PSK21, EMEpsk, and SlowPSK Documentation

Klaus von der Heide, DJ5HG

PSK21, EMEpsk, and SlowPSK are advanced specialized digital modes for amateur radio communication. They were designed to reach extraordinary sensitivities below -30 dB (in 2500 Hz bandwidth) with usual transmission periods of 1 minute (PSK21 and EMEpsk) and more (SlowPSK). This paper is a documentation of the technical details. The only difference between PSK21 and the variants of SlowPSK is the stretching factor of 6, 18, 36, or 72 in time and the corresponding gain in sensitivity of $10*\log_{10}$ ([6 18 36 72]) = [7.8 12.6 15.6 18.6] dB. EMEpsk is the same as PSK21 with the exception of Doppler-elimination in EMEpsk (Chapter 9).

1. The Modulation

PSK21 uses Binary Phase Shift Keying (BPSK) at a rate of 8000/348 = 20.8333 bits/s. The digital-to-analog conversion of the bitsequence is done by sequential replacement of every bit by a continuous pulse-function. The pulse-function is sampled at 8000 samples per second. The pulse function used in PSK21 is a sinc function. It is shown in Figure 1. Figure 2 explains how the PSK21-signal is generated from a given bitsequence.

The baseband signal of figure 2b finally is multiplied with a sine-wave of the audio carrier frequency 1000 Hz. The result is the audiosignal that is shifted by the SSB-transceiver to the target frequency. It is a Double Side Band (DSB) signal, but it is called a BPSK signal because the phase is 0° where the binary values (the red dots in figure 2) are +1, and 180° where they are -1.

The spectrum of a PSK21-signal is rectangular, and it extends from 989.6 Hz to 1010.4 Hz with extremely sharp edges. Stations running PSK21 therefore can be spaced by 25 Hz without causing interference.

All is the same with SlowPSK with the exception of lower Baud rate and lower bandwidth.

2. The General Binary Packet Format

All PSK21-blocks contain 996 bits. 332 bits of them (1/3) are the address. A general address serves as a broadcast address for CQ, QRZ, and QST. Messages addressed to specific stations use the encoded callsigns of the addressed stations as the address. All address bits are interleaved with the 664 data bits: The address bits are at the positions 1, 4, 7, 10, 13, ...

