8 September 2018

Designing the Fox-1E PSK Modulator and FoxTelem demodulator

Chris Thompson, G0KLA / AC2CZ
g0kla@arrl.net

Keywords: BPSK Modulation, BPSK Demodulation, FoxTelem, Costas Loop, Gardner Carrier Recovery, Java

Abstract. Unlike the FM Fox-1 spacecraft, Fox-1E will transmit telemetry and science data to the ground using Phase Shift Keying (PSK). There is a long history of PSK within AMSAT and a wealth of knowledge. This paper discusses the design of a PSK generator with the LTC5599 quadrature modulator. It then provides an update on FoxTelem and the capabilities needed to recover the carrier with a Costas Loop and cleanly demodulate PSK in software.

Part 1: PSK Modulator

With the decision to use PSK on Fox-1E, we wanted to design a modulator that would work seamlessly with the existing IHU. PSK changes the phase of the carrier to convey information. In its simplest form it changes the phase by 180 degrees or pi radians and is called Binary Phase Shift Keying or BPSK. The diagram below from the ARRL Handbook (ARRL, 2017) shows that this is conceptually simple. We apply the data signal to the local oscillator port of a mixer and use it to chop the RF carrier from one phase to another.

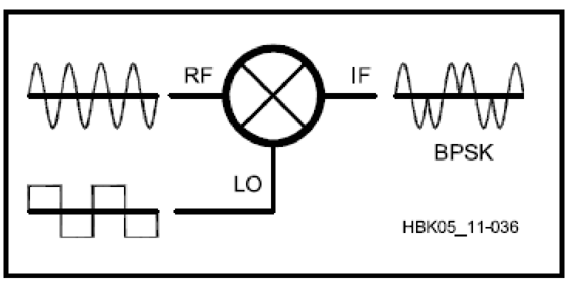

Figure 1 – BPSK Modulation, From the Section 10 of the ARRL Handbook

<u>Balanced Modulator</u>

BPSK is a double sideband suppressed carrier modulation and we can produce it from a simple balanced modulator (Miller, 1991). A simple test uses a balanced modulator such as the SA602 from the junk box.

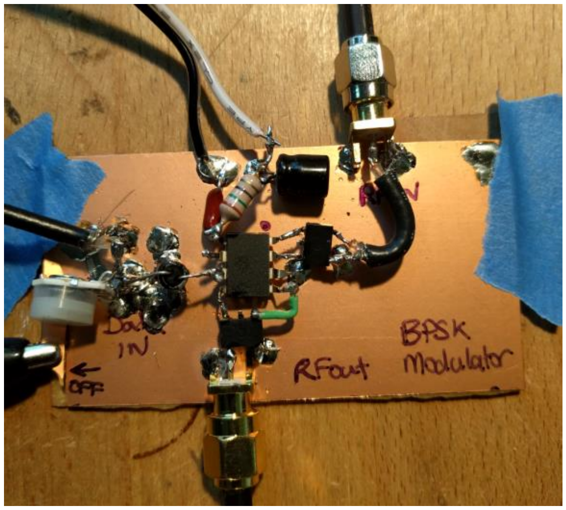

Figure 2 – A dead-bug SA602 prototype, one of many designs built

The Local Oscillator was driven with a 200mV p-p data signal. The circuit used 1:4 and 4:1 baluns to give balanced input/output and a better match to the 1.5k impedance that the SA602 has. The test signal was produced by an Arduino emulating the spacecrafts Internal Housekeeping Unit (IHU) because an IHU was not available.

This balanced modulator prototype worked well but gives a very wide signal as shown in the plot below from HDSDR. This signal is wide because a square wave has a very wide bandwidth.

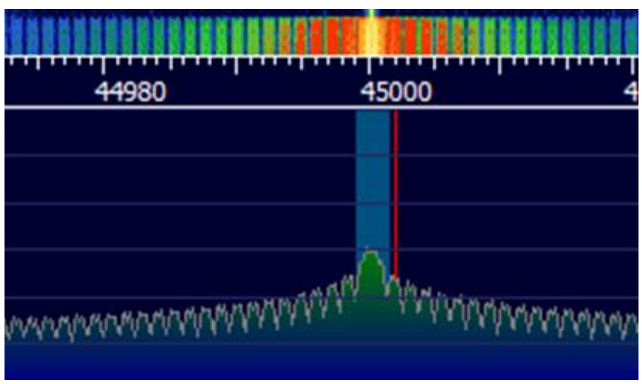

Figure 3 – Spectrum of BPSK signal with unfiltered bits

Filtering the Bits

The wide signal is addressed by filtering the input signal.

As a test an RC filter with a corner frequency of 600Hz was added to the prototype. It rolls off at just 6dB per octave. It is a 270ohm resistor and a 1uF capacitor.

The wave form is only slightly filtered as a result, so it looks like the image in the oscilloscope to the right.

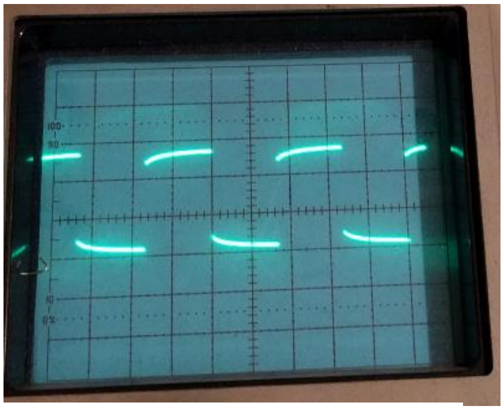

Figure 4 – Filtered bits

The filtered bit stream was applied to the mixer and the output spectrum is shown below. All other settings are the same, including making sure the drive level was adjusted to compensate for the filter loss. This demonstrates that only a gentle amount of filtering is needed to dramatically improve the bandwidth.

