QSO is well defined, we have attempted to pre-define the "QSO dictionary" in order to reduce the information to the minimum possible number of raw data bits. To these "raw" bits, we will add additional bits for error correction in order to improve the reliability and to facilitate signal processing; we'll return to this topic later, but first let's examine the "raw" information needed to conduct an EME QSO.

In communications theory, this is called 'source coding'. Our goal is to get the most out of every user data bit by encoding all our information as compactly as possible. It's a lot like playing 'twenty questions' by carefully phrasing each question to have a 50-50 chance of being answered with a yes or no. Later on when we do 'channel coding' we'll add some carefully designed overhead back in the form of error correction coding (ECC). Even though source coding removes redundancy while channel coding puts it back in, it's important that these two steps be kept separate to maximize the overall performance of the system, as well as its modularity. This philosophy is stressed by Andy Viterbi in an article that can be viewed at

```
http://www.qualcomm.com/people/pkarn/viterbi_lessons.html
```

Presently, most EME communications involves Morse Code. The Morse alphabet was optimized for the transmission of plain text. The characters ETOIANS which have a high occurrence frequency were assigned the shortest symbol length, while infrequently used characters like QYZ were given long symbols. The digits 0…9 were given still longer symbols. But the information in a QSO is more random. For example in amateur calls, all letters have nearly equal occurrence probabilities. An amateur with a short call like N5EE is more "Morse efficient" than an EMEer suffering with a call like ZJ9QYP. In Morse, a dot (counting the separator space) takes 2 bit times and a dash takes 4. Another 2 bit times are required as a character separator. Therefore a call like N5EE is 28 bits long while ZJ9QYP is 94 bits long.

We might consider coding in ASCII so that all characters require the same number of bits. The 7- and 8-bit ASCII sets have symbols we never will use, with both upper and lower-case letters plus punctuation and a lot of control characters. A 6-bit subset (often called Half-ASCII, using the ASCII range with decimal values 32 thru 95) gives upper case characters, numbers and punctuation and is more than adequate. With this character set, a call like ZJ9QYP becomes 36 bits long. Even this character set has symbols which are not needed to conduct a QSO and it is trivial to reduce the required number of bits even further.

All amateur calls consist of a 1 or 2 character prefix (one of which is a letter), a digit, and a 1 to 3 letter suffix. Let's consider all calls as being a 6 character field, and fill any use a space character to pad empty positions. If a field can be a letter character or a space there are 27 possible values. If it can be a letter or digit, there are 36 or 37 possible values depending on whether a space is needed. A digit has only 10 possibilities. Let the 6 characters of an amateur call be denoted *a,b,c,d,e* and *f*. The *a* field can be a digit, letter or space, the *b* field is either a letter or digit, and *c* must be a digit. The *d,e* and *f* fields are either letters or spaces:

	a	*b*	*c*	*d*	*e*	*f*	
Digit [0-9]	10	10	10				
Letter [A-Z]	26	26		26	26	26	
space character	1			1	1	1	28-bit Value
TOTAL POSSIBILITIES	**37**	**36**	**10**	**27**	**27**	**27**	in HEX
Examples:		**W**	**3**	**I**	**W**	**I**	F957F6B
	K	**A**	**9**	**Q**			8935D6F
	J	**Y**	**1**				86D0869
	A	**7**	**3**	**M**	**B**		44F474C
	9	**M**	**2**	**E**	**M**	**E**	40FD27E
		K	**5**	**J**	**L**		F72108B
				Miscellaneous			FA08318
				Functions			- to -
	C	**Q**	**C**	**Q**	**C**	**Q**	FFFFFFF

Note that a space is included in the *d* field. The only call we are aware of that needs this is JY1, but we don't want to exclude King Hussein from EME!

Based on this formulation, the total number of possible calls is 37*36*10*27*27*27 = 262,177,560. Since this is smaller than 2^{28} = 268,435,456, we find that every call can be represented as a 28-bit number. For the examples shown in the table, the 28-bit values for one possible coding are given in hexadecimal notation. The 2^{28} maximum value allows 6,257,896 additional special "calls" which can be allocated as tokens for special functions like ***CQCQCQ*** or ***TEST*** or weird "special event" calls that don't obey the normal rules.

Now we continue to enumerate the bits needed to make a QSO. While we could transmit the 2°x1° Maidenhead Grid Square in ASCII or compress it like we did with the calls, a more bit efficient representation is possible. On the earth there are 180 possible 2° longitude values and 180 of 1° latitude bands, so there are only 32400 possibilities. This can be packed into a 15 bits number (2^{15} = 32768) with 268 cases left over for special tokens (like "***I don't know my grid square. I'm lost!***").

Three bits (8 possible values) seems adequate to send the other station's signal report. This would range from zero indicating that nothing has been heard up to 7 to indicate "*Why are we bothering with this signal processing? Lets go on SSB and talk about the weather!*". This scale is similar to conventional EME practice with 1="T", 2="M", 3="O", 4="5", 5="539", 6="569" and 7="599".

Now let's allocate 4 bits for acknowledgements (Ack) and 4 bits for confirmation (QSL):

Ack Bit 1 = I have copied your call
Ack Bit 2 = I have copied you sending my call
Ack Bit 3 = I have copied a non-zero signal report from you
Ack Bit 4 = I have copied your Grid Square

When the receiving station copies a non-zero Ack bit, he sets the matching QSL bit, (i.e. QSL bit 1 means "*I know you have copied my call*", or bit 4 meaning "*I know you have my Grid Square*"). Thus when a station hears the 8-bit Ack+QSL field filled with all ones, he can quit, equivalent to sending "ROGER SK".

Finally we tally up our assessment of the "raw" bit requirements to make a QSO:

Sending Station's Call	= 28 bits
The other Station's Call	= 28 bits
Signal Report	= 3 bits
Grid Square	= 15 bits
Ack+QSL Bits	= 8 bits
Total	= 82 bits

As a preview of our subsequent discussions on coding, these “raw” data bits would be combined with a large number of carefully chosen error correction bits to make up the message actually transmitted. The resultant message is **x** times the length of the “raw” data and is called a “**rate 1/x**” code. This process is known as convolutional encoding and has been the key to decoding the very weak signals from interplanetary spacecraft like Galileo.

The 82**x** bits of data would be transmitted as a narrow-band spread spectrum signal occupying the ~2kHz passband available with typical SSB radios. As we will discuss later, **m-ary** FSK modulation (with a single one of **m** frequencies being transmitted at a time) looks optimum. The **m** discrete states result in $\mathbf{log_2(m)}$ bits being sent at any instant.

Each second, **B** signalling elements will be transmitted, resulting in a baud rate = **B** and a total data rate of $\mathbf{B*log_2(m)}$ bits/second). The timing of the **B** elements would be precisely synchronized to the UTC second (using GPS timing receivers) so that the receiving station can optimally detect the data (i.e. GPS would provide the clock synchronization function).

Back to the EME QSO and assuming that all 82**x** bits can be sent in a one minute sequence -- the minute-by-minute progress of an “easy” QSO might go something like this (not much different than normal EME practice):

- Minute 0: Station A sends with the Signal, Ack and QSL bits set to zero.
- Minute 1: Station B sends. If he copied A, then he sends a signal report and sets Ack bits 2,3,4
- Minute 2: Station A sends a signal report and Acks everything and sets QSL bits 2,3,4
- Minute 3: Station B sets Ack bit 1 and QSL bits 1,2,3,4
- Minute 4: The QSO is complete, and Station A sends a “courtesy SK” by setting QSL bit 1

As an aside, if the QSO isn’t “easy”, the coding and the fixed message formatting allows data from subsequent transmissions to fill in “holes”.

5. Looking at the Moon:

So far, we’ve only hinted at the hard part of the problem. The transmitter sends a pure CW carrier, it is reflected from the moon and finally received back on the earth. In this process, a lot happens. In this section, we will discuss a bit of physics and astronomy that applies to EME.

First, the signal is severely attenuated – it drops as $1/R^2$ on the way to the moon and it loses again as $1/R^2$ on the way back to earth for a net $1/R^4$ path loss. But amateurs are crazy enough to try to dig weak signals out of the noise. If it was easy, there would be no challenge!

Second, the signal is reflected from a large, rough, moving (nearly) spherical object. From the center of the moon’s visible disk (where the first signal is reflected) to the limb, we can observe up to 11.6 msec of time delay (The radius of the moon is 1738 km = 5.8 msec. The round-trip delay is twice the one-way value). Figure 1 shows contours of the reflection point at delays of 0,2,4,6,8 and 10 msec and the limb at 11.6 msec.

The moon has a prolate spheroidal shape (rather like a football). Thru the 4.5 billion years that the moon has orbited the earth, energy losses due to tidal friction have caused the moon to be captured so that one face is presented to the earth and we never see the backside of the moon. The means that the lunar rotation period is the same as the lunar month.