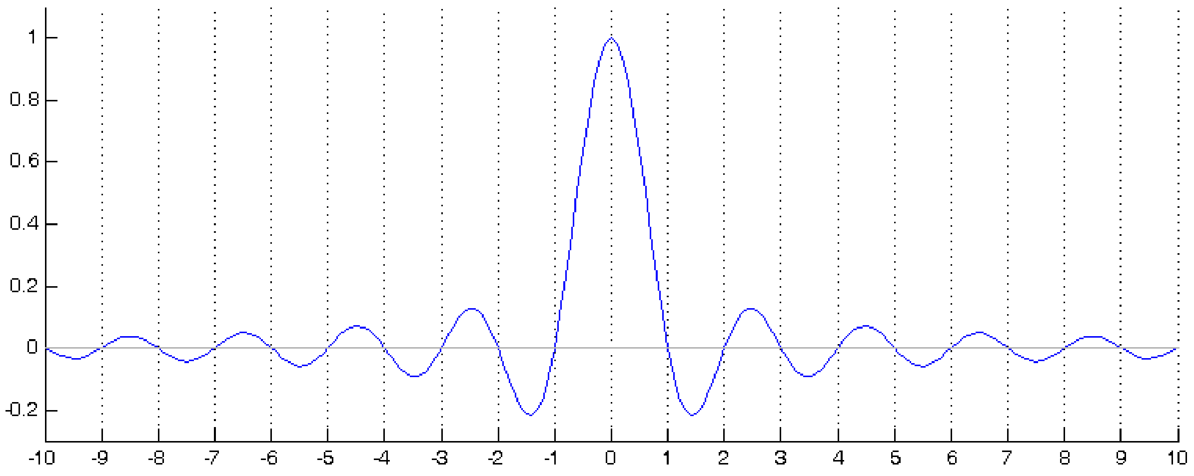
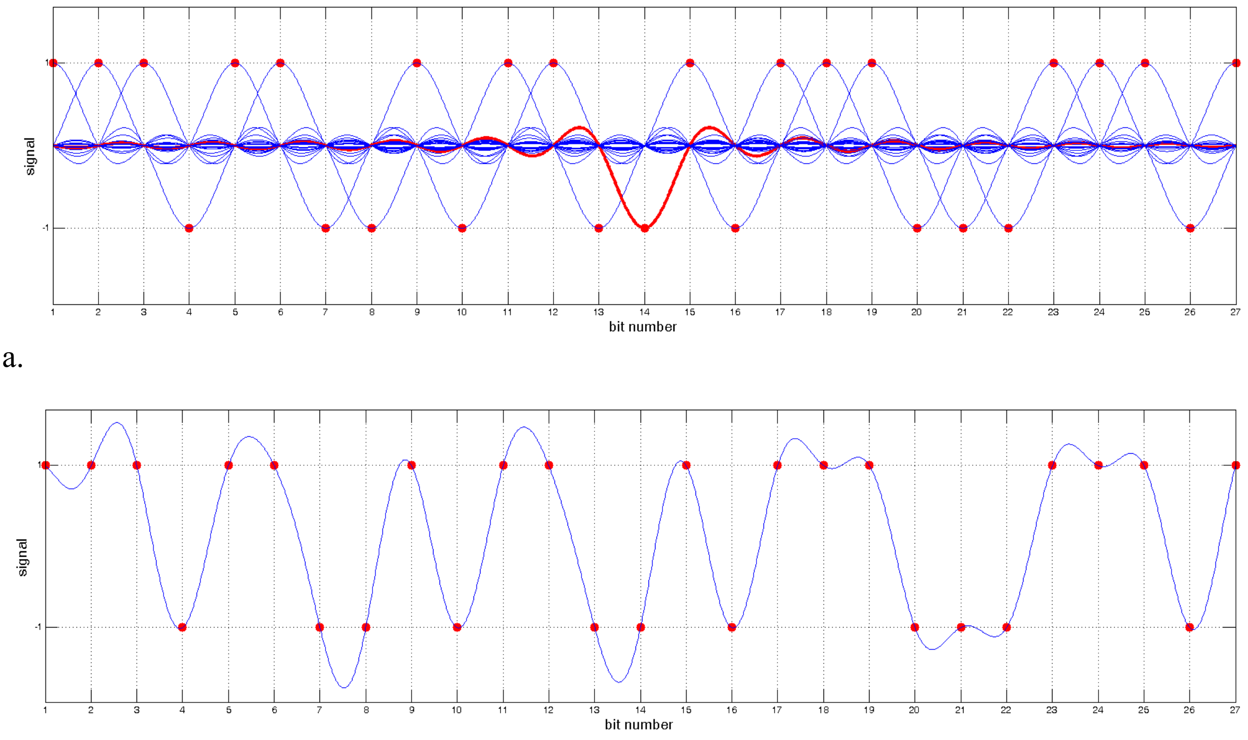


Figure 1. The sinc-pulse. It is infinitely long to both sides with slowly decreasing amplitude. It has zeros at k = ±1, ±2, ±3, ...



a.

b.

Figure 2. Generation of the PSK21 baseband signal. The example bits `[1 1 1 0 1 1 0 0 1 0 1 1 0 0 1 0 1 1 1 0 0 0 1 1 1 0 1]` are replaced by the values ±1. These values `v` are replaced by `v` times the sinc-function. All these functions are shown as blue lines in figure 2a except for one which is colored in red for demonstration. It is obvious that all other functions have a zero where one function has it's value `1` or `-1`. The sum of all sinc-functions in figure 2b therefore goes through all the values v which are marked as red dots.

The horizontal spacing of the bits (1.0 in the figure) is 48 ms because of the bitrate of 20.8333 bits/s. The PSK21-signal is an analog signal. But the PSK21 program generates it at the standard sampling rate 8000. The soundcard converts it into the desired analog signal.