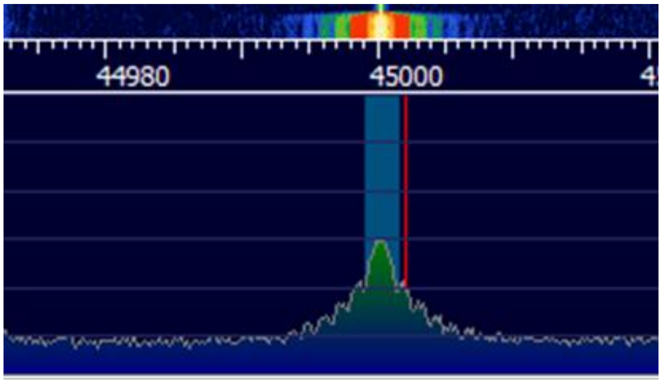

Figure 5 – Spectrum of BPSK Signal with Filtered Bits

The right filter does more than just narrow the bandwidth. Any filter introduces a delay so that one bit smears into the next (Miller, 1991). This causes inter symbol interference (ISI). Filter delay can not be avoided, but the best filter maximizes the amplitude in the middle of the bit and has zero output at the center of all other bits. This can be achieved with a raised cosine filter.

We followed the normal approach of using a root raised cosine filter for the transmitter and another root raised cosine filter for the demodulator. This gives a raised cosine response for the end to end channel. It was implemented as a short Finite Impulse Response (FIR) filter in the IHU on the spacecraft.

Quadrature (IQ) Modulator

To produce BPSK from a quadrature modulator, you only need one input. The other input can be held at a constant value. For example, if Q is set to zero and I varies between +1 and –1 then the output is BPSK. That is not intuitive to all people, certainly not to me. The I and Q inputs are used to set the amplitudes of the Cosine and Sine Local Oscillators. The local oscillators are then added to produce the output.

BPSK generation in this way is illustrated in the waveforms plots below, which show sine and cosine waves with 8 samples per cycle. Notice the first peak in the "SUM" or output waveform moves pi radians or 180 degrees from Figure 6 to Figure 8. Figure 7 shows I at -0.75, illustrating that the phase moves smoothly from one position to the other.

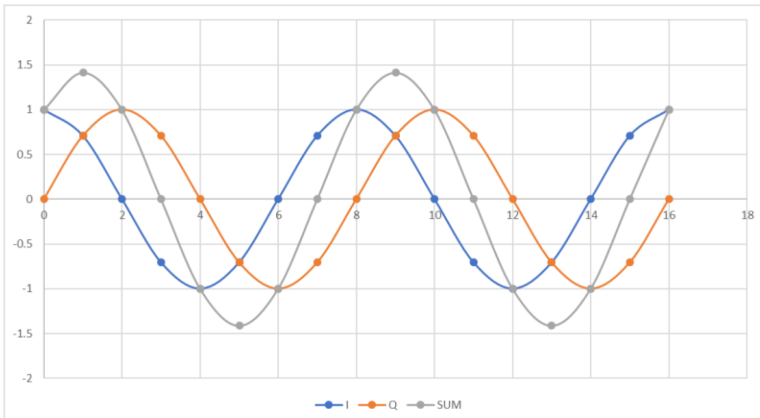

Figure 6 – IQ Modulator when I is at +1 and Q held at +1

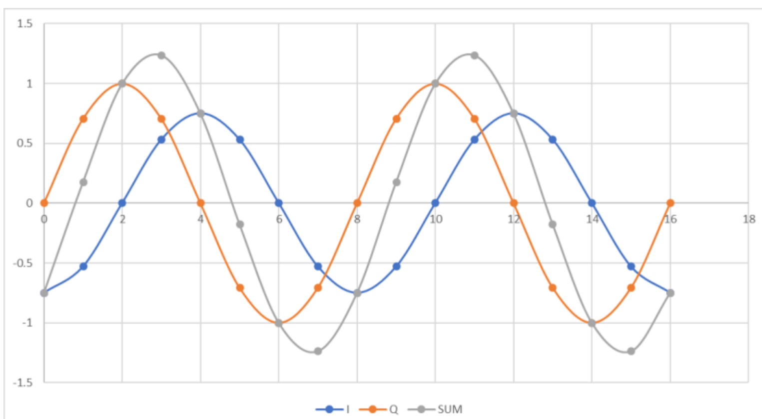

Figure 7 – IQ Modulator when I is at -0.75 and Q held at +1

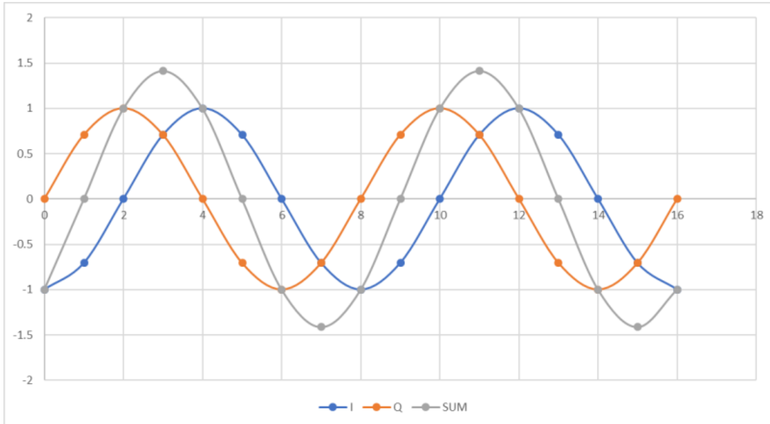

Figure 8 – IQ Modulator when I is at -1 and Q held at +1

IQ vs Balanced Mixer

Side by side testing was performed with a balanced mixer as the modulator, and an IQ modulator.

