

Digital Modes II

PSK31 and JT65

Tom Worthington

Maui Amateur Radio Club

October 8, 2014

(PSK31 is from 8/8/2006)

If you want the original pp with notes, email
worth@maui.net

PSK31, the first modern digital mode

- PSK stands for Phase Shift Keying
- PSK31 has a 31 Baud (bits per second) data rate
- Developed in 1998 By Peter Martinez, G3PLX
- Very narrow bandwidth, therefore high power density
- Keyboard to Keyboard, one keystroke at a time to mimic a CW QSO (non break-in)
- Like CW, it has no error detection or error correction

Requirements

- SSB Transceiver
 - A big antenna is better, but many, world wide, get by with compromised antennas
- Computer with a sound card
- Cables or interface to connect
- Software – many free programs
 - Digipan
 - Mixw
 - Ham Radio Deluxe
 - Fldigi



Advantages

- Very efficient use of bandwidth
- Can easily have 20 – 30 useable channels in 3000 Hz pass band.
- The software can decode them all simultaneously
- Most use low power, 20 to 50 Watts
- $20\text{W}/31\text{Hz} == 2000\text{ W}/3000\text{Hz}$

How does it work?

- Uses a variable length character set
- Like CW, common characters are short, less common ones are longer
- Lower Case letters are shorter than caps
- 50 words per minute needs a bit rate of about 32 bits per second.
- 31.25 because $8000 \text{ Hz}/256 = 31.25$ was convenient for the early DSP implementations

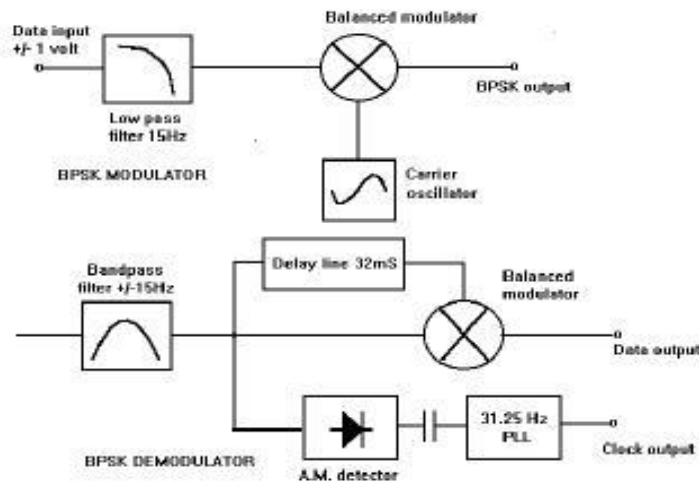
Phase Shift Keying

- A phase reversal is like switching the antenna leads (balanced feed line)
- Or multiplying the signal by -1
- Zero = phase reversal
- One = no phase reversal
- When Idling, send continuous phase reversals, this allows the receiver to sync his clock.
- So the transmitter is always on when in transmit mode.

Phase Shift Keying



PSK31 encode/decode



3. Block diagram of analogue BPSK modulator and demodulator.

For the decode to work we need to limit the number of 1's in a row so that the PLL doesn't drift.

The character set is a (1,9) run length limited code i.e. no more 9 bits without a phase reversal, but you can have two phase reversals in a row

Implementation

- All the hard work is done in the sound card
- You type, the software converts this to a digital stream that goes to the sound card
- Output of the sound card feeds the microphone input in SSB/USB (always) mode
- The sound card shapes the signal to keep the bandwidth narrow.
- Speaker output of the transceiver goes to the microphone input of sound card
- You select the signal and the sound card decodes the signal

14.070.000  Pre Att Nar NB Dig

SB 1 2 3 4 BPSK-31

Macros

- Manager
- Select Macro Set
- Preview On/Off
- Title Short/Full
- [0] CQ CQ CQ [Send immedi...
- [1] Callsign DE KH7CC [Sen...
- [2] Callsign de KH7CC [long]...
- [3] My Station
- [4] BTU
- [5] 73's and Bye-Bye
- [6] QTH
- [7] Intro
- [8] QRZ [Send immediately]

03:37:19 UTC

Logbook

Iags Options

Add Reset

Start End

03:34 03:37

Start = now End = now

Field (Tag)	Value
Callsign	NQ6U
Name	OM
QTH	
Locator	
Frequency	14.070.000
Band	
Mode	PSK31
Sent	599
Rcvd	599
Remark	

Output Level: 90

85

Thank you for the FB report Moe elo. Very nice signal from your station also.
Station here:
XCVR: FT-1999MP, 3#W
Antennas: 3 e-meeg 4#m invert*vee
Compute
K5RKS K5RKS de AC6YY AC6YY ok roger, I am relatively close to HAM RADIO OUTLET, I often go over there specially on Saturdays, a lot of hams go over there sifwe have a small hamfest. hi hiQX CQ
CQ CQ de NQ6U NQ6U NQ6U
CQ CQ CQ de NQ6U NQ6U NQ6U
CQ CQ CQ de NQ6U NQ6U NQ6U pse k
aof hams go over there sifwe have a small hamfest. hi hin. I am a member of SCCRa saneCQ de NQ6U NQ6U NQ6U
a
TX> 03:33:20 UTC
NQ6U NQ6U de KH7CC KH7CC K
RX> 03:33:29 UTC
kh7cc de nq6u thanks for call ur 599 here in simi valley, ca namers erik... back to you...et
TX> 03:34:48 UTC

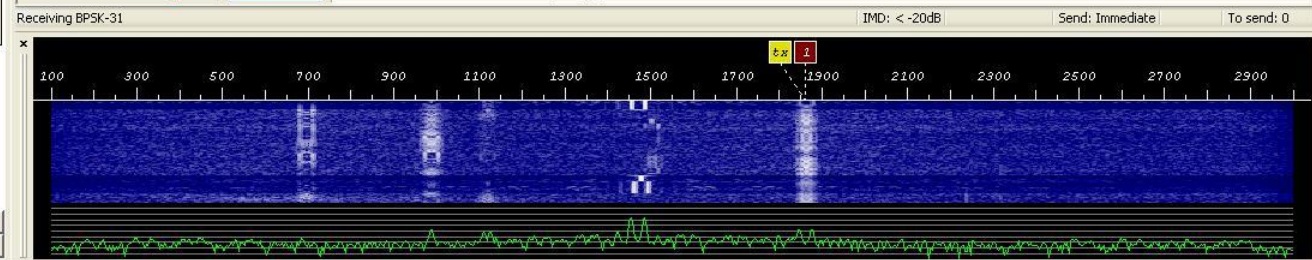
NQ6U DE KH7CC ok you h8br s have a nice signal here
My name is Tom Tom and the QTH is Maui Maui Hawaii Hawaii
(Loc: BL10TS BL10TS) (IOTA: OC-019)
I always have a hard time with the rst but I guess 559 559

there is a stronger stationdown the waterfall a bit tat is bout s7 s7
BTU NQ6U de KH7CC K
RX> 03:36:12 UTC
what program Ye you using...I am using ham radio deluxe, but I haven't figued out how to send in immediate mode - while I am typing...it only seems to work in batch mode. What pgm are you using there tom?

NQ6U NQ6U de KH7CC KH7CC K [stop]
NQ6U DE KH7CC ok you h8br s have a nice signal here
My name is Tom Tom and the QTH is Maui Maui Hawaii Hawaii
(Loc: BL10TS BL10TS) (IOTA: OC-019)
I always have a hard time with the rst but I guess 559 559

there is a stronger stationdown the waterfall a bit tat is bout s7 s7
BTU NQ6U de KH7CC K [stop]

RX RX Erase
RX Scroll
Rewind
Send (F1)
Auto (F2)
Stop (F4)
Unsend
TX TX Erase
TX Lock



- How can we calculate the bandwidth of a phase shift modulated signal?
- How can we shape the signal to minimize the bandwidth?

Jean Baptiste Fourier (1768-1830)

had crazy idea (1807):

*Any periodic function
can be rewritten as a
weighted sum of sines
and cosines of different
frequencies.*

Don't believe it?