3. The Binary Address Patterns

All stations know the general binary address pattern

```
0100101000100011010000000011110101010100111101000001011011001101011000100
0011000111000111011110101001000000110010000111110101101001011101100111000
0010000001011111000101011010010011001101110100111011111101101011111011100
1101010110100110101011100100001011001011010111011101110010101000010000011
0011110110111100101111100010110110110000
```

Additionally, each station knows it's unique own binary private address which is generated from it's callsign. The private address is a concatenation of the fixed pattern of 166 bits valid for all stations

```
1110000001000010011011110011011011100110010101111011010100010111001010100
0000101010010000110010010111011001110101010011011000100100101011010011111
00011011110100000010
```

and 166 individually differing bits. The fixed part enables all stations to synchronize and detect transmissions addressed to other stations by correlation with these 166 bits.

All stations correlate the received signal with the 332 bits of the general address and with the 332 bits of their own private address to detect blocks. The address pattern only serves for block detection and phase synchronization. A correct decode of the private address is not necessary because the addressed callsign is additionally encoded in the information data.

The encoding of the private address starts with a binary pattern of 54 bits generated from the callsign. This sourcecode of the callsigns is specified in Chapter 7. 17 checkbits are added for error detection yielding a pattern of 71 bits. The checkbits are the concatenation of the residual codes with `r = 3, 7, 31, 127`. The reason for using four residual codes instead of one is to increase the Hamming distance of the binary codes of similar callsigns. The 71 bits are encoded by a convolutional code of constraint length 13 and code rate 1/2. It' polynomials are:

```
 1 0 1 1 0 0 1 1 0 1 1 1 1   and   1 1 0 1 0 0 1 1 1 1 1 0 1
```

This results in two patterns of length (71 + 13 − 1) = 83. These 2 vectors are shuffled by the same index vector which defines the sequence of the bits:

```
  1  7 13 19 25 31 37 43 49 55 61 67 73 79  2  8 14 20 26 32 ...
 38 44 50 56 62 68 74 80  3  9 15 21 27 33 39 45 51 57 63 69 ...
 75 81  4 10 16 22 28 34 40 46 52 58 64 70 76 82  5 11 17 23 ...
 29 35 41 47 53 59 65 71 77 83  6 12 18 24 30 36 42 48 54 60 ...
 66 72 78
```

The shuffled two bit patterns are concatenated to a data bit vector of 166 bits.

5. The Binary Data Pattern

Blocks with a general address transmit 56 bits of information, and those sent to a private address contain 58 bits of information. In the first case, 15 check bits are added for error detection, in the latter case 13 check bits. In both cases this results in 71 bits to be transmitted. These 71 bits are encoded by a convolutional code of constraint length 13 and code rate 1/8. It's polynomials are:

```
1 0 0 1 0 1 1 1 1 1 0 1 1
1 0 0 1 1 1 0 1 1 0 0 1 1
1 0 1 0 1 0 1 0 1 1 0 0 1
1 0 1 1 0 0 1 1 0 1 1 1 1
1 1 0 1 0 0 1 1 1 1 1 0 1
1 1 0 1 1 1 0 1 0 1 1 1 1
1 1 1 1 0 1 0 0 0 1 0 0 1
1 1 1 1 1 0 0 1 1 0 1 0 1
```

The tail-ended encoding yields 8 bit vectors of length (71 + 13 − 1) = 83. All these 8 vectors are shuffled by the same index vector used for the private addresses in the previous Chapter 4. The shuffled eight bit patterns are concatenated to a data bit vector of 664 bits. These 664 bits are interleaved with the 332 address bits as described in Chapter 2.

6. Error Detection

6.1. Introduction to Error Detecting Codes

The error correcting capability of codes ist limited. If the signal is more corrupted than the code can correct then the decoded result is worse compared to that of an uncoded transmission. Such faulty messages can be recognized by an additional Error Detecting Code. This means that, at a first step, the k information bits are encoded by an Error-Detecting Code (also called the inner code) into $m>k$ bits. And these m bits are then, as the second coding step, encoded by an Error Correcting Code (also called the outer code) resulting in $n>m$ bits.