Here are our bits filtered with a root raised cosine filter.

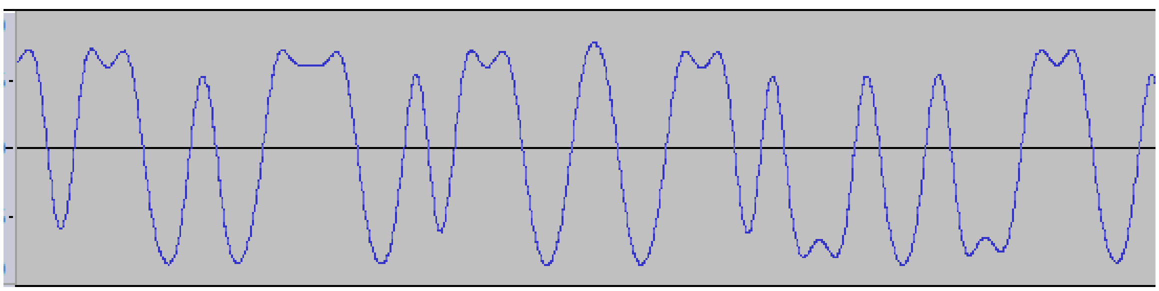

Figure 9 – Bits Filtered with a Root Raised Cosine Filter

When we play them through the IQ modulator, here is the spectrum. The signal was set to peak at -75dBm.

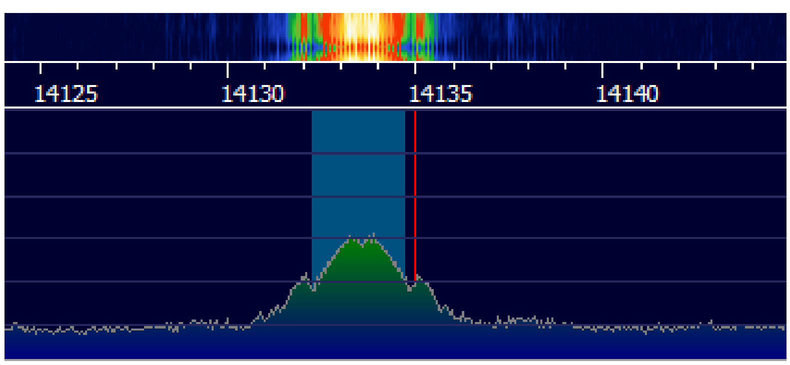

Figure 10 – Spectrum of BPSK Signal from IQ Modulator with Filtered Bits

Some careful measurements show the IQ modulated signal is 25dB down at +/- 1.77kHz. So it is 3.5kHz wide. The signal decodes well.

When we play the same audio through the balanced modulator we still get a decent signal, but it is not as clean as the IQ modulator. Again the signal was set to peak at -75dBm.

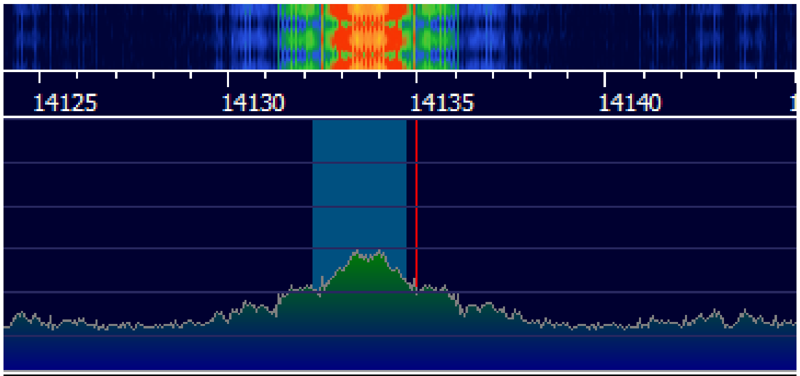

Figure 11 – Spectrum of BPSK Signal from Balanced Modulator with Filtered Bits

This signal is 6.4kHz wide at the -25dB point. It is nearly twice as wide. It decodes just as well, but the bit SNR is lower than the IQ modulator. These tests are shown in the graphs below, where the green line is the average. These were captured as the bits were decoded in FoxTelem.