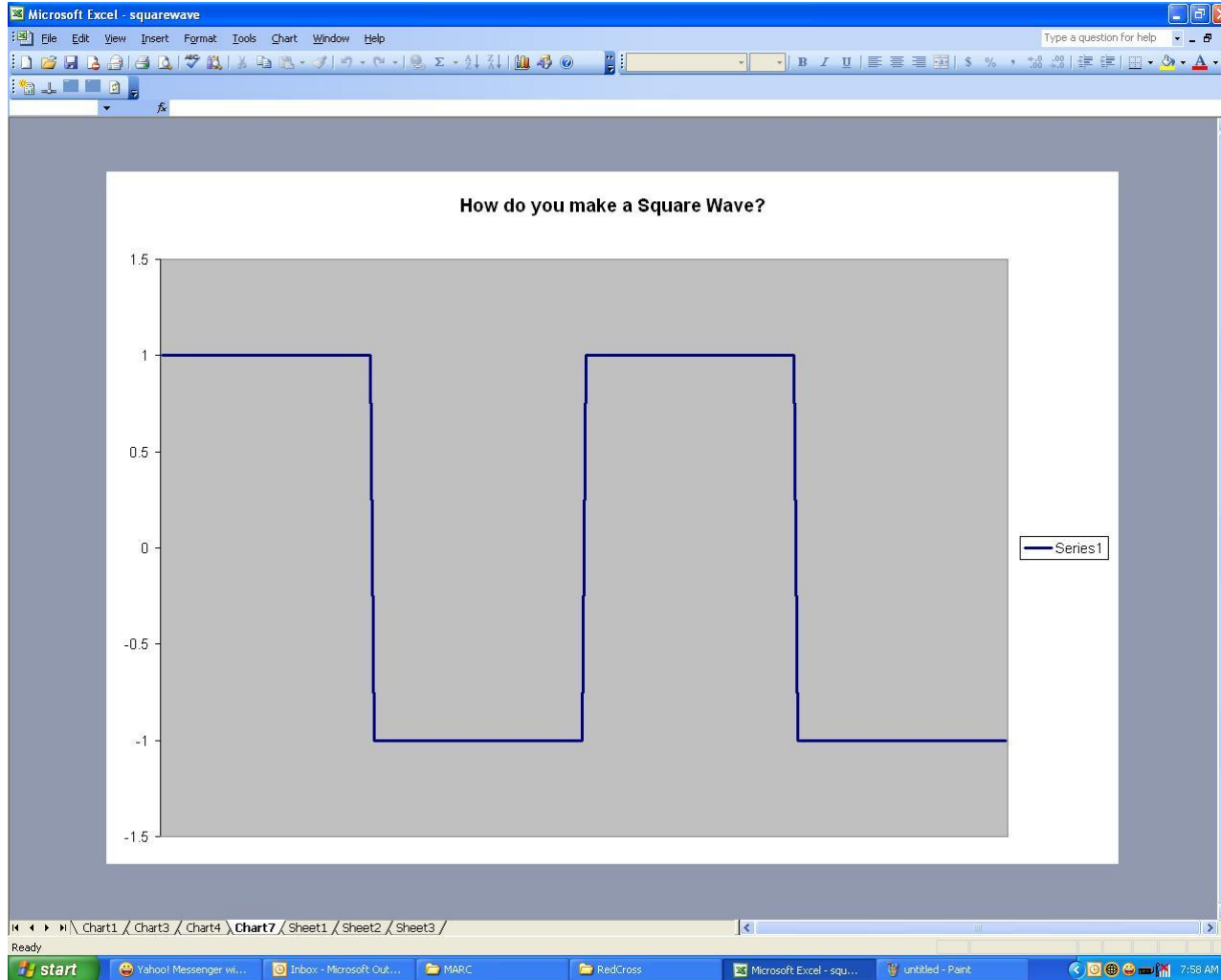
- Neither did Lagrange, Laplace, Poisson and other big wigs
- Not translated into English until 1878!

But it's true!