An exaggerated version of the basic geometry of the moon in its orbit is illustrated in Figure 2. The orbit of the moon is an ellipse with an eccentricity of 0.055. The rotation of the moon on its axis is uniform and keeps the *average* center facing the earth. But since the orbit is elliptical, a point on the surface appears to move thru several degrees of lunar longitude throughout the lunar month. This is illustrated in Figure 2 by a fixed “spot” on the moon as seen at 4 different phases during the lunar month. The total motion of a spot on the moon’s equator amounts to ±7.9° (see for example A.E.Roy, *Orbital Motion*, Hilger Press 1988, pg 288).

In addition, the moon’s orbit is inclined about 5° to the ecliptic, so we can see a bit over the north and south poles of the moon. These effects are known as Optical Librations (as distinguished from the much smaller Physical Librations which reflect small deviations from “smooth” rotation). This apparent wobbling of the moon permits earth-based observers to see about 60% of the total surface of the moon.

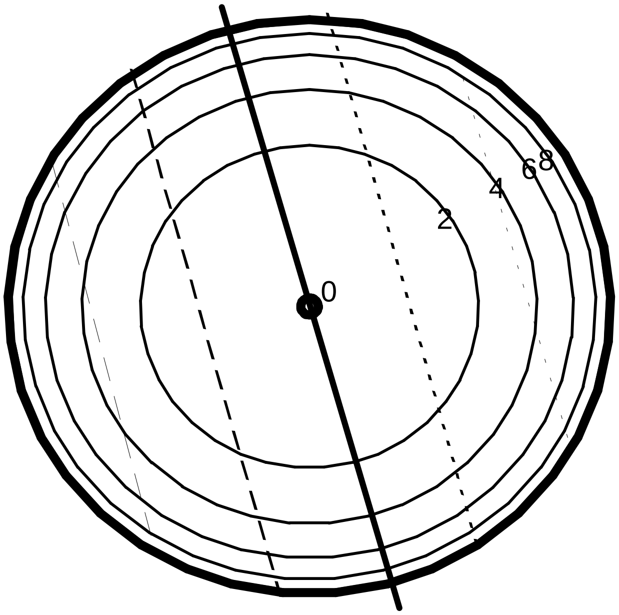


Figure 1: The radar moon, with contours of constant delay an Doppler

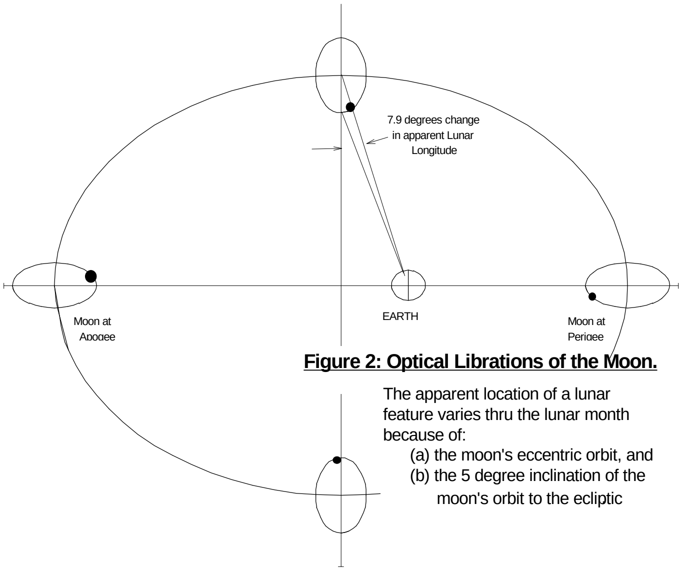


Figure 2: Optical Librations of the Moon.

The apparent location of a lunar feature varies thru the lunar month because of:
(a) the moon's eccentric orbit, and
(b) the 5 degree inclination of the moon's orbit to the ecliptic

The total apparent motion of a spot on the moon by up to ~±10° of lunar longitude and latitude is of considerable importance in EME. At any given time, the moon appears to be rotating slowly and any “hot spot” that gives good EME echoes is changing its range accordingly, giving rise to a Doppler offset. If there are several “hot spots”, the relative phases of the reflected signals changes and the signal can fluctuate wildly. The contours of constant Doppler shift are illustrated in Figure 1. The zero velocity contour is the intersection of the plane defined by the apparent angular velocity vector and the line from the center of the moon to the observer on earth (a great circle on the moon’s surface). Other velocity contours are parallel to this plane and are like the moon was cut by a giant bread slicer. Half of the bread slices are moving towards the observer and the other half away from the observer. The peak velocity at the limb of the moon is:

(Moon Radius = 1738 km) * (Apparent Angular Velocity, radians/second)

As an estimate of this velocity, we note that the position of a spot on the moon moves as a monthly sinusoid with ±8-10° amplitude (depending on the current astrological conditions) corresponding to peak velocities of 60-75 cm/sec with the maxima occurring at apogee and perigee (and minimum libration motion midway between apogee and perigee). The opposite limbs of the moon therefore have relative velocities of up to 120-150 cm/sec. On Figure 1 we have schematically shown ±3 velocity contours, so each contours represent ~20-25 cm/sec intervals at apogee and perigee.

Most of the usable EME signal arrives in the first ~2 msec which Figure 1 shows corresponds to the central half (radius-wise -- it's actually only ~¼ of the *area*) of the visible moon. Over this area we see somewhat less than half of the peak libration-induced velocity (i.e. ~±25-35 cm/second). For the EME case, the 2-way Doppler shift in frequency units (with the ± sign reminding us that half the moon is turning towards us and half is retreating) that we see over the reflecting moon is:

$$\Delta f_{Hz} = \pm 2 * F_{Hz} * \Delta v/c = \pm 0.067 * F_{GHz} * \Delta v_{cm/sec}$$

The signal bandwidth is about 2π larger than the Doppler shift since about 1 radian of random phase shift will cause a loss of coherence. We double this to get the full bandwidth (accounting for the ± velocity range):

$$BW_{Hz} \approx 2*2\pi*\Delta f_{Hz} \approx 0.84 * F_{GHz} * \Delta v_{cm/sec}$$

and, taking the coherence time as the reciprocal of the bandwidth, we also get:

$$\tau \approx 1/ BW_{Hz} \approx 1.2 / (F_{GHz} * \Delta v_{cm/sec}) \approx .04 * \lambda_{cm} / \Delta v_{cm/sec}$$

The lunar surface consists of flat areas (Mare) and mountains. The Mare are covered with dust and strewn with boulders. Any signal we get back from the moon is the vector sum of the voltages from all these regions and the resultant signal is spread in both time and frequency.

In the time domain, most of the signal comes from the area within the 2 msec contours of figure 1. Therefore the signals have a coherence bandwidth ~500 Hz to 1 kHz (essentially independent of signal frequency). This means that the signal fading seen at frequencies ~1 kHz apart are uncorrelated and there it is possible to achieve some improvement with frequency diversity even with the ~2 kHz bandwidth of an SSB radio. This also accounts for the fluttery, hollow sound of EME SSB signals. It also places an upper limit of ~300-500 baud on the signalling rates for EME digital communications. In 1986, W3IWI/KL7 had a "legal" packet radio QSO with W3IWI/KL7 on 70 cm EME using 300 baud FSK (the 2.4 second EME delay allowed the same 85' dish antenna to be used for both transmitting and receiving).

At longer EME wavelengths (like λ=2M), several effects occur:

- 10-20 cm/sec libration velocities mean that signals maintain coherence for several seconds.
- The signal is "reflected" from a region about one wavelength thick, so the dust "smooths" the reflection. Boulders and other small-scale surface roughness is relatively unimportant.
- This gives rise to signals with lots of "glints" much like the reflections from a "discoteque mirror ball". QRP EME on 2M thrives on waiting for a "glint" to happen!

By contrast at frequencies above 1 GHz, the “rules” are different:

- 10-20 cm/sec libration velocities reduce coherence times to small fractions of a second. The signals sound increasingly “rough” as shorter wavelengths are used.
- The surface “roughness” (measured in wavelengths) increases and boulders/rocks become much more important.
- “Glints” become unimportant – it’s the overall average properties that matter.

And the λ=70cm band represents a transition region. Since the typical Oscar satellite station is equipped for 2M and 70 cm operation and since equipment is readily available for these bands, we envision it will be the “home” the signal processing ideas we present here. Since our ideas are based on spread-spectrum techniques, we note that the bandwidth that can be used depends on three factors:

- The added bandwidth due to spectral broadening described above. Physics makes this the ultimate limit!
- intrinsic frequency stability of the radios. Frequency stability can be improved by using high stability frequency standards; in Appendix A we discuss the possibility of using GPS-aided crystal oscillators to achieve performance equivalent to a Rubidium frequency standard.
- The ability of the stations to set their frequencies to compensate for Doppler. It is easy for our computers to calculate the Doppler offsets since both the moon’s orbit, the rotation of the earth and the stations locations are well known; It is only necessary to devise a scheme to “steer” the radio’s frequencies to values predicted by the computer.

All these factors become more serious direct proportion to frequency. Schemes which may work at 2M or 70 cm will probably fail miserably at 10 GHz! The following section will develop a design which ends up with 64-ary FSK occupying a ~2 kHz bandwidth (i.e. channel spacing ~ 30 Hz), symbol rates ~2 baud and some degree of coherent detection suitable for 2M and 70 cm. At λ=13 cm or λ=2.8 cm, we may have to use binary or 4-ary FSK, symbols several seconds long and rely totally on incoherent signal processing.