Error Detecting Codes usually are systematic codes. A systematic code is one that starts each codeword with the information bits unmodified. The additional bits are called parity bits or check bits. These bits are generated from the information bits by a fixed algorithm. The receiver first decodes the outer (error correcting) code. Then it simply generates the check bits from the decoded information bits and compares them to the decoded check bits. If there is any difference, the message is discarded.

The capability of error detection also is limited. If only one check bit is used then on average every second faulty message would be accepted by random. In the case of n check bits the probability of accepting a faulty message is not lower than 2^{-n}.

6.2. Residual Codes

The error detection in PSK21 generally is based on residual codes. This means that the information bit array is interpreted as a natural number z. The check bits are computed as the binary representation of the residual of the division of z by a fixed denominator r.

Examples with $r=31$.

binary arrays	0000010001	1000111101	1111010010
z	17	573	978
remainder	17	15	17
check bits	10001	01111	10001

If r is a prime number then the chance to get acceptable check bits in case of a received random bit sequence is $1/r$.

6.3. Error Detection in PSK21

For CQ, QRZ, QST messages 15 check bits are generated with $r=32749$. The source information is encoded into 56 bits (see Chapter 7.2).

All private messages contain 58 information bits. 13 check bits are added generated with $r=8191$.

The private addresses contain 55 bits. 16 check bits are added generated with $r=65521$.

7. The Source Codes

The Error Correcting Code plus the elimination of unwanted false decodes by the Error Detecting Code has the consequence that all the bit arrays discussed in the following Chapters 7.2 and 7.3 can be assumed as error-free, when they are processed in the receiver.

7.1. The Binary Code of Arbitrary Text or Callsigns of up to 10 Characters

The alphabet for text (and callsigns) is given by

```
 / A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9 . , - ?
 0 1 2 3 4 5 6 7 8 91011121314151617181920212213242526272829303132333435363738394041
```

Each character to be encoded is replaced by it's index into this alphabet which is below the characters. So an `A` has to be replaced by 1, a `Z` by 26, a `/` by 0 and a blank by 41. If a text is shorter than 10 characters blanks are appended.

For example the callsign `DJ5HG` (plus 5 blanks) results in the sequence of indices

```
4 10 32 8 7 41 41 41 41 41
```

This sequence is cut in two sequences of 5 indices each:

`4 10 32 8 7` and `41 41 41 41 41`

Both sequences are interpreted as numbers written in base-42 notation instead of the usual base-10 notation. To get the value of the numbers `z1` and `z2`, we add the last number of the sequence, the second-last times the base, the third-last times the square of the base, plus the fourth-last times the cube of the base and so on. In the example, we get

```
z1 =  7 +  8*42 + 32*42*42 + 10*42*42*42 +  4*42*42*42*42 =  13244455
z2 = 41 + 41*42 + 41*42*42 + 41*42*42*42 + 41*42*42*42*42 = 130691231
```

The binary representations of `z1 and z2` are:

`000110010100001100000100111` and `111110010100011000010011111`

The concatenation of the bitpatterns is the binary code of the given text (in this example 'DJ5HG'):

```
000110010100001100000100111111110010100011000010011111
```

7.2. General Messages

QST, CQ, and QRZ messages are all built the same way:

The first 54 bits encode a text message TEXT of 10 characters as described in the previous chapter.

Two bits are appended to these 54 bits to define the message type. The type codes are:

type code	message
`00`	QST : arbitrary TEXT
`01`	CQ de MyCall (TEXT = MyCall, length of MyCall is 10 characters)
`10`	QRZ de MyCall (TEXT = MyCall)

The binary code of `CQ de DJ5HG` therefore is the binary code of `DJ5HG` plus `01`:

```
00011001010000110000010011111111001010001100001001111101
```

7.3. Private Messages

There are several predefined formats for private messages:

7.3.1. Addressed call of a station with or without report

The information to transport is the callsign of the sending station plus a possible report. The callsign of the addressed station is encoded in the private address and additionally in the check bits of the information.