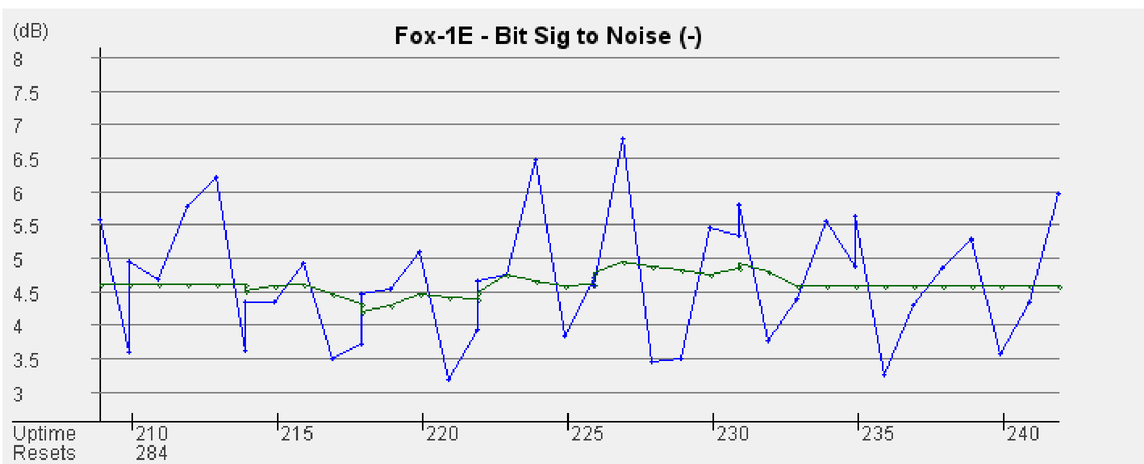

Figure 12 – Eye SNR for IQ Modulator

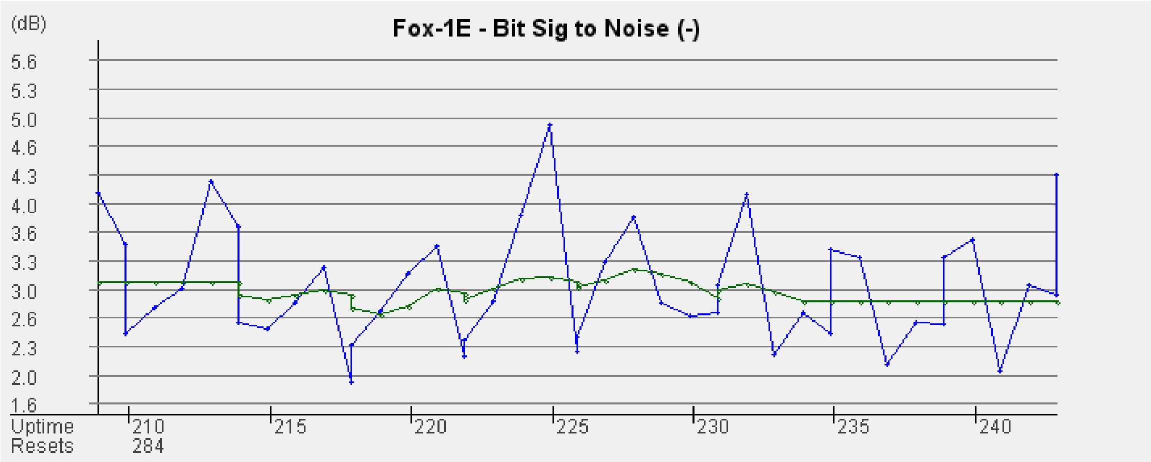

Figure 13 – EYE SNR for Balanced Modulator

The LTC5599 quadrature modulator

We choose the Linear Technologies LTC 5599 quadrature modulator as the BPSK balanced modulator. This chip had several key features that met our requirements:

- Works at our IF of 45MHz
- Low current consumption – 8mA if gain adjusted to minimum
- Configurable from our spacecraft IHU via a Serial Peripheral Interface (SPI) bus, which was already part of our spacecraft architecture
- Adjustable gain
- Adjustable carrier suppression to compensate for tolerances in components

It also has some useful additional features:

- On board telemetry such as temperature
- Able to support more complex modulation schemes in the future

The block diagram of the LTC5599 is shown below.

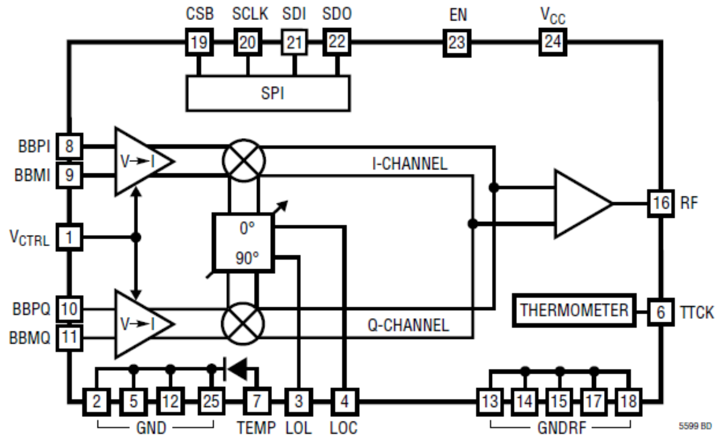

Figure 14 – LTC5599 internals

We needed to supply the LTC5599 with a balanced input and picked the Texas Instruments THS4531. This is a differential operational amplifier. It floats the IHU data bits around 1.4V as a balanced signal. This matches the requirements of the LTC5599.

A 45MHz Mercury Oscillator was chosen for the local oscillator. This is fed to the LTC5599 Local Oscillator port through a diplexer, as recommended in the data sheet.