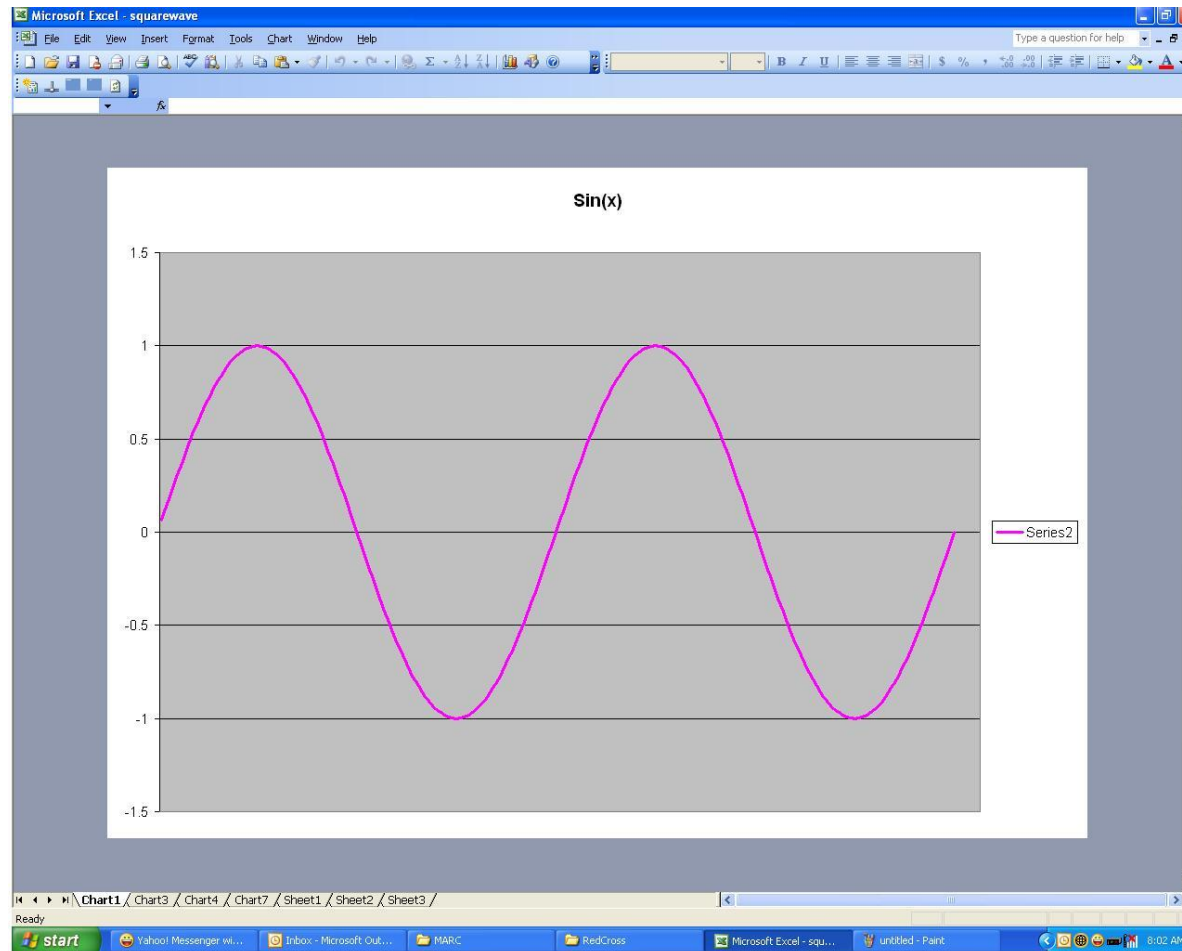
- called Fourier Series



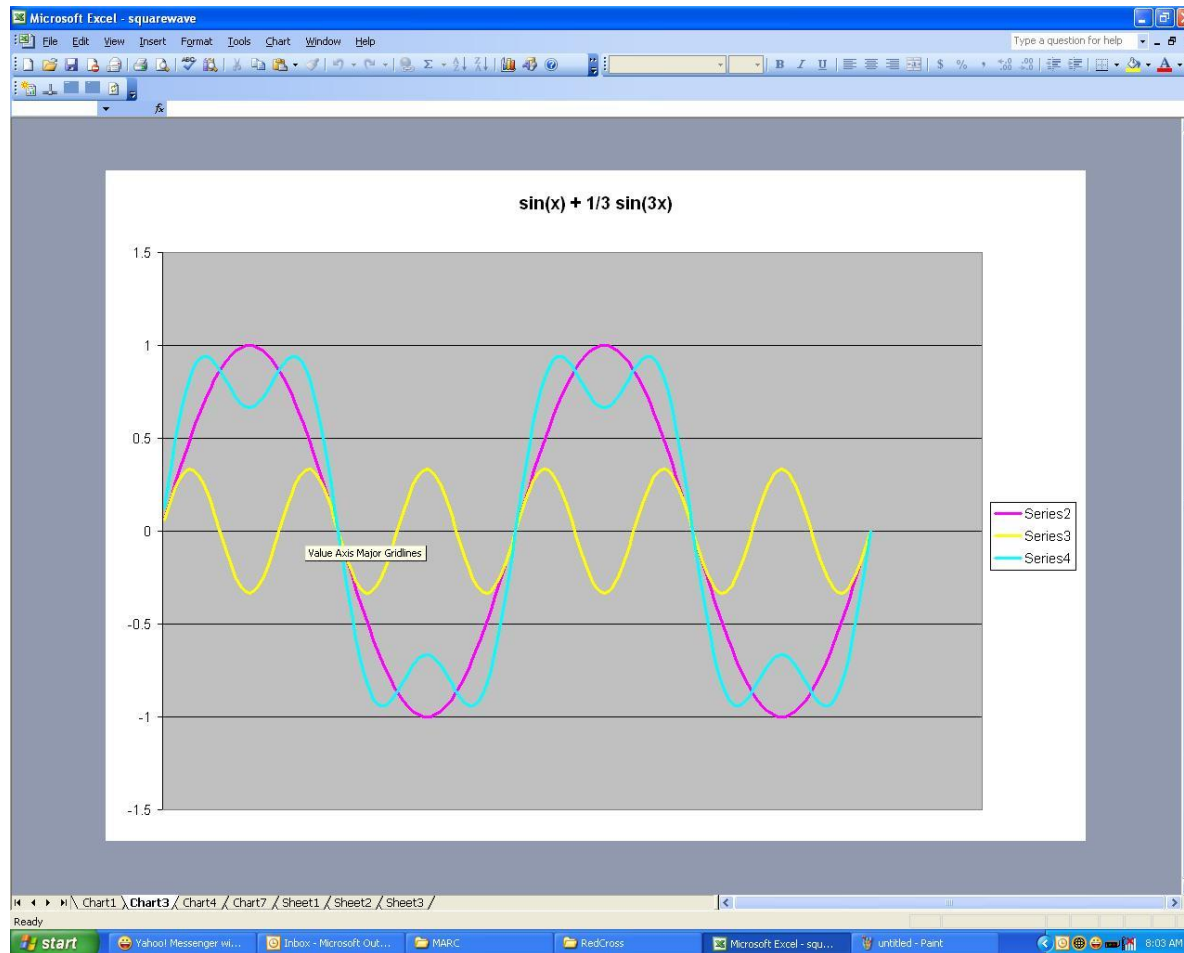
Making a square wave



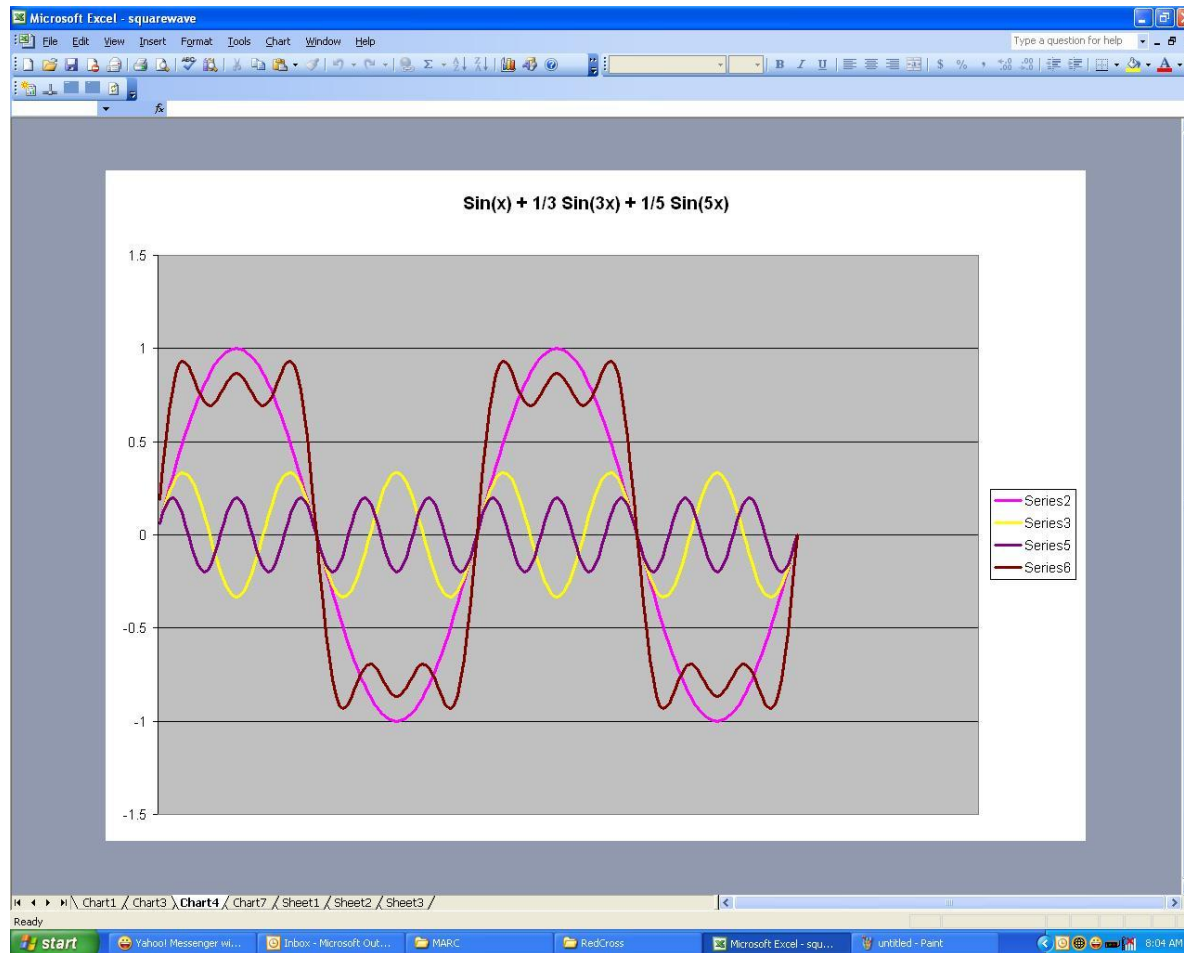
Start with sine wave at the fundamental frequency



Add some third harmonic



And a little 5th harmonic



Phase Shift Modulation

- How can we use what we have learned to calculate the bandwidth of a phase shift modulated signal?
- If we wanted to phase shift modulate a carrier at 31.25 Hz (PSK31 idle signal) we could just multiply the carrier by a square wave at 15.62 Hz
- But we know that a square wave is the sum of a series of sine waves and we know how to multiply sine waves

Modulation is Multiplication

Suppose I have a radio frequency signal: $\sin(2\pi ft)$

Suppose I multiply this signal by another signal: $\sin(2\pi f_m t)$

Remember back to taking Trigonometry in High School.....

$$\sin x \cdot \sin y = \frac{1}{2} [\cos(x - y) - \cos(x + y)]$$

We get two terms at the sum and difference frequency

$$\cos[2\pi(f - f_m)t] \quad \cos[2\pi(f + f_m)t]$$

Lower Sideband

Upper Sideband

Phase shift modulation

$$= \sum_{n=1,3,5,\dots} \frac{1}{n} * \sin(2\pi n * 15.62 * t)$$

The carrier looks like: $\sin(2\pi f t)$

That will give us a series of terms that look like:

$$\frac{1}{2} \frac{1}{n} \cos(2\pi(f + n15.6)t) \text{ and } \frac{1}{2} \frac{1}{n} \cos(2\pi(f - n15.6)t)$$

So there will be lots of harmonics, but we can suppress harmonics if instead we multiply by just the first term.