Example: `SM2CEW de DJ5HG W`

This is a message addressed to `SM2CEW`. The callsign of the caller `DJ5HG` is encoded in the same way as it is done in a CQ call. The message bits are preceeded by four message type bits. They encode the possible reports Weak, Medium, Large, and none (W, M, L, -):

no report	0000
W	0001
M	0010
L	0011

The binary information of the message therefore is

`0001000110010100001100000100111111110010100011000010011111`

The 13 check bits of the addressed message are generated by the algorithm of Chapter 6.2. But not the encoded information of 58 bits is used to determine the residual. It's concatenation with the binary code of the callsign of the receiving station is taken, in the example:

`0001000110010100001100000100111111110010100011000010011111`...

`011100101011001110011001111100011100111000100101111111`

The value in decimal notation is `356541683245198804824280844222224` (which is much more than a million times the number of nanoseconds since the bing bang). The remainder of the division of this huge number by `r=8191` is `4015`, and it's binary representation in 13 bits is

`0111110101111`.

The message to be encoded with the outer code is the concatenation of the binary information of the message with these 13 check bits:

`00010001100101000011000001001111111100101000110000100111110111110101111`

7.3.2. Addressed free message of up to 10 characters

The 10 arbitrary characters are converted into 54 bits in the same way as described in Chapter 7.1. The message type bits are `1110`. The check bits are generated as described in chapter 6 with `r=8191`.

Example: The addressed message `SM2CEW de DJ5HG :  TNX PETER`

`1110011110001100001110100011010001000001000010100100000001`

The check bits are generated as above from this bit array plus the callsign of the receiving station, but plus the callsign of the sending station too:

`1110011110001100001110100011010001000001000010100100000001`...

`011100101011001110011001111100011100111000100101111111`...

`000110010100001100000100111111110010100011000010011111`

The remainder of the division of this number by by `r=8191` is `2910` or in binary notation `0101101011110`. The message then is

`1110011110001100001110100011010001000001000010100100000001` `0101101011110`

7.3.3. Addressed contest message with report, roger, number, locator, power, antgain

The message type codes of these messages are:

W	1000
M	1001
L	1010
RW	1011
RM	1100
RL	1101

The bit array starts with the four message type bits which encode the report and roger. The additional contest information is encoded into the following 54 bits:

Let be

s	the report index into { W, M, L } starting at 0
n	the contest QSO-number (1 ≤ n ≤ 100000)
L	an array of 6 indices that defines the Maidenhead locator (starting indices are 0). The indexed alphabet of the first two indices is `ABCDEFGHIJKLMNOPQR` The indexed alphabet of the last two indices is `ABCDEFGHIJKLMNOPQRSTUVWX` The indexed alphabet of the two indices between is `0123456789`
p	an index which specifies the actual transmitter power (starting index is 0). The indexed list is `{ <0.1W, 0.1W, 0.2W, 0.5W, 1W, 2W, 5W, 10W, 20W, 50W, 100W, 200W, 500W, 1kW, 2kW, >2kW }`
g	an index which specifies the actual antenna gain (starting index is 0) The indexed list is `{ <-15dB, -15dB, -12dB, -9dB, -6dB, -3dB, 0dB, 3dB, 6dB, 9dB, 12dB, 15dB, 18dB, 21dB, 24dB, >24dB }`

The information bit array now is assembled:

`bits(1...20)`	binary representation of `n-1`
`bits(21...46)`	binary representation of the following number: `18662400*s + 777600*L(6) + 32400*L(5) + 3240*L(4) + 324*L(3) + 18*L(2) + L(1)`
`bits(47...50)`	binary representation of p
`bits(51...54)`	binary representation of g

Example: Le the actual data be no roger, report is W, contest number is 1234, the locator is

JO53IM, transmitting power is 250W, antenna gain is 17dB. The above variables are:

`n=1234,  L=[ 9, 14, 5, 3, 8, 12 ], p=11, g=12.`

With these values we finally get the bit array

`1010` `00000000010011010001` `00100100101000001111010001` `1011` `1100`

The 13 check bits are generated as above with `r=8191` from the concatenation of this bit pattern with those of both callsigns of the QSO.