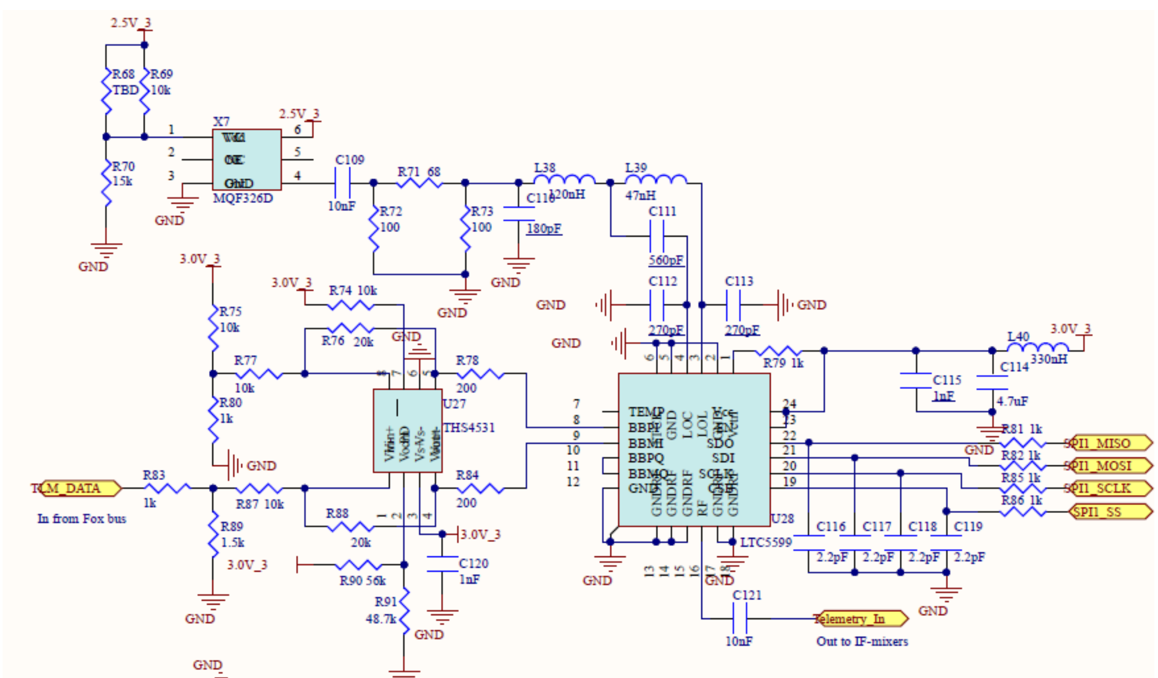

Figure 15 – BPSK Modulator for Fox-1E from 2017 Symposium Proceedings

Adjusting the Carrier

The diagrams below show the BPSK signal from the IHU before and after adjusting the carrier. We can adjust the I and Q offsets through the SPI interface to the LTC5599. We can also adjust the gain from the IHU and hence set the amount of power fed to the telemetry transmission.

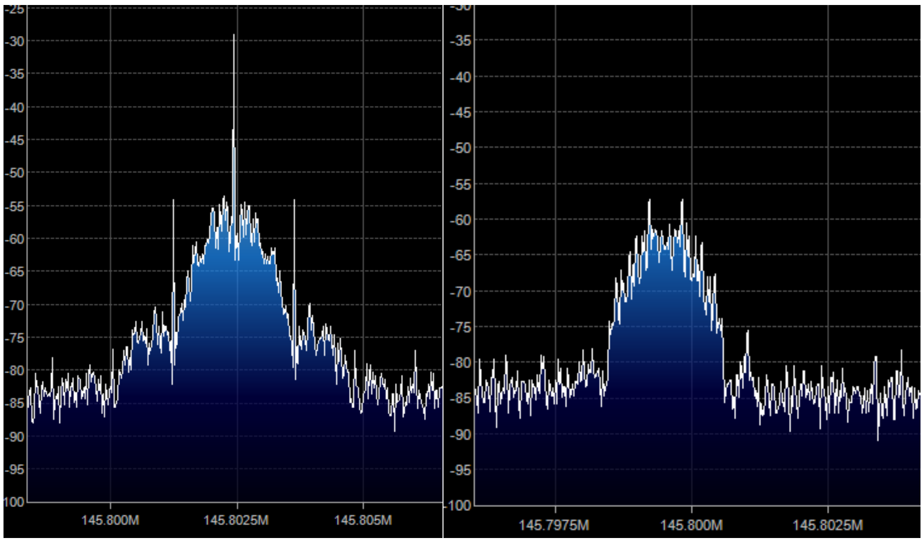

Figure 16 – BPSK Modulation before and after carrier suppression by the IHU

Part2: PSK Demodulator – using FoxTelem

Generating the PSK modulation is less than half the problem because decoding on the ground is always harder than encoding in space. This section discusses the decoder needed for PSK.

The data is encoded at 1200bps and has been Double Sideband modulated onto a 45MHz carrier and then mixed up to our 435MHz telemetry frequency for the spacecraft. Demodulation of a Double Sideband suppressed carrier signal is best achieved by mixing it with the original carrier. The two sidebands add and we get 3dB of gain. The signal can also be demodulated by an SSB demodulator, from a real radio or an SDR, but the 3dB of gain is lost. Either USB or LSB can be used in that case.

If the carrier is mistuned, even slightly, then the demodulation is impaired. With a small spacecraft, where the downlink frequency changes due to thermal conditions, we can't rely on a Doppler calculation to tune the carrier. We need to follow the signal accurately and insert the required carrier to demodulate it.

Coherent IQ Decoder

Coherent demodulation is more efficient than a non coherent decoder, especially with noisy signals (Best, 2007). Coherent demodulation requires not only recovery of the carrier but exact recovery of the phase of the carrier. Recovery is best achieved with a PLL and for BPSK, where the carrier has frequent 180 degree phase changes, it is recovered with a Costas Loop.

Historically the Costas Loop did not have a lot of support because it uses a quadrature oscillator. Like the phasing method of SSB, it relies on precise components and fell out of favor in analog designs. SDRs have brought the phasing method back into fashion because of their ability to simulate precise values and the Costas loop receives the same benefit.

A Costas Loop was added to the FoxTelem IQ receiver with the block diagram below.