Now with an idea of the limitations imposed by signal characteristics, we turn to the topics of coding error correction, modulation and signal processing.

6. Modulation and coding for the EME channel:

The design of any modulation and coding scheme must be carefully tailored to the channel. What works well on one kind of channel (e.g., a dialup telephone circuit) may perform poorly or not at all on others (e.g., EME). Although our understanding of the problem is still incomplete, we already know enough to suggest an approach that is likely to work.

a. The EME channel as seen by a communications engineer

The most conspicuous feature of the EME channel is its very high path loss. This is obvious from the large antennas, high power amplifiers and high performance preamps typical of amateur EME. On the other hand, amateur EME signals are usually narrowband; plenty of extra bandwidth is usually available.

So in communications engineering parlance, the EME channel is very much power limited -- as opposed to bandwidth limited -- and just about anything we can do to reduce the required signal power will be worth it. One way to reduce required signal power is to lower the user data rate. But we should first maximize the efficiency with which we use our power.

We can do this by trading off bandwidth -- which we have in abundance -- for scarce power. This tradeoff comes from Shannon's famous channel capacity formula:

$$\mathbf{C = B * \log_2(1+S/N)}$$

where C is the channel capacity in bits/sec, B is the channel bandwidth, S is signal power and N is noise power. As long as we do not try to exceed the capacity C of a channel, it is theoretically possible to communicate with an arbitrarily low error rate. Above C, it is simply not possible to do so.

Note the relationship between B and S/N; to a certain extent we can compensate for a lower S/N by ***increasing*** B. Now this may be counterintuitive to many of you; after all, it's standard practice to use the narrowest available filter when copying an especially weak CW signal. But it is very much true *provided that we carefully design a wideband signal and build a matched receiver filter.*

Since noise power is directly proportional to bandwidth, Shannon's formula can be rewritten as

$$\mathbf{C = B * \log_2(1+S/N_0B)}$$

where N_0 is the noise spectral density in watts per hertz. So as B goes to infinity we reach a point of diminishing returns. But more bandwidth is always beneficial, even though the incremental benefit may be small.

A way to rewrite Shannon's law that better expresses the tradeoff between bandwidth and power is as follows:

$$\mathbf{E_b/N_0 > B/R * (2^{R}/B - 1)}$$

Here we've introduced two new variables: R, the actual signalling rate in bits/sec and Eb, the energy per user data bit measured in watt-seconds (joules). N$_o$ is again the noise spectral density in W/Hz, but this too has units of joules. This formula says that the minimum required Eb/N$_o$ ratio is a direct function of the available bandwidth relative to the data rate. The more bandwidth, the lower the necessary E_b/N_0 ratio and vice versa. But the relationship is nonlinear; even with infinite B/R, Eb/N$_o$ must always be greater than ln(2). Expressed in decibels, this is -1.6dB -- the famous Shannon Limit.

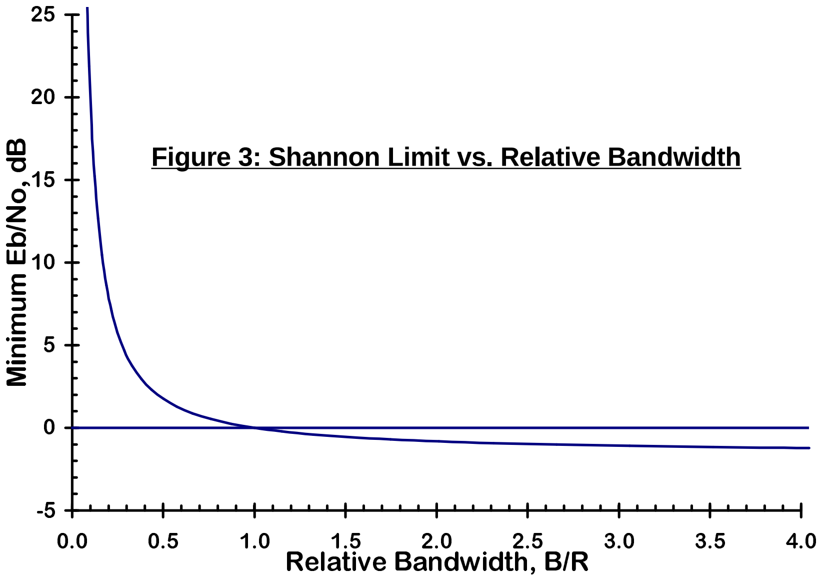


Figure 3: Shannon Limit vs. Relative Bandwidth

The ratio Eb/N$_o$ is an absolutely fundamental one in digital communications, as it directly expresses the power efficiency of a coding and modulation system. That is, it concentrates on what really counts: how much RF energy does it take to send each bit of user data over a channel with a certain noise spectral density?

b. Fading:

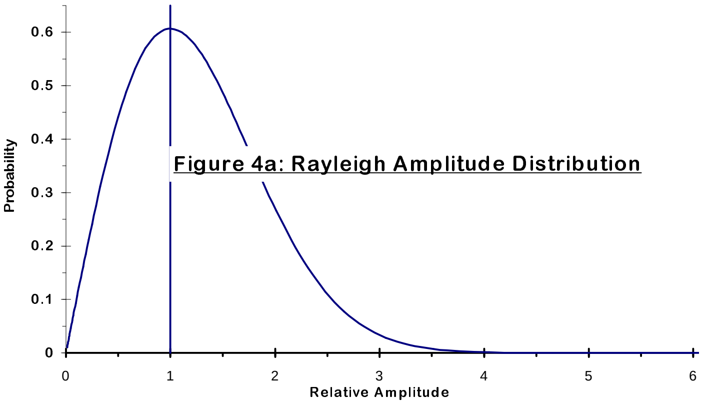


Figure 4a: Rayleigh Amplitude Distribution

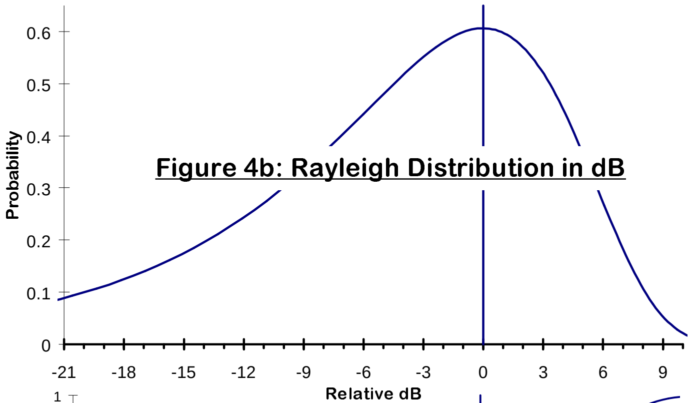


Figure 4b: Rayleigh Distribution in dB

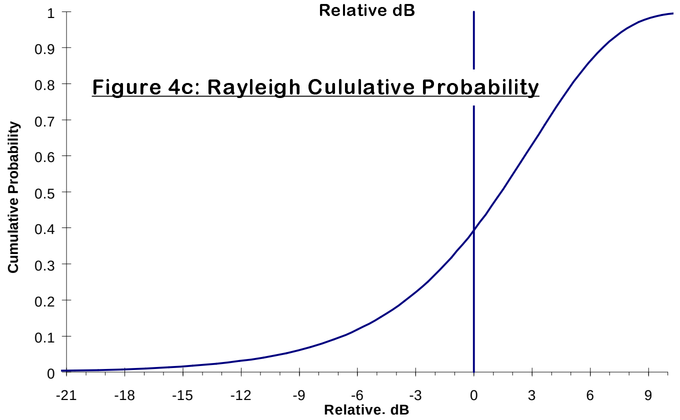


Figure 4c: Rayleigh Culultive Probability

Another important characteristic of the EME channel is that it fades. The signal takes many slightly different paths as it reflects off the uneven, slowly librating lunar surface. It can also split into multiple paths as it passes through the earth's ionosphere or troposphere. Because these paths vary in relative length, the various signal components arrive at the receiver at slightly different times and with different phases. The receiver "sees" the vector sum of these components. Since their relative path lengths change with time as the moon librates, sometimes they add to produce an enhanced signal. At other times they cancel, producing a degraded signal. At low antenna elevation angles, multipath reflections (ground reflection gain) from the surface of the earth are also important.

It turns out that the precise physical mechanisms that produce multipath fading aren't really that important. Somewhat different mechanisms produce the multipath fading seen in land mobile radio, but it resembles EME in that many separate paths are often present. And when there are many reflection paths without any direct line-of-sight paths, we have a "Rayleigh fading channel". That is, the signal amplitude distribution follows a Rayleigh probability function and the signal phase is uniformly distributed at all angles.

