DSP Modems
It's Only Software

by Dr. Robert W. McGwier, N4HY

Are you as tired of buying new modems as I am? There is an alternative and I think a superior one in several regards. From the title, and the joint paper with Tom W3IWI[1], it should be obvious that we are talking about doing modems on DSP engines. Using a single chip processor as the basic building block, a low cost, high speed modem can be programmed to operate through *unconditioned* radios. If you want a different modem ten minutes later, you just change the software driving the chip. The primary advantages stated more explicitly are (1) a single piece of hardware can make and demodulate the phase or frequency modulated synchronous signal of your choice, and (2) the software can do very effective adaptive equalization to ameliorate the bad things the radio (filtering) is doing to your data signal.

Lets take an example and go through a description of what software we need on the Texas Instruments TMS320 to do the modem functions for a. a FO-12 BiPhase PSK demodulator. We wish to use adaptive equalization and a fairly general demodulation scheme that can apply to other types of phase modulation schemes as well (QPSK, etc.). We will work our way "backwards" in the demodulation process from carrier generation to phase detection of a PSK signal. This is in order of increasing mathematical complexity but not necessarily computational complexity.

Sine Wave or Tone Generator

Common to both the receiver and transmit side of a modem on this chip is a sine wave generator. The process behind generating a sine wave or tone is fairly straightforward. It is usually called frequency synthesis. We store a table of values of sine at 64 values from the first quadrant. Stored will be the values of sin(X), where X = $0, \pi/128, 2*\pi/128, \ldots, 63*pi/128$ radians. For finer resolution, we can store a table of values $sin(Y)$, where Y = $0, \pi/8192, 2*\pi/8192, \ldots, 63*\pi/8192$. between 0 and n/128 or do interpolation. We will use the second table approach and do mod 16384 arithmetic for calculating angles. The highest order two bits are quadrant, the remaining twelve bits will be used to index into the the gross and fine value table as will be described below. For angles outside the first quadrant, or cosine, or to use the fine table approach we use a fundamental trigonometric identity

$$(1) \quad sin(X+Y) = sin(X)cos(Y) + sin(Y)cos(X).$$

In our dual table approach at frequency sysnthesis, let X one of the gross values indexed by the six bits just below the quadrant bits of the current phase. Y will be a fine resolution angle got ten from the low order six bits of the phase. Since Y is small, a moment on your calculator will show you that $|cos(Y)-1.0$ is so close to zero that we will replace $cos(Y)$ with 1 in our formula. Y is always between 0 and $\pi/128$. Our formula' becomes

$$(2) \quad sin(X+Y) = sin(X) + sin(Y)cos(X).$$

where sin(Y) is read out of the fine table and $sin(X)$ and $cos(X)$ are read from the coarse table. Let Φ be the phase of a sine wave or carrier. To get a tone of constant frequency, we just add a constant value at each sample time to the old value of Φ. We wrap back to zero when the Φ value goes past 2π which is 16384 in our arithmetic. The size of the value we add between each sample output value corresponds directly to the frequency since it meaures how fast we wrap ourselves around the circle. The implementation we have on the chip now gives fourteen bit resolution of the circle and 0.5 Hz frequency resolution from 0.5Hz to about 4 Khz given a sampling rate of 8.192 Khz. We can use this algorithm to generate FSK, M-ary PSK, and others.

Carrier Tracking

To demodulate a signal carrying digital data by means of a phase shift keying approach, we need to know the carrier frequency and phase so that we can do noise rejection by using narrow filters and to recover the clocking of the digital data so that we can reasonably expect to be able to distinguish a zero from a one. In the PSK modem used to demodulate JAS-1/FO-12 both the Clark/TAPR adaption of the JAMSAT modem[2] and James Miller's PSK modem[3] use a *phase locked* loop, hereinafter, PLL. Gardner[4] has written the definitive treatise on analog PLL's. In it can be seen how much effort has gone into making analog PLL's work. When done in software on these DSP engines, many of the difficulties inherent in making an analog loop disappear. In a digital representation, you can make an attempt at doing a minimum variance unbiased estimate of the current frequency and phase rather than trying to zero some phase error. The free running frequency of a digital "VCO" is wherever the frequency was last estimated to be located in a second order PLL (the only type we will consider). So there is not this tendency to pull away and back to the "natural" free running frequency of the VCO in an analog implementation. Thus temporary loss of lock is not as disastrous as in the analog implement ations.

Our modem receiver which will be a PLL is a trivial program on DSP chips. The tone generator we described earlier will be our VCO. Suppose our incoming complex signal is S_r and the VCO produces S_0. Our "phase de-

tector" is simply $S_r - S_0$ = AS. Ask W3IWI how much effort he expended in getting just the right parts and setup so that he would have no DC offset in the phase detector in JAMSAT/TAPR PSK modem. We feed this through a loop filter, which when combined with natural integrator in our tone generator will be our second order loop filter. Take our current estimate of the frequency, the constant we add to the-phase at each sample time, and add to it a constant k_1 * AS and the old phase. This is the phase we will use at the next sample time. To make a new estimate of the current frequency we multiply the old estimate by k_2 and add it to k_3 *AS. These values of k_i, $i = 1,2,3$ are (in Kalman filter parlance) called "gains." Simply put, it is how much we wish to believe the current phase error AS when making changes to our estimates of the phase and frequency. The updated frequency estimate just determined is fed to the tone generator for the next sample period and the process is repeated. This phase locked loop is 13 instructions on the T.I. TMS32010, and with the implementation we are using (Delanco-Spry Model 10) the processor is running at 6.25 Mhz. The phase detector and loop filter takes 2 μs. When this is combined with the sine generator algorithm, the total time for the entire PLL algorithm per sample is 17 μs. A block diagram of the PLL algorithm from the phase detector to the VCO output is given in figure 1.