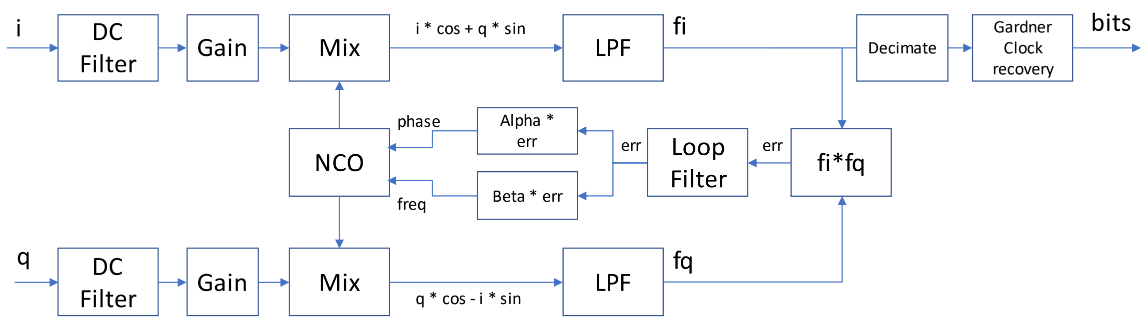

Figure 17 – FoxTelem Costas loop block diagram

I and Q samples are taken from the SDR receiver, typically a Fun Cube Dongle or an RTL Dongle. FoxTelem mixes (in software we do a complex multiply) the I and Q values with the Cosine and Sine values of the Numerically Controlled Oscillator (NCO). Which is a look up table of sine and cosine values that we step through at the sample rate.

The I output is then the recovered bit stream and an unwanted component at twice the NCO frequency. A Low Pass filter at the bit rate leaves us with the recovered bits, called “fi” in the diagram above.

The Q output is the phase error, again with an unwanted frequency component that we filter out. “fq” is zero when we are exactly in phase with the bit stream. It is non zero otherwise and proportional to the phase difference. The sign changes depending on the bit value, so we multiply fq by fi to neutralize the sign change. This gives us an error signal which tells us which direction and how much to move the phase to align the carrier.

The Costas Loop feedback is controlled by two key parameters which set the gain of the loop. The first is Alpha, which is a weighting of the error signal and is used to adjust the phase of the NCO. Alpha was set to 0.1.

The second is Beta, which is proportional to alpha squared and is used as a weighting of the error to control the frequency of the NCO. Beta was initially set to 16 * alpha * alpha and as high as 1024 for large frequency ranges.

These two parameters are required because this is a second order loop. We can not get the loop to converge reliably with a simple gain value.

The low pass filters in the arms of the Costas loop needed to be very fast with little delay or the loop would not track. There were implemented as 1200Hz 0.5% ripple 4 pole Chebyshev Infinite Impulse Response (IIR) filters. (Following a design from Smith, 2003)

The Loop Filter is simpler still and is required to smooth the error signal. It is a Single pole IIR . (Smith, 2003)

FoxTelem is shown below locked to the carrier frequency with the Costas loop in operation. The double sideband demodulation has recovered the bits. Note the raised cosine shape.

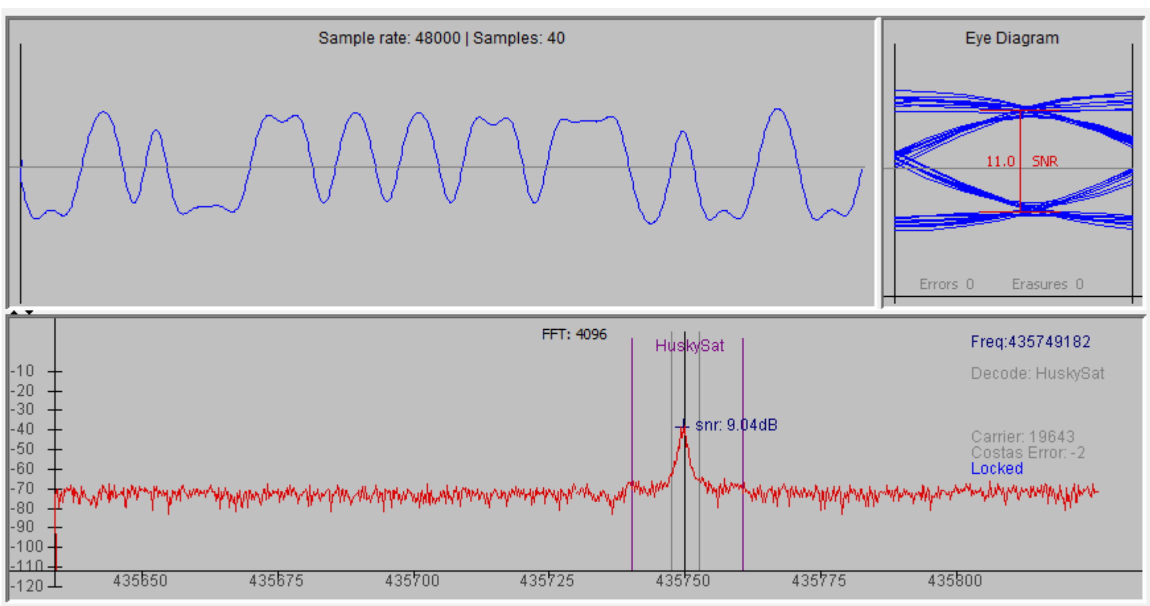

Figure 18 – FoxTelem with Costas loop locked to PSK signal

False Lock

The Costas loop frequently locked onto slightly the wrong frequency as shown in the FoxTelem debug output below. This is a false lock and is common when the frequency error is of the same order as the bit rate. In the example below it is locked 600Hz off. This is a serious issue that makes the Costas Loop decoder unusable in about 25% of lock attempts.