That has the effect of bringing the amplitude to zero before reversing the phase

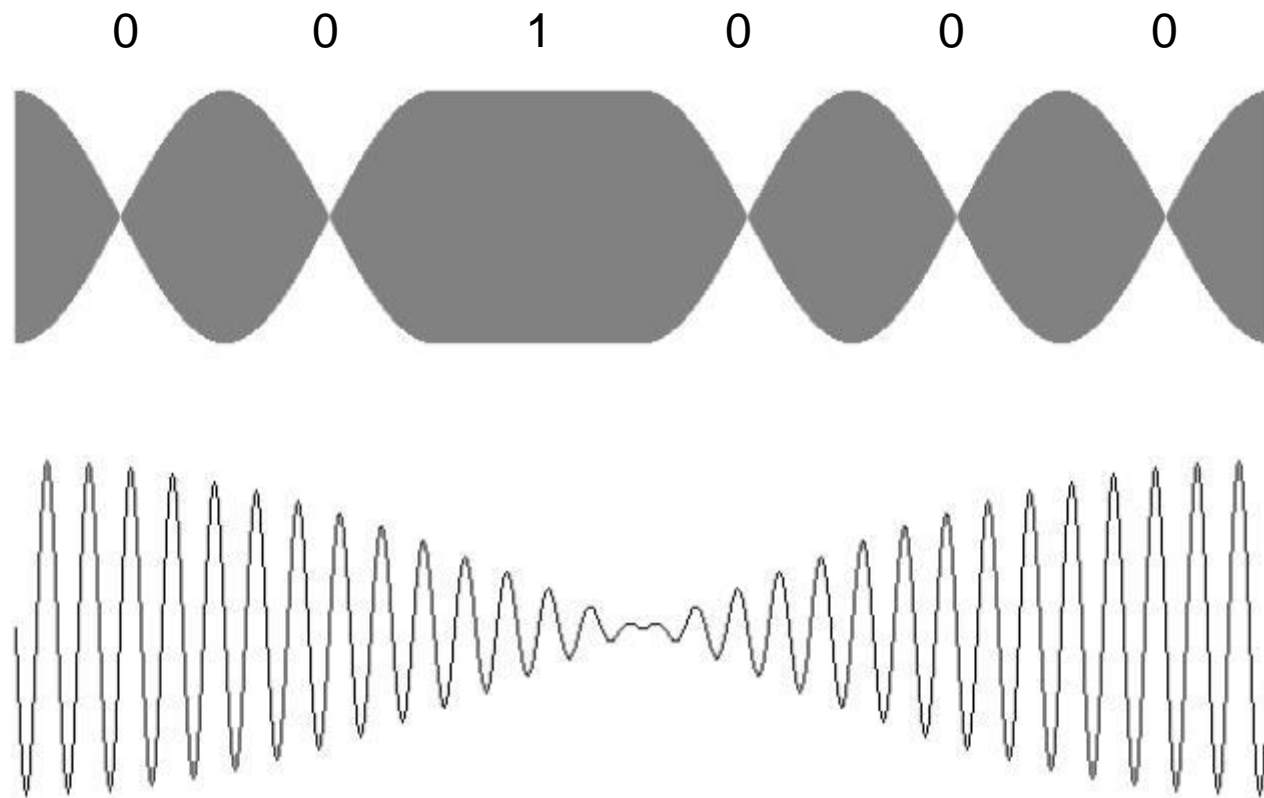


Fig 2. Showing the envelop of a BPSK carrier sending the Varicode "space" symbol, with a close-up of the phase-reversal at the cross-over.

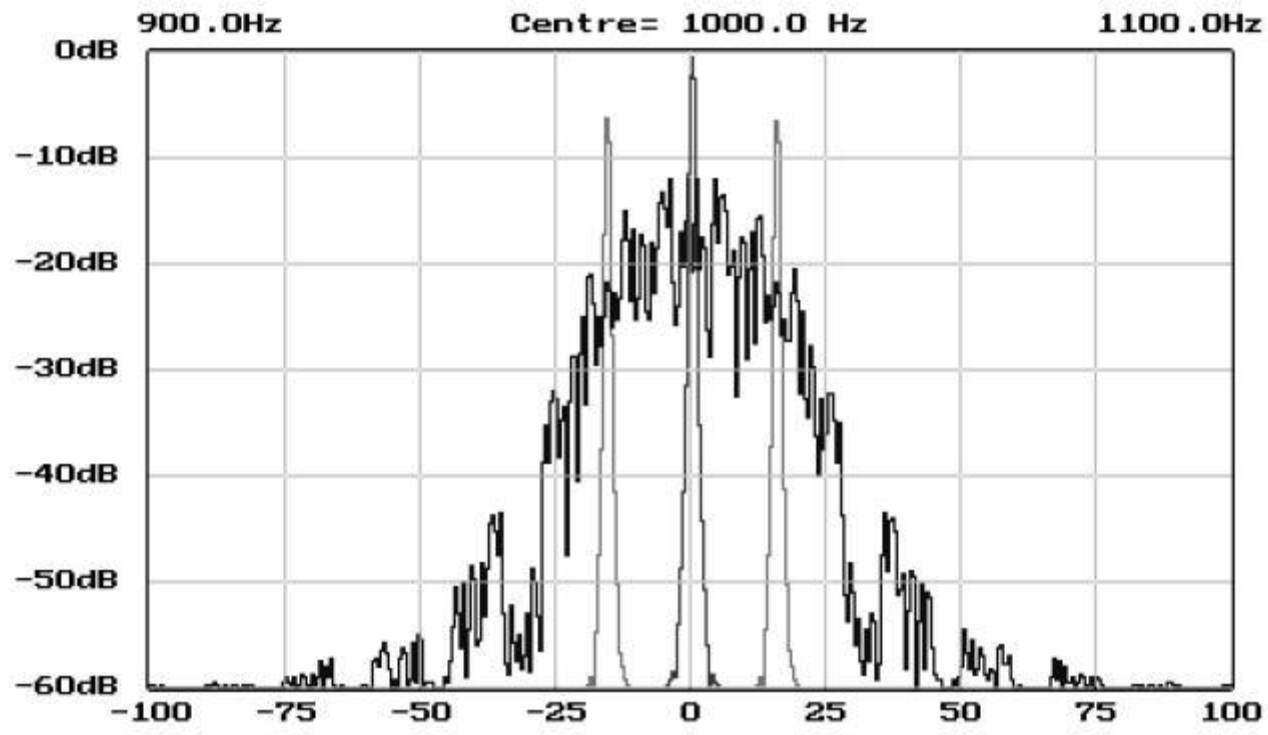


Fig 4. Spectrum of PSK31 sending carrier (red), reversals (green) and random data (blue)

PSK31 Phase Reversal

- If we didn't shape, i.e. reduce the amplitude to zero first, we would generate lots of harmonics
- If your transmitter is linear, your signal will be down 20 dB at ± 25 Hz
- Run low power, well below the point where you get an ALC indication, typically 30-50 Watts max
- If your signal is too wide people will let you know

Summary

- PSK31 (BPSK31, PSK63, QSK31....) is a simple, low power capable, narrow, cheap, popular HF mode

What about errors?

- In SSB and CW, and PSK31 we deal with errors by repeating
- CQ CQ CQ de NH6Y NH6Y NH6Y K
- AGN? AGN?
- Can we do better?