7.3.4. The Short Message

In PSK21 a short message has the same length as all other messages. So it is not short, but it transports only 3 bits of information plus 13 check bits.

There are 5 different messages. They are listed below with their source code bitpatterns:

```
RW:       0111000111010000110100011001100011101001010010001101010111
RM:       0111001010111010001000111001011001010110010001011101101010
RL:       0111011110111111010011110010010010000001101010101101010000
RRR:      0111111001000101000100010111011100111100011100011101010100
TNX 73:   0111011011100110011000000101101011101101100110110000101001
```

13 check bits are added as in the other private messages including both callsigns.

Example: `SM2CEW de DJ5HG RRR` is encoded by the following bitarray:

`0111` `111001000101000100010111011100111100011100011101010100` `0110111110110`

These messages are expected by the receiving station when the QSO is in the corresponding state. In any case, the receiver first tries to decode all messages by the Viterbi decoder. If this fails, the receiver correlates the demodulated signal with the expected 996 bit long encoded bitpatterns. This enables detection of expected messages down to -34 dB in PSK21. If a good correlation occurs, the phase synchronization and demodulation is repeated using the complete known pattern of 996 bits. The new demodulated signal again is decoded by the Viterbi decoder. If this succeeds and the error detection does not find any bit-error, the message is accepted as „received“ and displayed. Otherwise the correlation is displayed in time and frequency domain with a text notice of the corresponding message, but it is never considered as a „received message“.

8. Message Timing

The transmitting periods are fixed by the minutes of the local computer time. The PTT is switched ON at 0.0 s. The TX-signal switches ON at 0.5 s. So there is a 0.5 s delay for safe relay operation. The end of the transmission within the last minute of the period depends on the chosen period:

period [minutes]	1	5	15	30
end of TX [s]	48.3	47.4	21.0	41.6

9. Doppler Elimination in EMEpsk

The carrier recovery procedure of PSK21 / EMEpsk does only work if the phase irregularities are of statistical nature. Any existent systematic frequency shift or frequency drift must be eliminated before. Therefore transceiver frequency control by GPS or equivalent is necessary. But also the Doppler shift caused by the relative motion between the reflecting area on the Moon and a given location on the Earth must be eliminated. The location of the reflecting area is not precisely known. But fortunately, the difference is small if Moon's center is taken instead. **Figure 3** explains the principle of the Doppler elimination of EMEpsk. The idea is, that transmitted frequencies are shifted such that they are received at the Moon just on the dial-frequency of the transceiver (plus the nominal audio carrier 1500 Hz). All frequencies received from the Moon also are shifted such that the dial-frequency of the receiver is identical to that of the virtual transmitter on the Moon (and to that of the sending station).

Advantages of this Doppler elimination are:

(1) A reply on your CQ-call will exactly be received on your calling frequency.

(2) The search-space for incoming signals in frequency is drastically reduced.

(3) The bandwidth of an EMEpsk-transmission is 21 Hz. Stations can be spaced by only 25 Hz without any interference to each other via the Moon.

EMEpsk is the same as PSK21 with the exception of the Doppler elimination. It should be noted that such a Doppler elimination should be done whenever the reflecting object moves on a known trajectory (ISS for example).