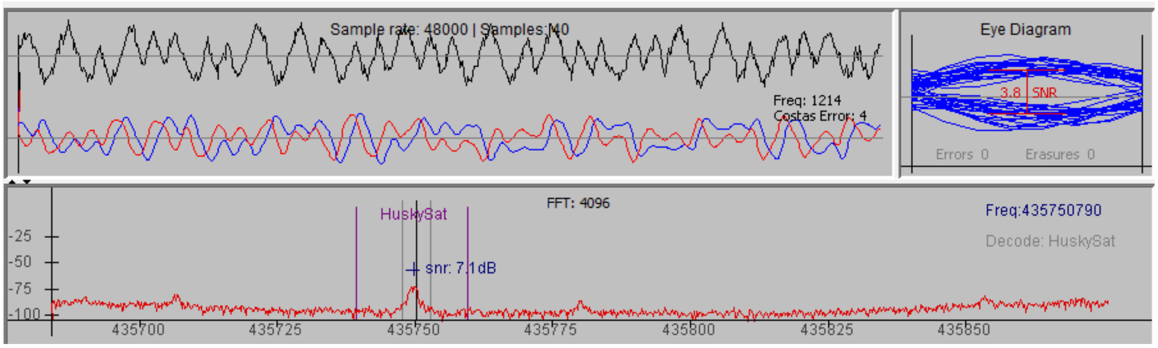

Figure 19 – FoxTelem showing False Lock, and debug information. The black trace is the psk signal, blue trace is I and the red trace is Q (the error signal). Q should be zero, but it is not.

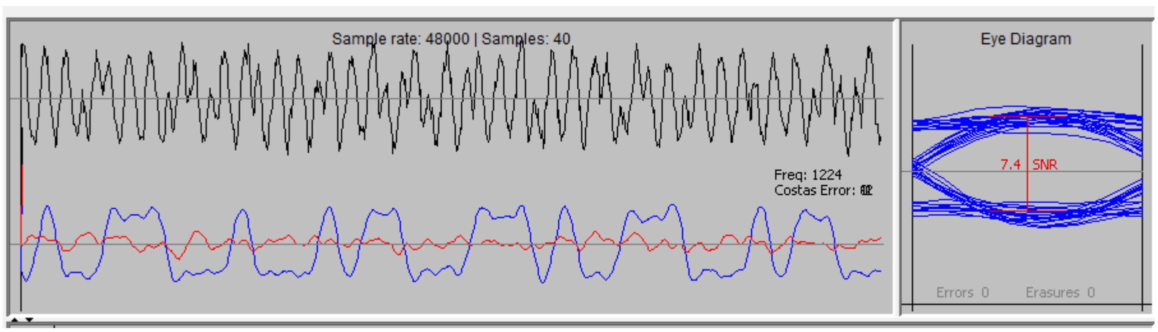

Figure 20 – FoxTelem locked onto carrier

Tom McDermott, N5EG, recommended a review of the patent literature and in particular US Patent US4713630A (Matthews, 1986). This is an expired patent by David Matthews that implements a false lock defeat mechanism. This was originally implemented in hardware but looked like a good candidate for a Software solution.

The original design is shown below.

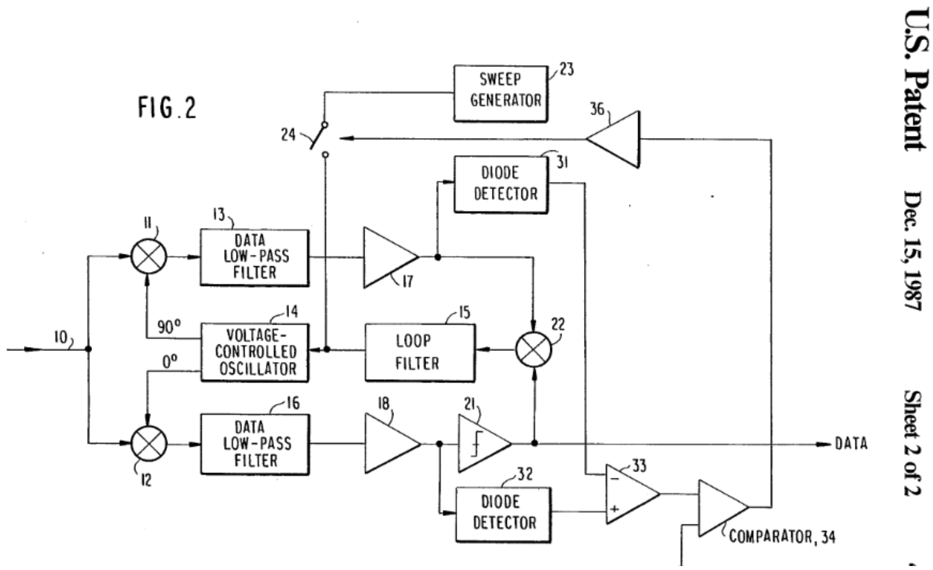

Figure 21 – Costas Loop that defeats false lock from expired US Patent

Implementing this in software was straightforward. “Diode Detection” needs to be full wave rectification and in software is as simple as this:

```
double fullwaveRectify(double in) {
   if (in < 0) return -1*in;
   return in;
}
```

The Costas loop was modified to calculate a “locklevel” value from the filtered signals fi and fq. This worked best if some automatic gain control was used.

```
ri = fullwaveRectify(fi*gain);
rq = fullwaveRectify(fq*gain);
lockLevel = ri - rq;
```

The lock level was summed over a bit and used as a control signal. When it was above a threshold e.g. 5, then the loop was locked, otherwise it is close to zero.

When the loop is not locked then the decoder scans the expected frequency range. In software we just increment the frequency of the NCO by a fixed amount. I had to determine experimentally how fast the decoder should scan so that the Costas loop will still lock when it is at the right frequency.

This design works and false lock is now prevented.

Clock Recovery

With the Costas Loop implemented we are left with the issue that the bits are not centered. The screen shot below shows the first few demodulated bits from a signal, together with a small spike showing where FoxTelem made the bit decision. We can see that it is almost in the center of the bit. But as the bits flow past, the decision point continues to scroll to the right moving from one bit to the next. This causes the eye diagram to continually scroll and nothing is decoded.