This doesn't quite tell the whole story. If you sample a fading signal, wait a very short time and sample it again, the two measurements will tend to have similar amplitude and phase, i.e., they are correlated. If you wait a long time between measurements, they will tend to be different, i.e., uncorrelated. The time scale beyond which a pair of signal measurements tend to become uncorrelated is the "channel coherence time". It depends on the relative physical velocities of the various objects involved in reflecting or refracting the RF energy. It also depends on RF wavelength. Faster velocities and shorter RF wavelengths produce shorter coherence times, i.e., "faster" fading. For example, with all other things constant, mobile FM "flutter" and EME fading are both typically 3x faster on 70cm than on 2m. As an extreme example, aurora scatter has extremely short coherence times because of the very high apparent "velocities" of the ionized cloud particles.

Coherence times on 2m EME are typically several seconds, but this can vary over the month and with ionospheric or tropospheric activity. Because the multipath components arriving at the receiver travel many different distances, their relative phase (and whether they add or cancel each other) depends on the precise wavelength. A small change in frequency may be enough to cause two paths that formerly canceled to enhance each other, or vice versa. The frequency scale over which this occurs is called the "coherence bandwidth".

Coherence time depends on the relative velocities of different parts of the propagation medium with respect to the observer: e.g., lunar libration, which varies with time. On the other hand, coherence bandwidth depends on the size of the propagation medium: the larger the medium or object, the wider the possible range of multipath delays that can occur. Longer delay differences between multipath components mean that smaller frequency differences are enough to comprise a significant fraction of a wavelength that can turn a peak into a null and hence the narrower the coherence bandwidth.

As previously discussed, the maximum EME delay spread possible given the moon's diameter is 11.6 milliseconds, corresponding to a coherence bandwidth of about 86 Hz. If the lunar surface reflected RF the way it reflects light, this would indeed be the coherence bandwidth; but in fact it's more like 500-1000 Hz on the lower EME bands.

Why the discrepancy? Because the lunar surface reflects RF differently than it does light. At optical wavelengths, the full moon appears uniformly bright (discounting albedo variations between maria and highlands) because, thanks to the fine layer of dust covering its entire surface, the lunar surface is extremely rough at scales corresponding to visible wavelengths. And the moon's surface is fairly rough at large scale (much longer than VHF/UHF wavelengths) due to mountains, craters and the like. But in between, the moon is relatively smooth over a fairly wide range of radio wavelengths. So instead of uniformly scattering RF at all angles, the lunar surface tends to reflect it more specularly; the center of the disk appears brighter than the limb. But the limb isn't totally dark because of the large scale surface undulations. So the moon can be best seen as a collection of many independent specular reflectors, something like a dance hall mirror ball where the glue holding the mirrors on has softened and allowed the mirrors to shift their angles with respect to the surface.

So the net result is that the delay spread is somewhat shorter (and the coherence bandwidth wider) than it would be if the moon scattered RF as uniformly as it does light. The coherence time is somewhat longer too, thanks to the smaller contribution from the more rapidly moving limb. On the other hand, the smaller reflecting area means a higher average round trip path loss.

In the discussion that follows, it's important to remember that fading affects signal phase as well as amplitude. It is typical to see very rapid phase shifts (approaching 180°) across a deep fade. This is easiest to understand by example. Consider a path with just two multipath components of nearly equal amplitude and 180° out of phase. If the first component starts at a lower amplitude than the second component but then increases to where it becomes the stronger, it's easy to see how the amplitude of the vector sum of both components goes through zero and how the phase will suddenly jump 180° during the transition.

c. Coherent vs. noncoherent (sometimes called incoherent) demodulation:

A receiver always works best when it has a signal carrier phase reference. With it, noise in phase quadrature to the desired signal can be ignored. But a receiver without a phase reference must respond to all incoming signal phases. The benefit is slight at high SNR (where the signal "swamps" the noise) but can be large at low SNR. This is why AM envelope detectors have a threshold effect on weak signals while SSB receivers do not. (Noncoherent FM demodulators have an even more pronounced threshold effect.)

[Note from KA9Q: see diagram of signal vector plus noise for large and small signal case.]

[Note from W3IWI: Phil and I were working on this paper right up to the last minute and it wasn’t possible for us to get the figures merged with the text. Sorry!]

Some digital modulation schemes require the receiver to maintain a stable carrier phase reference over relatively many data bits. A good example is binary phase shift keying (BPSK), used for telemetry on the AMSAT Phase III satellites and on the low-earth-orbiting PACSATs. Since the BPSK signal suppresses the actual carrier, the receiver must reconstruct a local carrier reference from the data sidebands. (BPSK is just suppressed carrier DSB-AM with digital modulation.)

Two essentially equivalent methods are commonly used to generate a BPSK carrier reference: the Costas loop and the squaring loop. Both use a nonlinearity to regenerate the suppressed carrier: an I-Q multiplier in the Costas loop or a signal squarer in the squaring loop. This necessarily degrades the SNR of the recovered carrier by 6dB for binary PSK, 12 dB for QPSK and more for higher orders. This is generally not a problem when signals are strong, the channel is stable and relative frequency uncertainty is small, as in high speed line-of-sight links. One simply narrows the loop filter enough to raise the recovered carrier SNR to the desired level. I.e., the loop "smooths" its carrier phase estimate over many data bits. This works well as long as the channel coherence time (as well as other sources of phase noise, such as local oscillators) is long with respect to the reciprocal of the loop filter bandwidth. In a typical system designed for a relatively high SNR, the loop filter bandwidth might correspond to 30 bit times.

[Note: see diagram of costas loop with loop filter highlighted]

Things get difficult when strong FEC is used to get the most out of the limited signal power. At such low "raw" SNRs (or more precisely, Es/N0, the ratio of energy per channel symbol to the noise spectral density) an additional degradation called "squaring loss" appears. To compensate, the loop filter must estimate carrier phase over an even longer period. For example, KA9Q's 1200 bps experimental QPSK satellite modem estimates carrier phase over several hundred bit times, and yet the modem's overall performance is still limited by the noise that gets through this very narrow filter.

Things only get worse as we scale to lower signal powers and data rates. At 1 bps, several hundred bit times is several hundred seconds. This corresponds to a carrier loop filter bandwidth of less than .01 Hz! Even if the frequency instabilities in an amateur EME system could be kept below this level, we are faced by the insurmountable fact that the EME channel simply won't stay still for more than a few seconds -- the fading will "modulate" the signal right out of the carrier filter bandwidth. So it's pretty clear by now that conventional suppressed carrier BPSK with coherent demodulation is probably not a good choice for EME.

d. Coherent demodulation with a pilot carrier:

One way to mitigate the fading problem with BPSK is with a "pilot". That is, you put some fraction (e.g., 10%) of the total RF power into a residual unmodulated carrier (the pilot) for use by the demodulator in regenerating a local reference (e.g., with a narrow bandpass filter or PLL). You do this by phase modulating the carrier with less than +/-90 degrees of phase shift; the resulting signal is equivalent to a suppressed carrier BPSK signal plus an unmodulated carrier in phase quadrature.

[Notes: see diagram of signal vectors] [see diagram of loop with simple PLL tuned to carrier]

Because this carrier component can be recovered without a nonlinearity, there are no losses. Of course, the pilot is only part of the total signal power, and this power can't carry data. So there is an optimum pilot power fraction that depends on the situation.

Pilots are sometimes used even on channels that nominally don't fade. NASA has long used them on deep space links where signals are weak, data rates are low and relative frequency uncertainties are so high that the inherent losses in regenerating a suppressed carrier are just too great. Pilots also help deal with the small amounts of multipath fading that occurs when signals have to pass through planetary atmospheres or through the solar corona.

e. Noncoherent demodulation (especially m-ary FSK):

Many other alternatives exist when the receiver can't maintain a stable, long term carrier phase reference.

BPSK can be differentially detected by simply multiplying the previous symbol by the current one, in essence using just the previous symbol as a carrier reference. Unfortunately this doesn't perform nearly as well as coherently demodulated BPSK, especially at low SNR, because only one bit's worth of energy is used as the reference.

[Note: see diagram of differential BPSK demod]

The most popular noncoherently demodulated signaling method is binary frequency shift keying (FSK). The demodulator simply measures the total signal energy -- ignoring phase -- over the bit time for each of the two "tones" and picks the one with the greater energy as its decision. This too has worse weak-signal performance than coherent BPSK.

But it turns out that frequency shift keying -- even with noncoherent detection -- can be made more power efficient by simply adding tones. Consider 4-ary FSK - i.e., FSK with four "tones". Only one tone is sent at a time, each representing log2(4) = 2 user data bits.

The 4-ary FSK demodulator is just an extension of the classic binary (2-ary) FSK demod. It has 4 filters and envelope detectors, one for each tone. At the end of each tone pulse (called a "symbol") the demodulator picks the filter channel with the greatest accumulated energy and outputs the 2 data bits that correspond to that symbol.

[Note: see diagram of 4-ary FSK demod]

Why does this scheme use less power per user data bit than binary FSK? Because each symbol represents 2 user data bits, we can afford to invest twice as much RF energy in each channel symbol as in binary FSK while maintaining the same energy per user data bit (Eb). Raising the symbol energy reduces the chance that a random noise peak in the three "incorrect" channels will cause it to be chosen over the correct channel. At reasonably low bit error rates, the improvement is dramatic; 4-ary FSK requires an Eb/N0 of 11.5 dB to maintain a 10^{5} BER, and that's nearly 3dB than binary FSK. 8-ary FSK requires 9.5 dB, and that's slightly less than perfect BPSK!