Error Detection and Error Correction

- Parity – add one bit that is the sum, base 2 of the bits in the word.
 - Detects all single bit errors
 - No correction
 - Can't detect all multiple bit errors
- Add more bits to detect multiple bit errors
 - Hash codes or checksums
- Ask for a retransmission, AGN? AGN?

History (from Wikipedia)

The most famous early systematic use of error detection was by Jewish scribes in the precise copying of the Torah. An emphasis on minute details of words and spellings evolved into the idea of a perfect text in 135 CE, and with it increasingly forceful strictures that a deviation in even a single letter would make a Torah scroll invalid.^{[1][2]} The scribes used methods such as summing the number of words per line and per page ([Numerical Masorah](#)), and checking the middle paragraph, word and letter against the original. The page was thrown out if a single mistake was found, and three mistakes on a single page would result in the entire manuscript being destroyed (the equivalent of retransmission on a telecommunications channel). The effectiveness of their methods was verified by the accuracy of copying through the centuries demonstrated by discovery of the [Dead Sea Scrolls](#) in 1947–1956, dating from c.150 BCE-75 CE.^[3]

What about error correction?

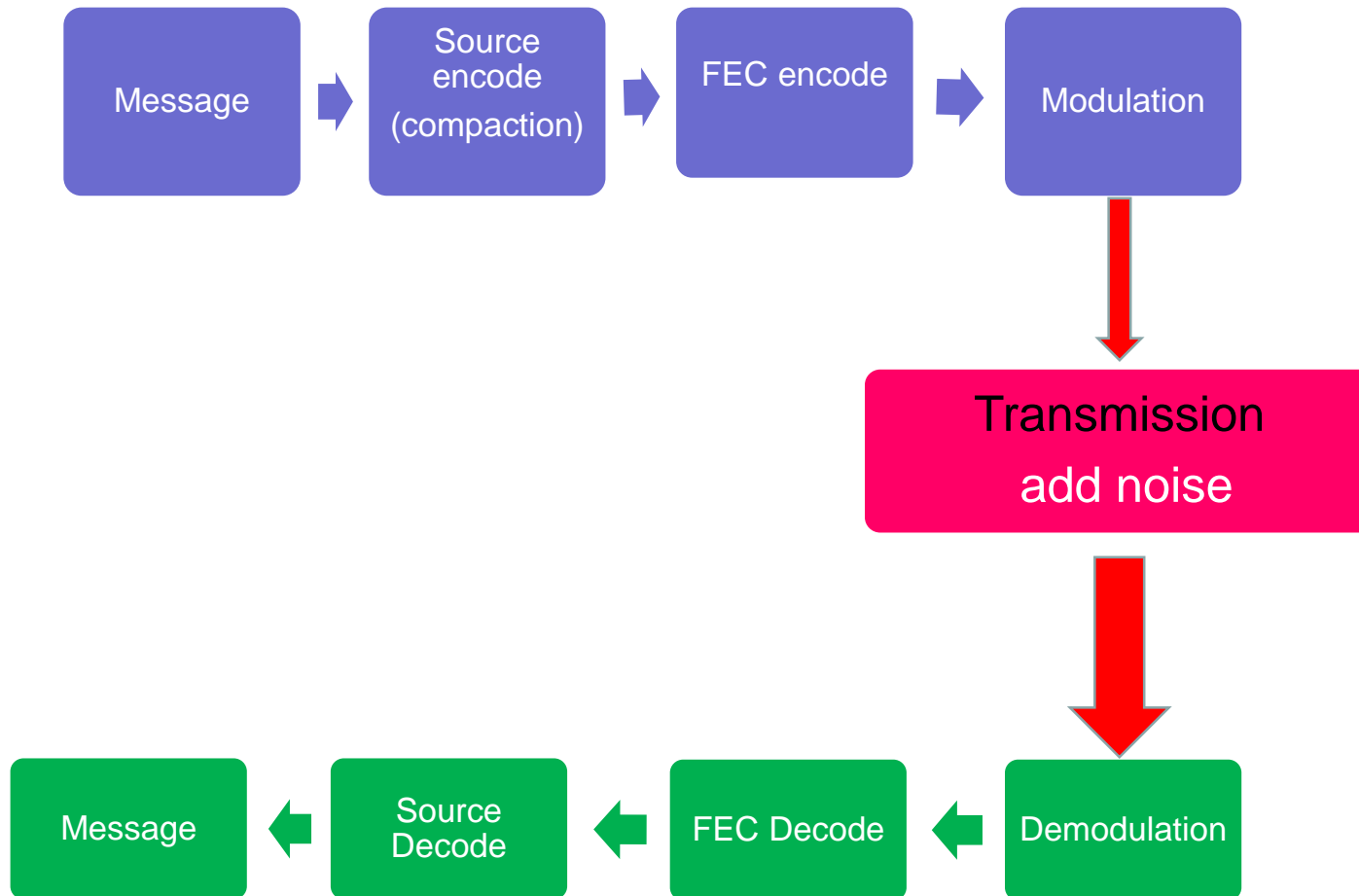
- This the field of Information and Coding Theory
- Many techniques have been developed to maximize the data transmission speed over a noisy channel that guarantee an acceptable error rate
- A typical error rate might be 1 in 10^{12} for a hard disk
- An audio CD can correct 4000 missing bits in a row (scratch)
- The best technique depends on the kind of defects or noise in the transmission channel

What about Radio Transmission?

- Interference
 - Other people talk on top of you
 - Nearby signals confuse the receiver
- QSB (fading)
 - Signal gets weak for a while
- Static
 - Missing signal for a while
- Multipath distortion
 - Same signal, but delayed in time

Amateur Modes that use Error Correction

- MT63
- Olivia.
- DominoEX-FEC
- Throb
- JT65



FEC = Forward Error Correction, no reverse channel to ask for repeats so error correction bits are added to the data stream

JT65

- Developed in 2003 by Joe Taylor, K1JT
- Follow on to WSJT (2001) developed specifically for EME – Moon bounce
- Slow
- Very structured
- Very good – opened up EME for small stations
 - can decode signals with a s/n of -28 dB!
- Extended to HF

Structured Messages

- CQ NH6Y BL10
 - Anybody want to talk to Maui?
- NH6Y W1AW FN31
 - Please talk to me, I'm in Newington CT
- W1AW NH6Y R-10
 - W1AW, I received your location and call sign correctly, and your signal strength is -10 dB
- NH6Y W1AW R-22
 - NH6Y, I correctly received the signal strength you sent, your signal strength is -22 dB
- W1AW NH6Y RRR
 - OK I got that
- NH6Y W1AW 73
 - Thanks for the nice qso

Each message takes one minute to transmit

- The previous QSO took 6 minutes to complete!
- Transmit for 47 seconds, starting 1 second after the start of the minute.
- Need to synchronize your clock to UTC

Use knowledge of content to reduce the message length

- JT65 Compresses messages based on knowledge of content
 - Call Signs = prefix (1 or 2 characters), then a digit, then a 1 to 3 letter suffix
 - $= 37 \times 36 \times 10 \times 26 \times 27 \times 27 =$ about 262 million possible call signs
 - $2^{28} =$ 268 million
 - Grid Square = 180×180 requires 15 bits
- A message has two call signs plus
 - Grid square, signal report, CQ, 73, R...
 - 72 bits = $28 + 28 + 15 + 1$
- Special Event and Portable call can be a problem
 - W1AW\KH6 or SI2014ECC for example
 - A 13 character arbitrary message can also be sent.