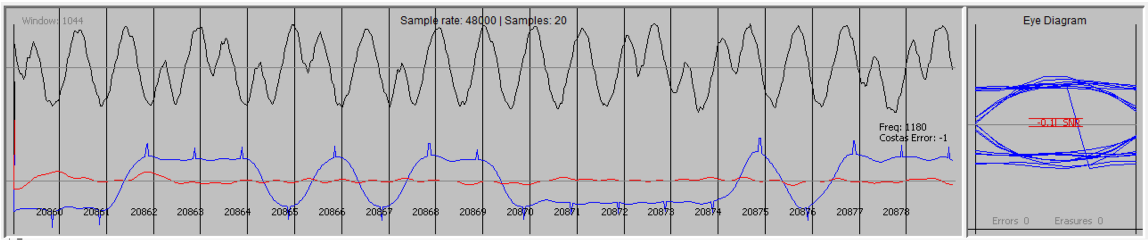

Figure 22 – FoxTelem showing Bit misalignment in "Debug Values" mode

It might not seem obvious, but just because we have recovered the carrier does not mean we have aligned with the clock that the spacecraft used to create the bits. The clock is nominally 1200Hz, but it is subject to thermal variation as it is based on the crystal oscillator of the IHU. The bit decision point and the eye diagram may scroll in one direction or another depending on the speed of the IHU clock compared to the clock in the decoder. If we do not compensate, then nothing will decode.

Gardner Clock Recovery

To recover the clock and adjust our timing we count the samples as they come in and remember where we are in the bit. In the illustration below we show three sampling points. Yn-2 is the value of the previous bit, in this case -1 indicating a zero bit. Yn is the the value of the current bit. Yn – 1 is half way in between. In the example below we know that our bit is nominally 4 samples long, so we count from 1-4 in a loop. When the counter is 2 we sample the bit, when it is 0 (or 4) we take the middle sample.

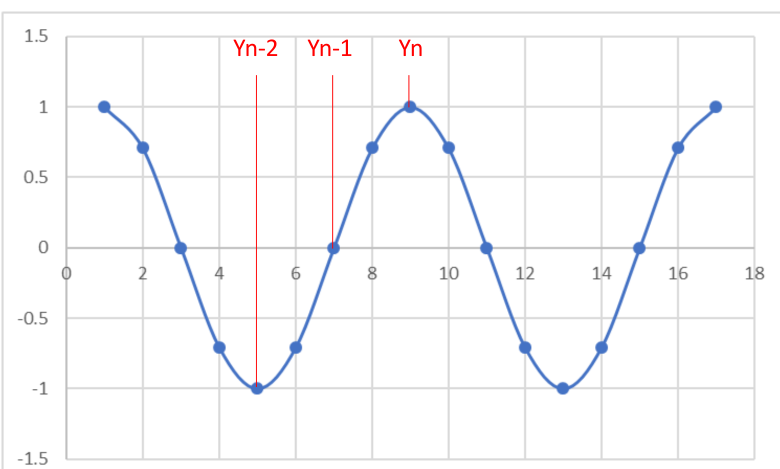

Figure 23 – Sampling points for Gardner Clock Recovery

The error signal is then given by:

Clock_error = (Yn – Yn-2) * Yn-1

If the clock error is positive then our sample is late. In the example below we see that the error is (0.75 - -0.75) * 0.75 = 1.125. So we adjust our sampling so that our next Yn sample point will be at sample 13 and not sample 14. If the clock is early then our error will be negative and we do the opposite.

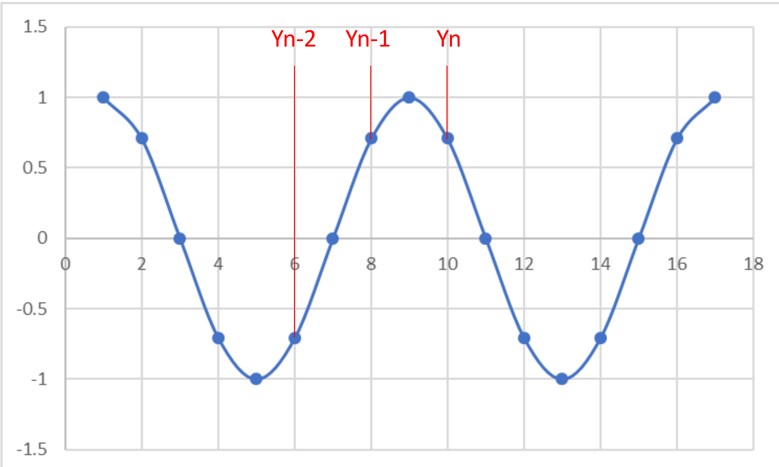

Figure 24 – Late sampling in Gardner Clock Recovery

In the clock recovery algorithm we have to make sure that we don't accidently sample Yn or Yn-1 twice when we adjust our position in the bit. Assuming that does not happen, then the Gardner clock recovery algorithm works well. Bits are recovered and sent to the RS Decoder for Forward Error Correction and eventual decoding.

References

[1] American Radio Relay Leauge (2017), Handbook.
[2] James Miller G3RUH (1991), The Shape of Bits to Come. *Amsat-UK's Oscar News, 1991 Apr No.88 p.29-37*
[3] John P. Costas (1956), "Synchronous communications". Proceedings of the IRE. 44 (12): 1713–1718
[4] Feigin, Jeff (2002). "Practical Costas loop design" (PDF). RF Design: 20–36
[3] Roland E. Best (2007), Phase-Locked Loops: Design, Simulation and Applications, 6e
[4] Steven W. Smith (2003), Digital Signal Processing, A practical guide for Engineers and Scientists
[5] Richard E. Lyons (2011), Understanding Digital Signal Processing, 3e
[6] David Matthews (1986), BPSK Costas-type PLL circuit having false lock prevention. US Patent US4713630A.