But as the number of tone channels M increases, the improvement slows down. The probability that any one tone channel will cause an error is unchanged; it's just there are so many more of them and just one can cause an error. Also, the symbol energy increases only as $\log_2$(M), so it falls further behind M.

[Note: see diagram of BER vs Eb/N0 curves vs M]

The bottom line is that as the number of channels M approaches infinity, M-ary FSK approaches an Eb/N0 of -1.6 dB. This is exactly the famous Shannon limit for infinite bandwidth. In fact, this was essentially the system that Shannon analyzed in his landmark 1948 paper on noisy channel capacity.

In a real system, of course, we have to pick a finite M and layer additional forward error correction coding on top of it. How large can we make M? There are two limits: complexity and bandwidth.

Complexity is no longer a practical problem. The British Foreign Office "Piccolo" system of the early 1960s used 32-ary FSK with each symbol representing one of the 32 letters in the Baudot alphabet. The Piccolo demodulator consisted of 32 separate hardware filters, one for each tone. That's obviously rather cumbersome.

Today we can use the Fast Fourier Transform (FFT) to build the receiver filter bank for just about any value of M we like. FFTs with thousands of points (bins) are no sweat on modern microcomputers. We just take the FFT of the received symbol in the time domain, square and add the real and imaginary components of each frequency "bin", and pick the bin with the largest energy.

The other limit is the bandwidth needed for all the tones. As M increases, the tone signalling rate decreases as $\log_2(M)$ because that's how many data bits are represented in each tone. Since the tones must be at least 1/T Hz apart, where T is the symbol time in seconds, we can space the tones somewhat more closely together as we increase M thus partially compensating for the extra tones. So the overall bandwidth requirements increase as $M/\log_2(M)$.

[Note: see frequency diagram of tone spacing]

It's interesting to note that the relative bandwidth required for 4-ary FSK is $4/\log_2(4) = 2$, the same as 2-ary FSK: $2/\log_2(2) = 2$. In other words, "the first one is free", much like the transition from binary PSK to QPSK (4-phase PSK). (QPSK can carry twice the data in the same bandwidth as BPSK with the same Eb/N0. But higher order PSK constellations require greater Eb/N0 for the privilege of further increasing the "bandwidth efficiency", just as higher order FSK constellations require greater bandwidth to save power.)

Some final comments. M-ary FSK is just one example of what is more generally known as a "M-ary orthogonal signal set". A set of properly spaced tones (sine functions) is just one example; others include Walsh functions (the Qualcomm CDMA digital cellular system uses 64-ary Walsh code modulation on its mobile-to-base link) and the pulse-position modulation (PPM) codes often used on optical fiber. The theoretical performance is exactly the same for all these schemes, but sine waves (i.e., FSK) have the practical advantage of maintaining orthogonality without precise time synchronization.

Also, the term "noncoherent detection" is slightly inaccurate here, even though it's standard in all the textbooks. It would be more accurate to say that detection is "noncoherent from symbol to symbol". The tone filter/detector is still very sensitive to tone phase ***during*** a symbol interval. For example, an incoming tone that suddenly phase flipped 180 deg degrees exactly halfway during the symbol interval would produce a zero output. To minimize this effect, the the signalling interval must be kept short relative to the coherence time of the channel. This creates another practical limit on M.

Here's another way to look at it: channel fading AM modulates the signal, and the receiver tone filters must be wide enough to capture the sidebands so generated. That puts a minimum bandwidth limit on the filter, which corresponds to a maximum coherent integration time.

f. FEC with coherent and noncoherent modulation:

The "rate" of a forward error correction (FEC) code is the ratio of the user data bit rate to the encoded channel symbol rate. For example, a convolutional code that produces two encoded symbols for every user data bit would be a rate 1/2 code, as would a block code that produces 24 encoded symbols for every 12 user data bits.

Many FEC schemes exist. Some, like Reed-Solomon (RS) block codes, are well suited to M-ary orthogonal modulation because they use non-binary symbols. 8-bit RS symbols are particularly popular, being used in the Compact Disc. Non-binary convolutional codes do exist (the "dual-k") codes, but it's also possible to adapt binary codes to the purpose.

By the way, it's important to remember that when FEC is discussed, Eb/N0 ratios always apply to the original user data bits and not to the encoded, redundant symbols. For example, a rate 1/2 K=32 convolutional code can operate with an Eb/N0 down to about 3 dB. The Es, or energy per FEC encoded symbol is 1/2 (3dB) less, or 0 dB. For a rate 1/4 code the Es/N0 is 6dB less than the Eb/N0, and so on.

When FEC is layered on top of an ideal coherent modulation scheme like BPSK, lowering the code rate always improves the coding "gain" (i.e., it decreases the Eb/N0 needed to attain a certain bit error rate). But this is no longer true when FEC is combined with M-ary noncoherent demodulation. Below a certain rate, the Eb/N0 requirement actually increases again.

[Note: see diagram showing Eb/N0 vs code rate for AWGN channel]

Why is this so? A coherent demodulator is a linear device. Postprocessing of a noisy signal can improve it by coherently combining very small signal components that are still spread over time or frequency. But a noncoherent demodulator is a nonlinear device, and it has a threshold effect much like that of an FM discriminator. (We can actually think of M-ary FSK as FM with M discrete quantized steps). Above threshold, the SNR in the filter bandwidth is high enough that the accumulated signal energy swamps the noise. But below threshold, the output SNR decreases more rapidly than the input SNR.

So as we lower the FEC code rate while keeping the user data rate constant, the symbol rate coming out of the encoder increases. As the encoder symbol rate increases, we need to increase the signalling rate on the channel (the encoded FEC symbols become the data bits to the M-ary modulation scheme). And because we're spreading our fixed transmitter energy out over many more channel symbols, the energy per symbol must decrease. If it decreases below threshold, the demodulator losses increase faster than the coding gain, and overall performance suffers.

The optimum code rate depends on M and whether there is fading. On a nonfading channel, the optimum code rate is about 1/2 for a very wide range of M. On a fading channel, however, the optimum code rate is much lower, typically ranging from 1/10 to 1/3 depending on M.

[Note: see diagram showing Eb/N0 vs code rate for Rayleigh channel]

Why is this? First of all, we must understand that Eb/N0 ratios given for a fading channel refer to averages. Sometimes the signal is much greater than average, and sometimes it's much worse. When it's much worse, we pretty much have to write off our energy investment because it's so far below threshold that even a slower channel symbol rate wouldn't help. But when the signal peaks well above average, we also waste energy because each symbol carries much more energy than is necessary for reliable demodulation.

By sending at a higher channel symbol rate, we can take better advantage of these peaks when they occur. But we don't want to go too fast lest we almost never get peaks that are strong enough. So the optimum code rate is higher than for a nonfading channel.

g. Diversity:

To paraphrase a real estate agent, the three most important things in the design of a communication system for a fading channel are diversity, diversity and diversity. This is simply another way of saying that you shouldn't put all your bit "eggs" in one basket. You should spread them over as many physical dimensions as possible: in frequency, in time and in space. In so doing, you exploit the same "law of large numbers" that enables insurance companies and casinos to, on average, make money. The basic idea is that while one frequency channel, time slot or physical path may be in a fade, another may be at its peak or at least somewhere in between. With diversity you get something more like the channel's average performance all the time.

The really remarkable thing is just how close in practice it's possible to come to this ideal. With the right coding and modulation and enough bandwidth, it is fairly straightforward to achieve an average operating Eb/N0 of only 6-7 dB on a Rayleigh fading channel. The same system might require only 3-4 dB on a nonfading channel, a "fading penalty" of only 3 dB.

Using FEC to add redundant symbols is one particularly effective way to exploit diversity in the time domain. It can also add diversity in the frequency domain if the value M is sufficiently large, which essentially constitutes spread spectrum. In fact, diversity is so powerful that the very simplest "FEC scheme" -- simple repetition of the user data -- can provide dramatic gains if the "diversity order" is high enough. (I'm sure EMEers know this instinctively, given their emphasis on repetition). In practice, though, the best systems use true FEC because the improvements come more quickly.

That the optimum FEC code rate for a fading channel is lower than for a nonfading channel is another example of diversity in action -- spreading the energy budget for each user data bit as thinly as possible in time. In the ideal signal, every instant in a transmission would be a function of every user data bit in the message. As long as you get a certain minimum fraction of the total transmitted energy in a message, you could decode it; the precise fading pattern wouldn't matter. We can come surprisingly close to this ideal.

Obviously, if we could tell when (or where) the channel is going to be at its peak and send at high speed just at those times we could avoid wasting a lot of energy on deep fades. But this is much easier said than done. Even with the slow fading seen on 2m EME, the long round trip delay to the moon and back means that by the time we detected a peak and let the other station know about it, it would probably be gone. So we have no real choice but to hedge our bets and diversify.

h.Interleaving:

Certain FEC codes are limited in their ability to "spread out" (i.e., diversify) the effect of each user data bit widely enough in time. A long deep fade may take out all of the coded symbols affected by a single data bit, causing it to be lost. This is particularly true for convolutional coding, which otherwise has the significant advantage of being adaptable to "soft decision" decoding. (Long block codes like the Reed-Solomon code don't have this problem, but they are not readily adapted to soft decision decoding).