Error Detection/Correction

- Take the 72 bit data and add 306 error correction bits
- $378/72 = 5.25$, so it is sort of like sending the message 5 times
- Actually much better than that.
- The message is spread out over the 47 seconds so that if you can copy only a few seconds you can correctly decode the entire message
- If the message decodes, it is correct

JT65

- Uses Reed-Solomon Polynomials to spread the data into the entire transmission interval and add error correction.
 - Developed in 1960 at MIT but they didn't know how to invert efficiently
 - In 1969, Berlekamp and Massey developed a decoding algorithm
 - Used in the Voyager program to transmit images
 - Used in CDs and Disk Drives
- The RS Polynomial takes the 72 data bits and generates a 378 bit message that includes the error correction data

JT65 Modulation

- 126 time intervals each 0.372 sec long
 - Sound card samples 4096 points at 11025 samples/sec in each time slot
- In each time interval the waveform is a constant amplitude sinusoid at one of 65 pre-defined frequencies
 - Half the slots are used for time synchronization
 - The 378 bit word is sent as 63 6 bit symbols
 - The sound card matches the phase whenever the frequency changes to keep the bandwidth small

JT65 Modulation - 2

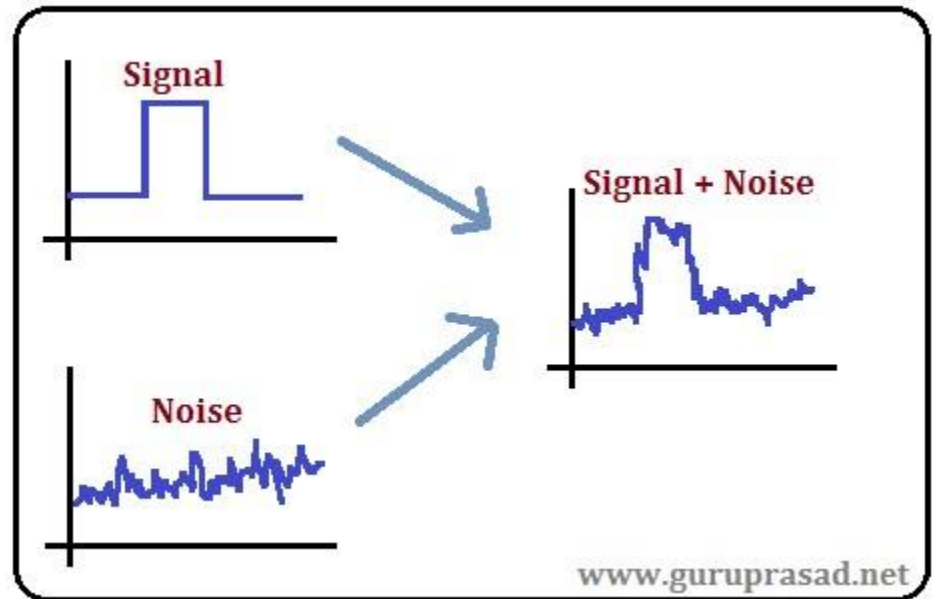
- Sync tone is 1270.5 Hz
 - Sent when sync value is 1
 - Absent when sync value is -1
 - Other tones are:
 $1270.5 + 2.6917(n + 2)$ where $0 \leq n \leq 63$
- In order to decode correctly we need to accurately synchronize the clocks

The Synchronizing Vector

- Every other time interval is used to send the synchronizing vector.
- A tone of 1270.5 Hz = 1, no tone for a -1
- The vector used has a very sharp autocorrelation function that makes it possible to synchronize the receiver to the transmitter, i.e. Allows the receiver to know which data bit is which and to separate the 64 different data tones

Suppose I want to detect the time that a signal arrives.

Noise can introduce errors, but I can't reduce the noise, how can I do better?



What if I send a sequence of binary signals, either a 1 or a -1 and record the data. Each separate signal is noisy and therefore the arrival time is uncertain but **If I use the fact that I know what was sent, I can improve my knowledge of the arrival time.**

Slide the received data back and forth, trying all possible arrival times and compare it with what I know was sent. This is called correlation.

If I choose the sequence well, there will be a distinct peak when they line up.

Consider a sequence +1, +1, -1
Multiply then add, i.e. integrate

+1 | +1 | -1

Sum = 0

+1 | +1 | -1

+1 | +1 | -1

Sum = -1

+1 | +1 | -1

+1 | +1 | -1

Sum = 0

+1 | +1 | -1

+1 | +1 | -1

Sum = 3

+1 | +1 | -1

+1 | +1 | -1

Sum = 0

+1 | +1 | -1

+1 | +1 | -1

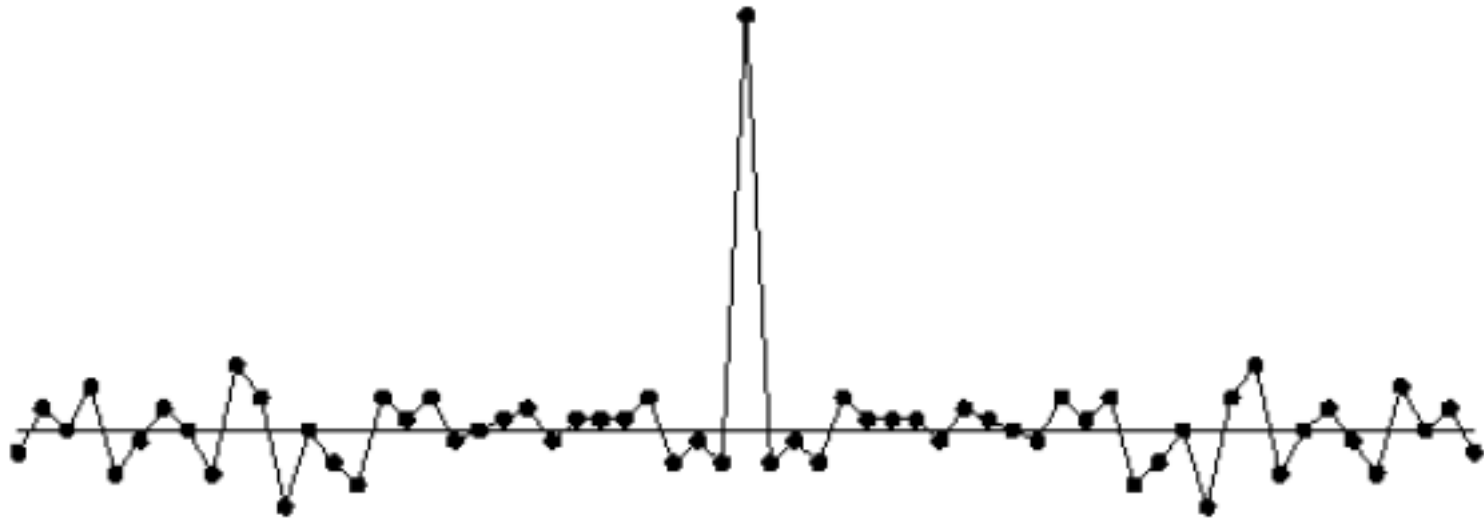
Sum = -1

+1 | +1 | -1

The Autocorrelation Function is: ...0,0,0,-1,0,3,0,-1,0,0,0.....

This is a Barker Sequence, one of only 4!

Autocorrelation of the JT65 synchronizing vector



```
100110001111110101000101100100011100111101101111000110101011  
001101010100100000011000000011010010101010011001001000011  
111111
```

This sequence is called a *pseudo random vector*, but there is nothing random about it. It is carefully selected to have the sharpest autocorrelation.

JT65

- Once we synchronize the incoming data...
- Find the time slots corresponding to the data
- Calculate the 6 bit value in each time slot
- String them together to find the 368 bit message.
- Use the Berlekamp/Massey algorithm to correct errors and generate the 72 bit received message
 - Or say we couldn't decode the message
- Uncompact the 72 bit message to make it person readable

What if the decode fails?