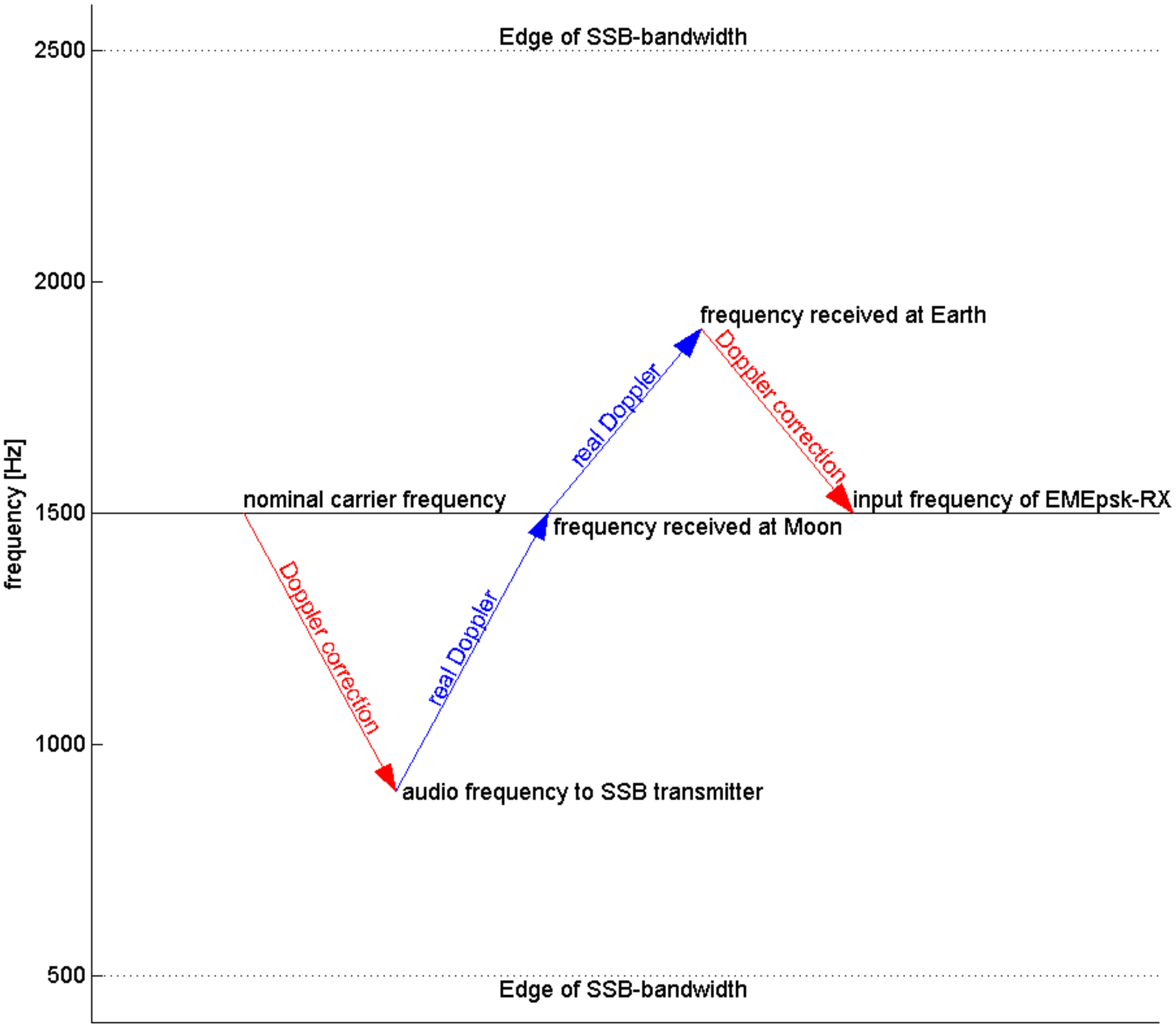


Figure 3. Doppler-elimination of EMEpsk. The relative motion between Moon's center and a given location on the Earth can precisely be predicted. Therefore, the Doppler shift of a carrier transmitted on Earth and heard on the Moon also can be predicted. EMEpsk shifts any transmitted carrier such that it is received at the Moon on the nominal carrier frequency 1500 Hz plus the dial frequency of the transceiver. This carrier is reflected by the Moon and received at a second place on Earth with a Doppler shift depending on that second location. EMEpsk again shifts the signal by this Doppler shift such that the EMEpsk receiver hears the carrier on just the nominal carrier frequency 1500 Hz. This Doppler-elimination works well as long as the audio frequency sent to the SSB-transmitter at one end and the frequency received on Earth at the other end both are within the SSB-bandwidth. On 144 MHz and on 432 MHz this is always true. But on 1296 MHz and higher, this requirement may not be satisfied at Moon-raise or Moon-set.