It turns out that a very simple trick can nearly solve this problem. Instead of sending the FEC encoded symbols in the exact order they are sent, you can scramble them in time before transmission and put them back in the right order at the receiver before decoding. A long fade (burst error) is then transformed into widely scattered short errors that the code can easily handle.

The main parameter for an interleaver is the "span", i.e., the range over which adjacent symbols are scattered in time. It's important that the span be considerably larger than the coherence time to maximize the benefit. Memory being cheap, the only real limit in practice on interleaving span is the maximum delay the user will allow. In a packet-like environment, interleaving can and probably should be over the entire packet.

[Note: see diagram of typical block interleaver]

7. Summary and Recommendations for EME:

So putting all this together, what kind of modulation and coding schemes can we recommend for the EME channel?

The overall structure is now clear: efficient source coding of the EME message, as discussed earlier, a forward error correction encoder, an interleaver and a M-ary FSK modulator feeding the transmitter. All these steps up to the generation of the transmitted audio waveform can be done in software on a PC driving a standard sound card.

[Note: see diagram of overall scheme]

The receiver performs the reverse steps: demodulating the M-ary FSK with a fast fourier transform (FFT), deinterleaving, decoding and display in readable text for the operator.

Starting from the bottom up, the allowable FSK symbol time is upper bounded by the channel coherence time (typically a second at 70cm, several seconds at 2m) and lower bounded by the multipath delay spread (a few milliseconds at 2m and 70cm). These parameters are independent of station performance, so they represent hard limits.

The minimum symbol time is also bounded by the need to attain sufficient SNR to meet detector threshold. It is also lower bounded in practice by how accurately we know the beginning and end of every symbol. But with GPS providing a sub-microsecond timing reference and computer programs to compute round trip delay to the same accuracy this is not likely to be a limiting factor.

Let's consider the typical 70cm Oscar station described earlier. For transmit power = 150W, antenna gain (TX and RX) = 19.1 dBi, and an average path loss of 261.2 dB, the average received power will be -201.2 dBW (-171.2 dBm). For a system noise temperature of 100 K, the receiver N_0 will be -208.6 dBW/Hz, giving a C/N_0 ratio of +7.36 dB-Hz. This is much too weak to detect by ear. [see link budget table]

On a Rayleigh fading channel, a strong rate 1/10 FEC code, sufficient interleaving and 64-ary FSK with soft decision decoding needs an average Eb/N0 of about 6.5 dB. That means our link can support an average user data rate of 7.36 - 6.5 = +0.86 dB-bps, or about 1.2 bps. If the message is 82 bits long, it will take about 67 seconds to transmit.

So let's round down to 1 bit/sec. The rate 1/10 FEC will produce 10 symbols/sec. The 64-ary FSK will consumes 6 of these symbols per tone symbol, since log2(64) = 6. So the FSK symbol interval will be 600 milliseconds.

This is uncomfortably close to our upper bound imposed by the channel coherence time, so we might consider shortening it somewhat (W3IWI suggests 500 msec to make the GPS timing synchronization task “cleaner”). If we doubled the FEC code rate to 1/5, this would double the FSK symbol rate (halving the symbol length to 300 ms) (and W3IWI suggests 333.333 msec to keep the bit timing locked to UTC) at the expense of perhaps a half dB in Eb/N0 performance.

To achieve orthogonality, tone spacing for 300 ms FSK symbols must be a minimum of 3.33 Hz. If M=64, the minimum total bandwidth is therefore 213.3 Hz. So to simplify the implementation we could space the tones considerably farther apart than necessary and still fit them all into a SSB transceiver bandwidth. Or we might consider larger values of M, which could lower Eb/N_0 requirements somewhat.

8. Conclusions:

There is clearly a lot of room for experimentation and fine tuning here, but we are confident the system just described should work as advertised. These techniques are not new; they have been known, studied and used (mainly by the military on HF) for decades. More recently they have also been applied to digital cellular telephony in the Qualcomm IS-95 CDMA system. The IS-95 reverse link uses 64-ary orthogonal Walsh code modulation in combination with rate 1/3rd K=9 convolutional coding and soft decision Viterbi decoding. It also uses direct sequence spread spectrum. IS-95 has been heavily field tested where it performed very closely to theory: an Eb/N_0 of about 4 dB on a nonfading channel and about 7 dB on a fading channel. This latter figure is due to the use of rate 1/3rd coding, chosen as a compromise between the fading and nonfading case, and a limited interleaver span of only 20 milliseconds imposed by real time voice delay constraints. So we're confident that with a little tuning our scheme can easily achieve or exceed these performance figures on the EME channel with a typical 70cm Oscar-class station.

It is interesting to consider scaling this modulation scheme up to the capabilities of full scale EME stations. The link calculations performed for the EME station described at the beginning of this paper show that a continuous user data rate of 300bps should be possible. However, there are scaling problems caused by the immutable link parameters such as coherence time and bandwidth, plus somewhat more changeable parameters such as transceiver bandwidth.

Since a better link could support a faster MFSK symbol rate, we could move away from the limit imposed by coherence time. In fact, we could use this margin to increase M and further lower our Eb/N_0 requirements. But even if we keep M=64, scaling this scheme up to 300 bits/sec would need almost 64KHz of bandwidth, well beyond the range of most transceivers and PC sound cards. To stay within these limits we'd have to decrease M and sacrifice Eb/N_0 performance somewhat. Also, with faster symbol times we'd need more precise symbol timing, though with GPS and accurate real-time computer calculations of moon and station position this might not be hard to do.

A more fundamental problem is the inherent EME delay spread of several milliseconds. If the symbol interval is shortened to where the delay spread becomes an appreciable fraction, intersymbol interference appears. The best way around this is to frequency hop or chirp from symbol to symbol to allow time for the limb reflections on a given frequency to die down before it is used again -- and this means still more total RF bandwidth.

So it seems pretty clear that the potential of a conventional (large) EME station can be fully realized only with a true spread-spectrum signal.

On the other hand, scaling this technique to even smaller stations runs into more fundamental problems. The big barrier is the channel coherence time; with our parameters we are already perhaps too close to it. There would be no choice but to limit symbol times to less than desirable values. We could keep M=64 and reduce the FEC code rate to that which the link can support, and we could also increase M. But in either case, the FEC code rate would probably have to be well below the optimum for the corresponding value of M on the Rayleigh fading channel. The FEC decoder would then be in the position of noncoherently combining symbol energy in a less than optimum manner, raising the overall E_b/N_0 requirements and further lowering the attainable user data rate. The system would work, but not nearly as efficiently. And it would certainly tax the patience of most operators since our scheme necessarily gives no real feedback to the operator until after an entire transmission has been transmitted and received.

July 24, 1996

Tom Clark, W3IWI
clark@tomcat.gsfc.nasa.gov
-or-
w3iwi@amsat.org

Phil Karn, KA9Q
karn@qualcomm.com
-or-
ka9q@amsat.org

Appendix A:
GPS for Timing and Oscillator Stabilization

Tom Clark, W3IWI

TIMING: The techniques described in this paper depend on being able to decode the digitally encoded data without the need for extracting a timing clock from the weak signals. Fortunately, GPS has given us the ability to do this reliably and at low cost.

For the past several years I have been experimenting with GPS timing receivers. In part this was driven by a need high accuracy timing in my "real life". As a scientist I participate in the world-wide network of radio telescopes used for Very Long Baseline Interferometry (VLBI). At each VLBI station we use a Hydrogen Maser frequency standard to provide a "flywheel" so that we can coherently cross-correlate weak signals from Quasars received at stations thousands of km apart at frequencies up to 100 GHz. The H-Masers used as frequency references have relative stability of $\sim 1:10^{15}$ at time scales of hours. To cross-correlate the noise signals recorded on magnetic tape, we need to keep track of the relative timing epoch at each station to a few tens of nsec. And the measurements need to have "absolute" time-tags tied to UTC(USNO) at levels ~100 nsec.

I was also driven to this topic because I can forsee the day when we, as amateurs, make use of *a priori* timing of digital signals to improve performance. And I found it fascinating to really dig deeply into how GPS works and how well small GPS receivers can perform.

These efforts led to my development of the "**Totally Accurate Clock**" at home in my basement. As a bottom line, using a GPS receiver as a timing clock we can easily achieve RMS timing **accuracy** and **precision** at levels of ~30 nsec RMS with hardware that can be easily duplicated for < $500. The "**TAC**" name was chosen with tongue-in-cheek:

- Years ago, Heathkit sold a WWV receiver they called the "Most Accurate Clock". With GPS I was able to achieve more than 10^{4} better than the old "MAC", so "Totally" seemed like a good "40+ dB" adjective (I also note that Hammaker-Schlemer and Sharper Image sell the German Junghans WWVB/DCF77 VLF clock with "World's Most Accurate Clock" advertising even today).
- And **TAC** are my initials!