- If we had a very powerful computer we could try each of the possible 2^{72} messages and see which one gives the best fit to the received data
 - That is not (yet) possible
 - $4.7(10)^{21}$ messages
 - So keep a list of common call signs, for EME that is relatively small
 - Try messages with your call and those stored calls and look for good matches
- This is the Deep-Search Decoder
 - Provides about 4 dB more sensitivity

JT65 Software

- WSJT-X v1.4 is the current version from Joe Taylor.
 - Supports a new mode, JT9 that has smaller bandwidth
 - Includes Rig interface and logging
 - Supports: Windows, Linux and OS X
 - Source code available
- JT65-HF is available from SourceForge

RS Coding 0.1

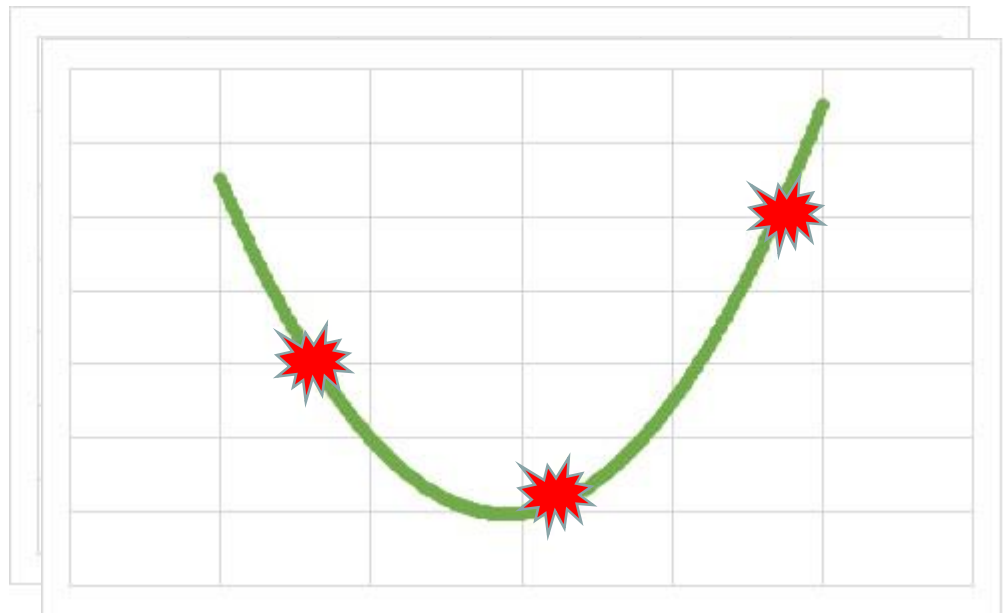
Suppose I want to send 3 numbers a, b, c

Use these three numbers to create a quadratic equation $y = a + bx + cx^2$

We know that three points define a parabola and that given any three points we can calculate the equation of the parabola that goes thru these points

Instead of sending a, b, c
Send three points that lie on
The curve

So what does that add???

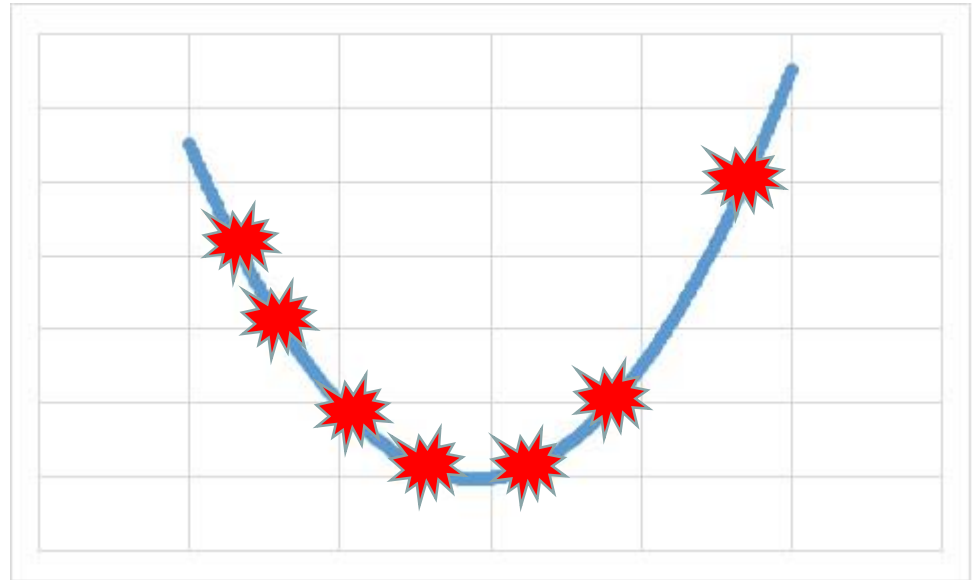


RS Coding 0.2

I can send some extra points

If I send $2s$ extra points

I can correct s errors



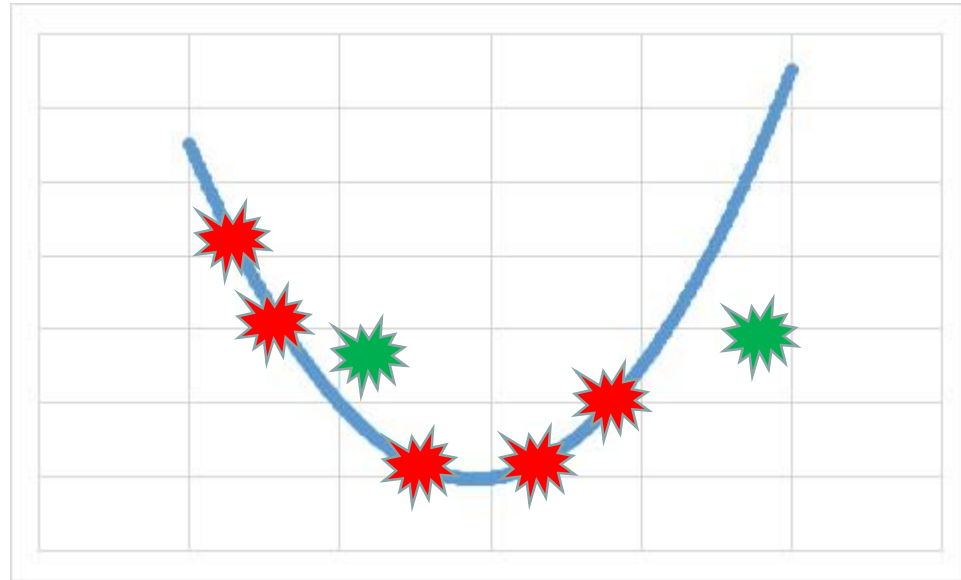
How??

RS Coding 0.3

$$n = k + 2s$$

Need to send k numbers,
Add $2s$ extra points

In our case $k = 3$, $s = 2$
And $n = 7$



Look for a subset of the data with at least $k + s$ points that all lie on the same Polynomial, in our case a parabola

Since there are at most s errors, and since k points are all that are required to form a unique polynomial, we can solve for the coefficients of the polynomial and know, that since at least k of them are not in error, the polynomial is identical to the original that was sent. Therefore, can calculate the original message

RS Coding 0.4

- Of course it's just a bit more complicated.....
- To send r symbols, map these symbols to the coefficients of a polynomial of order $r-1$
- Can't use real numbers, and the set of coefficients must be closed under $+$ \times
 - (if not you can't do the arithmetic exactly)
 - This is a *Galois Field*
- JT65 use the RS(63,12) code to encode the 72 bit message into 63 six bit channel symbols

Digital Modes Conclusions

- Increasingly popular
 - Allows qrp operation
 - Allows DX with antenna restrictions
 - Seeing more Dxpeditions using PSK/JT65
- Unfortunately many stations use macros, i.e. canned messages
 - That's fine for Contest or DX exchanges but I don't want a 500 word resume.
- Automatic Spotting Networks help in finding DX

How can I spread out the data in time?