The original **TAC** was based on the Motorola PVT-6 (now called ONCORE®) OEM receiver. After working with Motorola to correct some firmware errors a couple of years ago, I constructed the first prototype **TAC** package and learned how to use it optimally. I then showed it off to some VLBI colleagues and was deludged with requests for clones. To date we have more than 40 **TAC**s operating all around the world.

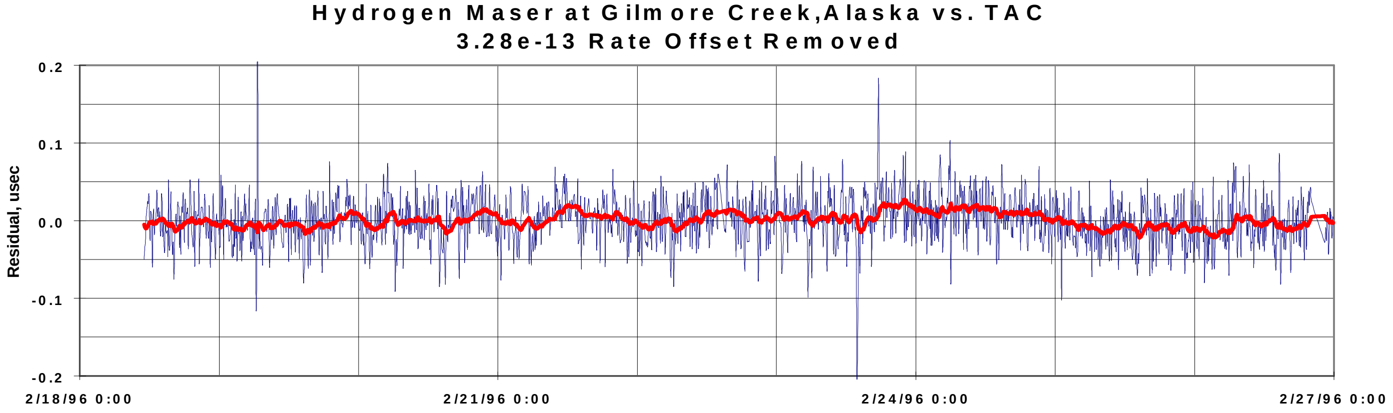


Figure A-1 shows the typical performance of the TAC over an 8-day period; in this case the data is from the Gilmore Creek Geophysical Observatory near Fairbanks where my group operates a VLBI station with an 85' telescope (this is the W3IWI/KL7 EME QTH). The plot compares 100 second averages of timing pulses from the TAC with the GCGO H-Maser with ~31 nsec RMS agreement between the two clocks. Thru the middle of the trace is a 2-hour running average showing 9 nsec RMS performance.

This performance is achieved ***DESPITE*** the DoD's imposition of SA (Selective Availability) on the GPS satellite signals (SA involves the intentional degradation of the performance of the Cesium frequency standards onboard the GPS satellites). The performance improvement is obtained by using multiple satellites (SA is incoherent between the different GPS satellites), by time-domain averaging and by not solving for position of the observer.

An early criticism of the **TAC** was that the PC-based software Motorola provided was not as "user friendly" and "turnkey" as people wanted, and the **TAC** was an ugly gray box that didn't look like a clock! So to support the **TAC** I also wrote a controller program called **SHOWTIME**.

The details of the ONCORE-based **TAC** and the **SHOWTIME** software are found on my anonymous FTP server at URL:

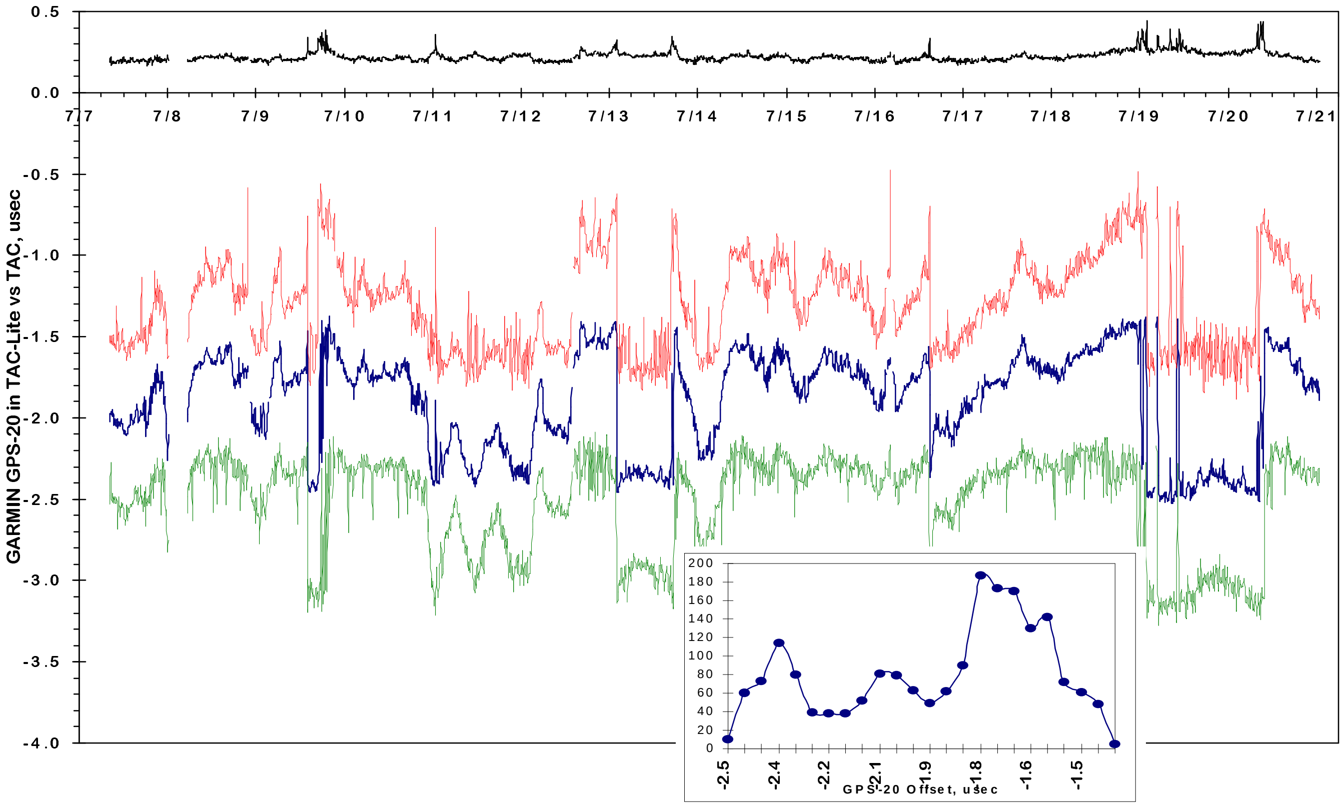
```
ftp://aleph.gsfc.nasa.gov/GPS/totally.accurate.clock/
```

An Email exploder called gps-timing has grown up to support different applications of low-cost GPS timing widgets. You can see all the activity with your news reader (like Netscape or Mosaic) by establishing

```
news://aleph.gsfc.nasa.gov
```

as a news server and then "subscribing" to the gps-timing list.

The Motorola ONCORE is an EXCELLENT timing receiver, but the Motorola "engine" costs ~$400 in small quantities when equipped with the mandatory timing "Option A". I wanted to see if there was a lower-cost



alternative for amateur applications. After some initial tests on Garmin GPS-20 board (costing less than $200) I found it was a fairly good clock at the few hundred nsec level and dubbed the new receiver the "**TAC Lite**".

The plot above shows a 2-week period comparing the **TAC-Lite** (running in "2-D" height fixed mode) with the **TAC.** The Garmin's 1PPS signal seems to have a consistent ~2 usec (early) bias with respect to the ONCORE (which is known to have < 50 nsec offset to UTC(USNO)) as seen in the middle trace. The middle trace is bracketed by (± ~0.5 usec above and below) with the extremes seen on any single 1PPS tick. The middle trace (mean) represents 10 minute (600 second) averages. The RMS of the 600 individual 1PPS "ticks" is shown on the top trace, typically 200-250 nsec. The Garmin's performance is nearly an order of magnitude poorer that the ONCORE and it shows unexplained jumps of up 0.5 usec. But the performance is probably adequate for most amateur applications.

Bob Bruninga (WB4APR) and I proposed to TAPR (Tucson Amateur Packet Radio) that the GPS-20 would be of interest to the amateur community and TAPR has made them available thru a group purchase. For details on obtaining a GPS-20, see the TAPR Home Page at URL:

`http://www.tapr.org`

My **SHOWTIME** software has been expanded so that it handles both the Motorola and Garmin receivers.

My **TAC** hardware implementation uses a small, add-on circuit board that provides precision low-Z 1PPS outputs, an RS232 1PPS signal (on the DCD line) for the computer and provides other desirable interfaces. The original PCB was specifically designed for the original Motorola PVT-6 (now called ONCORE BASIC) and I have had difficulty in supplying the PCBs for amateur users. I am in the process of doing a new **TAC** interface board to support all the receiver types and TAPR has agreed to make it available to the community (if/when I ever finish it!). In the meantime, my "aleph" FTP server has information on the existing design along with details of ways to make a functioning **TAC** or **TAC Lite** now.

FREQUENCY CONTROL: In addition to providing the timing needed for the QRP EME schemes proposed here, the GPS receiver can also be used to stabilize ("discipline") a crystal oscillator to provide the performance equivalent to a Rubidium (Rb) or Cesium (Cs) frequency standard. On time intervals from minutes to weeks, a GPS receiver can be used to measure time differences with exquisite precision. The ONCORE-based **TAC** achieves RMS levels ~30 nsec and the lower quality Garmin-based **TAC Lite** achieves ~300 nsec. Even if we are pessimistic about the performance (calling it 50/500 instead of 30/300 nsec) and multiplying by $\sqrt{2}$ (since both the starting and ending times are noisy), locking an external oscillator to GPS could achieve performance levels shown in the table.

	TAC	TAC Lite
RMS:	50 nsec	500 nsec
1 min	1.2E-09	1.2E-08
10 min	1.2E-10	1.2E-09
1 Hour	2.0E-11	2.0E-10
3 Hours	6.5E-12	6.5E-11
12 Hours	1.6E-12	1.6E-11
1 Day	8.2E-13	8.2E-12
1 Week	1.2E-13	1.2E-12
1 Month	2.9E-14	2.9E-13

The trick is to find the time when the stability of the oscillator matches that achievable with GPS – that will be the time constant τ that you use in the locking loop. For a fairly good crystal, this will probably be in the range of minutes up to an hour. If you are locking a low-cost Rb (like has appeared on the surplus market recently), several hours are probably OK.

Use a divider to count your crystal (or Rb) down to 1PPS. Use a time interval counter to measure the difference between the two 1PPS signals. Watch how it is drifting as averaged over the appropriate time interval. Then develop a correction voltage to tune the crystal, and VOILA! - You have a lab quality frequency reference!

My plans are to develop a couple of different locking circuits to do precisely this. The **TAC Lite** matches low-cost crystal oscillators, and it looks like a complete locking loop (including the dividers and counter) can be done with about 5 IC’s and a Parallax “BASIC STAMP 2”. A high performance lock loop suitable for a Rb or a high quality crystal matches the ONCORE-based **TAC** and is more complex. I’m eager to learn of others interested in this activity. Contact me by Email at

clark@tomcat.gsfc.nasa.gov
-or-
w3iwi@amsat.org

Appendix B:
EME Path Loss Calculations

Tom Clark, W3IWI

To develop this presentation, we needed to understand the path losses well. Several authors have done this in the past (see for example D.S.Evans and G.R.Jessop, *VHF-UHF Manual, Third Edition*, RSGB 1976, pg 9.12) but I thought it best to go back to first principles. The following table summarizes my calculations:

ISOTROPIC EME PATH LOSS CALCULATIONS

			Perigee	Mean	Apogee	
(A)	Earth-Moon Distance	km	356,000	384,400	407,000	
(B)	Flux Density at Moon	*w/m^2*	*6.3E-19*	*5.4E-19*	*4.8E-19*	
	for 1 Watt Xmtr	**dB(w/m^2)**	**-182.0**	**-182.7**	**-183.2**	
(C)	Radius of Moon	km	1738	1738	1738	Note:
(D)	Total power incident on moon,	*w*	*6.0E-06*	*5.1E-06*	*4.6E-06*	Evans &
		dBw	**-52.2**	**-52.9**	**-53.4**	Jessop
(E)	Reflection Albedo = 6.5%		*3.9E-07*	*3.3E-07*	*3.0E-07*	use **4pi**
		dBw	**-64.1**	**-64.8**	**-65.3**	in step (F)
(F)	Lambert scattering into hemisphere		*6.2E-08*	***5.3E-08***	*4.7E-08*	***2.6E-08***
	= **2pi** steradians	**dBw**	**-72.1**	**-72.8**	**-73.3**	**-75.8**
(G)	Flux Density at Earth	*w/m^2*	*4.9E-25*	*3.6E-25*	*2.8E-25*	*1.8E-25*
		dB(w/m^2)	**-243.1**	**-244.5**	**-245.5**	**-247.5**
(H)	(Wavelength^2) / (4pi) = Equivalent Isotropic Antenna Collecting Area:					
	F, MHz	Area, dB(m^2)	Total dB Path Losses			E&J
	50	**4.6**	**-238.6**	**-239.9**	**-240.9**	**-242.9**
	144	**-4.6**	**-247.7**	**-249.1**	**-250.1**	**-252.1**
	432	**-14.2**	**-257.3**	**-258.6**	**-259.6**	**-261.6**
	1296	**-23.7**	**-266.8**	**-268.2**	**-269.2**	**-271.2**
	2304	**-28.7**	**-271.8**	**-273.2**	**-274.2**	**-276.2**
	5760	**-36.7**	**-279.8**	**-281.1**	**-282.1**	**-284.1**
	10368	**-41.8**	**-284.9**	**-286.2**	**-287.2**	**-289.2**

In this table, *values in italics* are powers and flux densities in watts and watts/m^2 assuming 1 watt was transmitted from the Earth into an isotropic antenna. The **bold face values** are the corresponding values in **dBw** and **dB(w/m^2).**

Steps (A) and (B) compute the signals arriving at the moon at three different distances. Given the size of the moon, (D) computes the total power incident on the visible disk of the moon. The moon is an imperfect "mirror" and step (E) accounts for reflection losses using a canonical albedo of 6.5% (which is only a guess!). Down to this point, the Evans & Jessop calculations mirror mine perfectly.

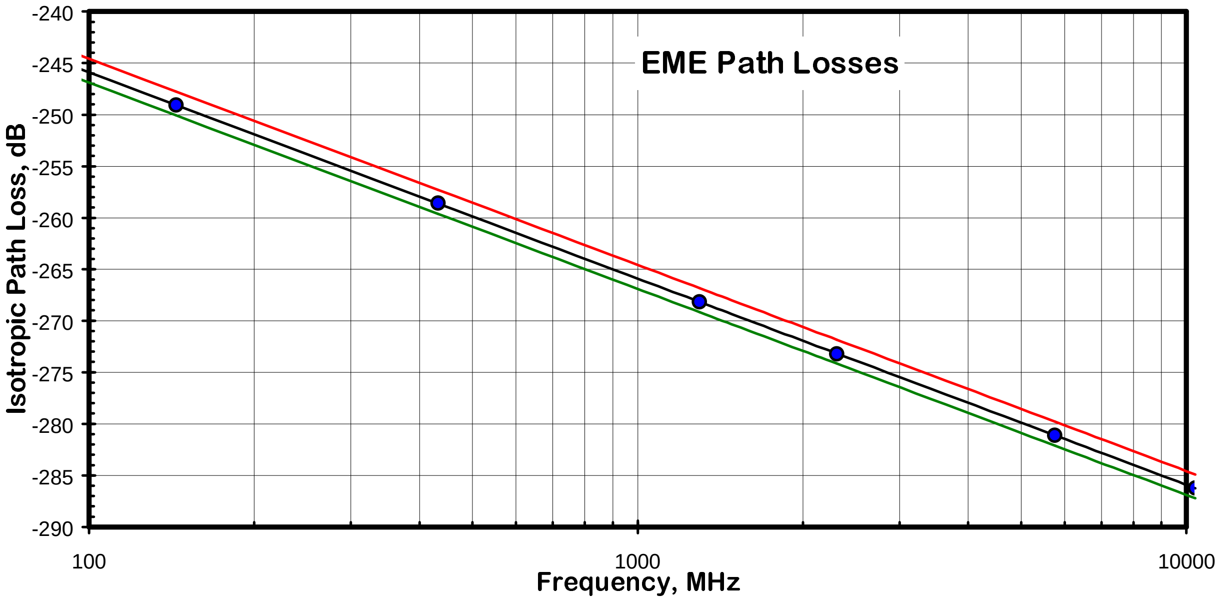
At step (F), we account for the fact that the moon scatters the signal in all directions. If the moon were a uniformly rough surface, this would be called Lambert scattering, and the energy would go into 2π steradians (half the sky). At this point, Evans & Jessop chose to regard the moon reflection as isotropic and use a factor of 4π! This immediately results in a 3dB discrepancy between our formulations!

In point of fact, the moon is not a uniformly rough Lambert scatterer and the reflections (especially from the central disk) favor the earth. For those interested in the physics, a Lambert surface would appear uniformly bright, but in point of fact the central region of the radar moon is MUCH brighter than the limb. All these subtleties may well be soaked up in the choice of an average albedo that makes the observations fit the theory (i.e. the albedo is really a fudge factor!). Anyway, from step (F) onwards, there are 4 columns, with the 4th being the 4π version of the mean distance column.

In step (G), we compute the flux incident on a 1 square meter area at the Earth.

The final step in the calculation is to figure out the equivalent isotropic gain of a 1 m^2 antenna. In step (I) we use the standard $G = 4\pi A/\lambda^2$ formula.

Graphically, the EME path losses from the table look like this:


EME Path Losses
Isotropic Path Loss, dB
Frequency, MHz