- Instead of sending each bit sequentially, transmit all the bits at the same time using set of signals that are long in time but distinguishable.
- To do that we need a set of symbols, i.e. functions, that don't interfere with each other.
- These are called orthogonal functions

$$\int f_n(t)f_m(t)dt = 0 \text{ unless } n = m$$

Sine or Cosine functions will work

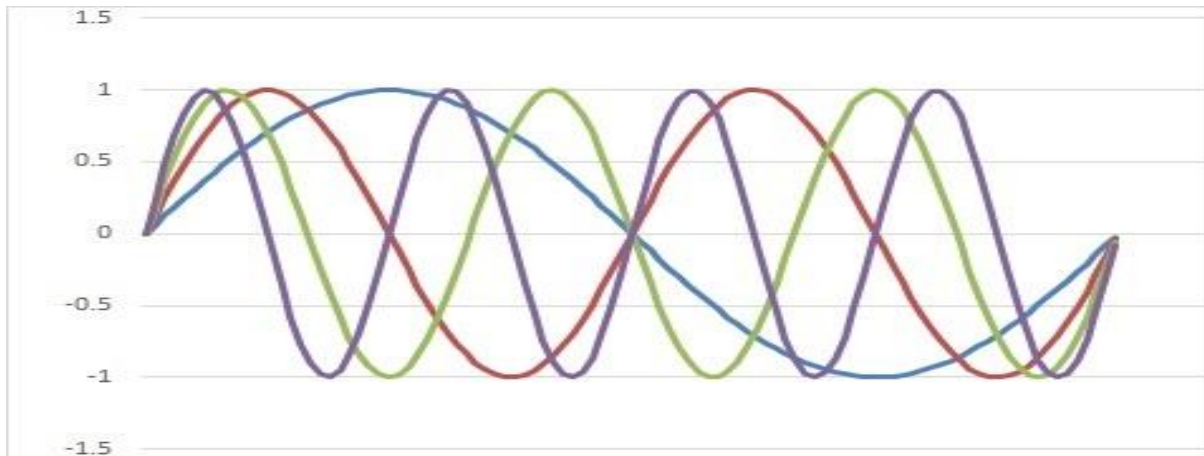
If you took calculus, you learned: (right at the end of the second semester)

$$\int_0^1 \sin(2\pi nt) \cdot \sin(2\pi mt) dt = 0 \text{ unless } n = m$$

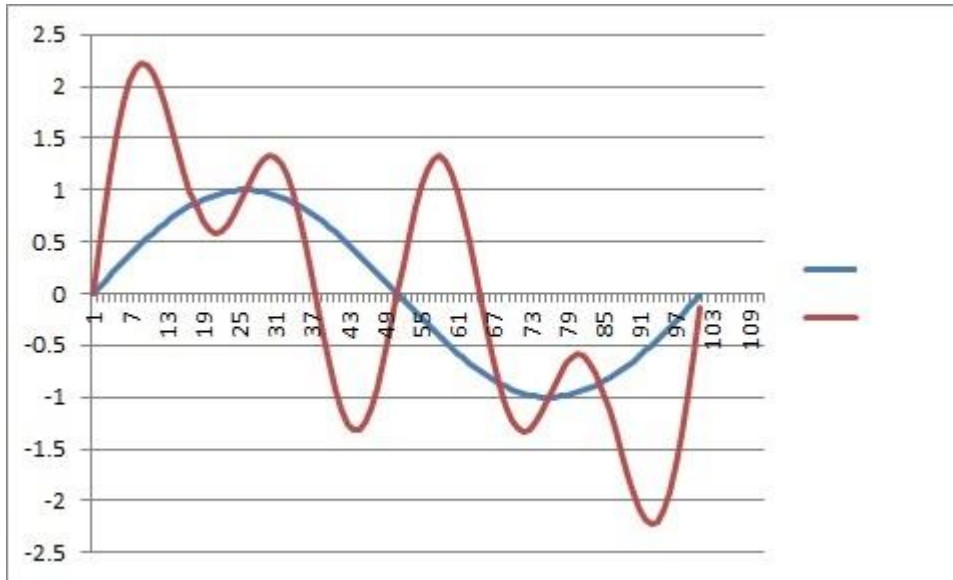
We can use this to spread out our data and then detect it at the other end

For an exercise we will send a 4 bit word, each bit will be represented by one of 4 basis functions:

$\sin(2\pi nt)$ where $n = \{1,2,3,4\}$



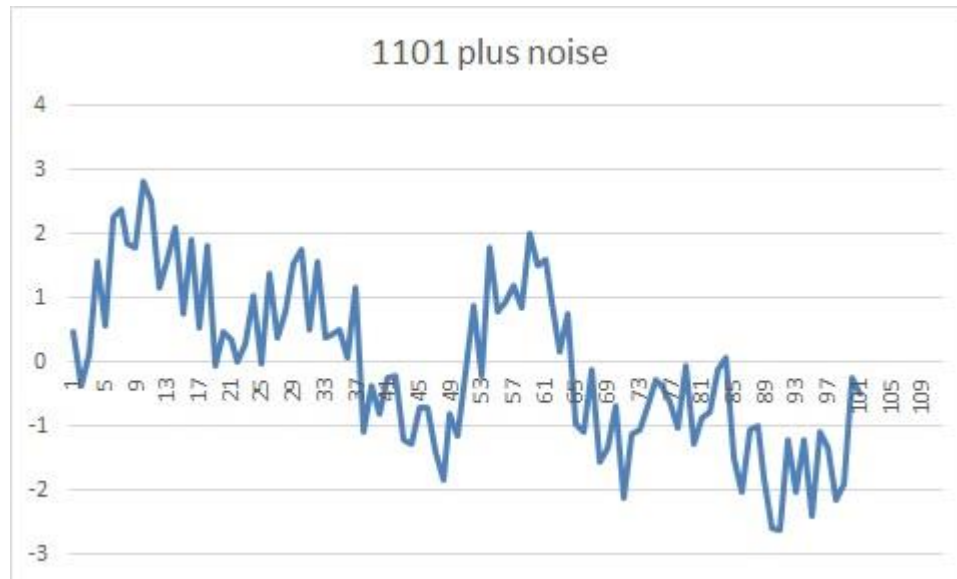
Lets transmit 1101 = 13 in decimal or 'C' in hex



To decode: Multiply the received signals by each of the basis symbols and add
i.e. integrate

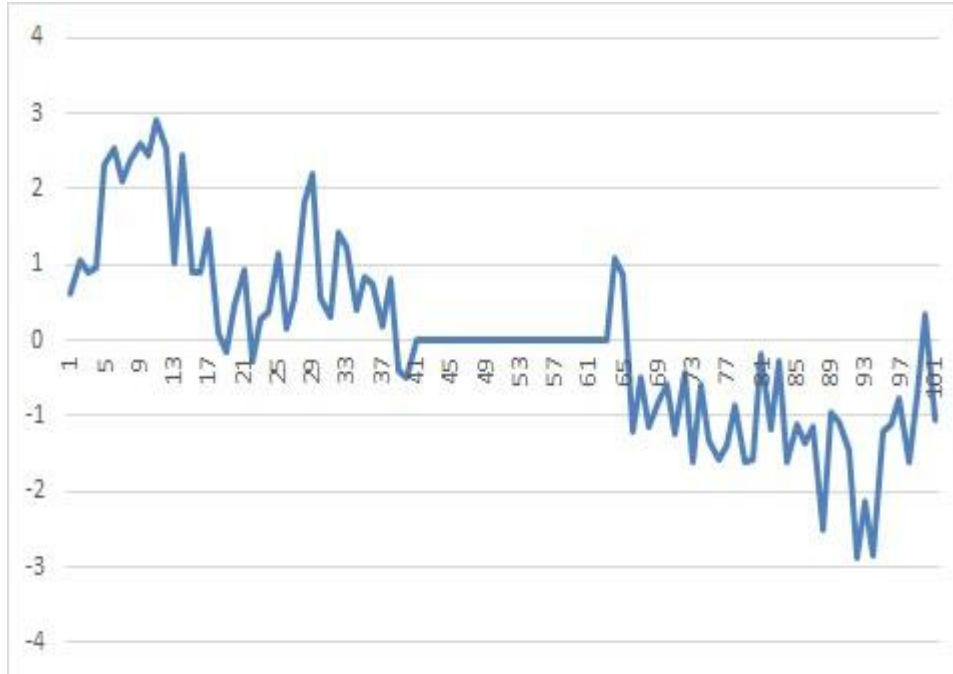
In this case the result is: 50, 50, 0, 50, so it's easy to tell 1's from 0's
but we don't have any noise.

1101 plus noise



Sums: 47.7, 47.7, -2.5, 47.7 so it is still easy to decode

How about some data loss?



Sums: 56, 37, 18, 31

Now it is getting harder to distinguish between 0 and 1