Demodulation

It may seem odd that in attempting to extract zero's and one's from a PSK modulated carrier, that we must somehow remove the effect of the modulation in some way to be able to make adequate decisions about which bit is being transmit ted. Let S,(t) represent the received signal at time t. To do this, this the extracted carrier is split into in-phase (I) and quadrature (Q) components by use of a Hilbert transform. We will call this S,(t) and $\hat{S}_r(t)$ respectively. A discrete-time Hilbert transform is a way of taking a sequence of comples signal samples and rotating the phase of the Fourier components forward by 90^o. This is very difficult to do over a range of frequencies in analog devices but is easily implemented in software by using an FIR (finite impulse response) filter[5]. Simply put, an FIR filter is a sequence of N numbers that you multiply the by the preceding N signal samples at each time sample t to get a filtered output.

$$y(t) = \sum_{i=0}^{N} x(N - i) * h(i) \qquad (3)$$

where $y(t)$ is the filter output, $x(N$ -i) are the current and previous N signal samples and the h's are the filter coefficients. This formula is "THE" formula is DSP software. This structure is used to produce bandpass filters (low, high, complex bandpass, etc.) and the adaptive equalization filters whose coefficients *h axe* allowed to change in time according to some preset rules. It is the same formula used to compute discrete Fourier transforms. The only place we do not use FIR filters in this modem will be in the loop filter described above. The schematic found in [5] for these filters is in figure 2. The coefficients are for a bandpass filter and are found using the Remez Exchange Algorithm[5]. Rather than go through the mathematical formulae associated with the Hilbert transform demodulator, we give it to you in block diagram form in figure 3.

Referring to figure& the (Q) channel output, $\hat{S}_r(t)$ is mixed with carrier references I and Q and the (I) channel (the unshifted input samples). The real and imaginary parts of the signal are taken from the adaptive equalizer and "arctangent" is used to compute the phase angle. On the TMS32010, the arctangent is a table of 25 to 30 words along with a linear interpolation algorithm that will give the arctangent to a few tenths of a degree accuracy. The equalizer filter we will use will be a twenty tap (twenty coefficients) affair and its evaluation takes 20 μs. The outputs of the demodulator are phase angles which are used to feed the PLL. In figure 3, this demodulator output will replace the phase detector. Other outputs are from the real and imaginary leg equalizers are the data encodings and these are fed to a decision algorithm which decides which bit pattern we are trying to encode. This decision algorithm also output s the error from the nominal value of the encoding and this is used to modify the values in the adaptive equalizers. The arctangent evaluation takes 18 μs. Most modems use scramblers. This does NOT mean encrypt ion! This means we are trying to prevent bad sequences from occuring and causing the demodulator to lose the ability to follow the baud transitions (clock). We can implement a seven stage shift register in 18μs and the bit error rate will be improved considerably.

We are just beginning the modem development work and if you have some expertise in the area, please let us know so that we can move more quickly toward a nice product for the amateur community.

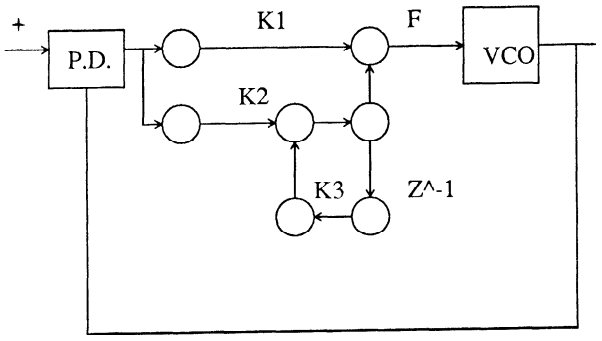
[1] Clark and McGwier, "Digit al Signal Processing", this volume.

[2] JAMSAT, "A PSK Modem for JAS-l", QEX, April 1986.

[3] Available from AMSAT-UK.

[4] Gardner, "Phase Locked Techniques", Prentice Hall, 1972.

[5] Rabiner and Gold, "Digital Signal Processing", Prentice Hall.



Digital Phase Locked Loop

FIR Filter

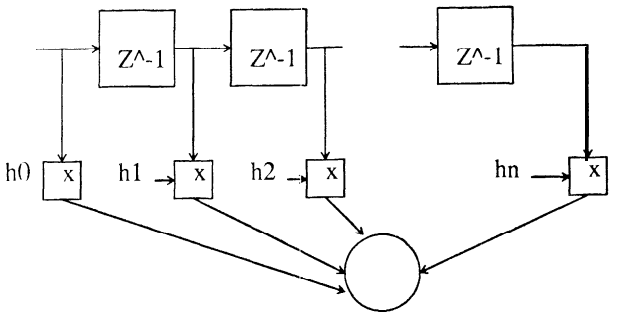


Figure 2

Hilbert Transform Demodulator for BPSK (M-PSK is done by doing modulo pi/m)

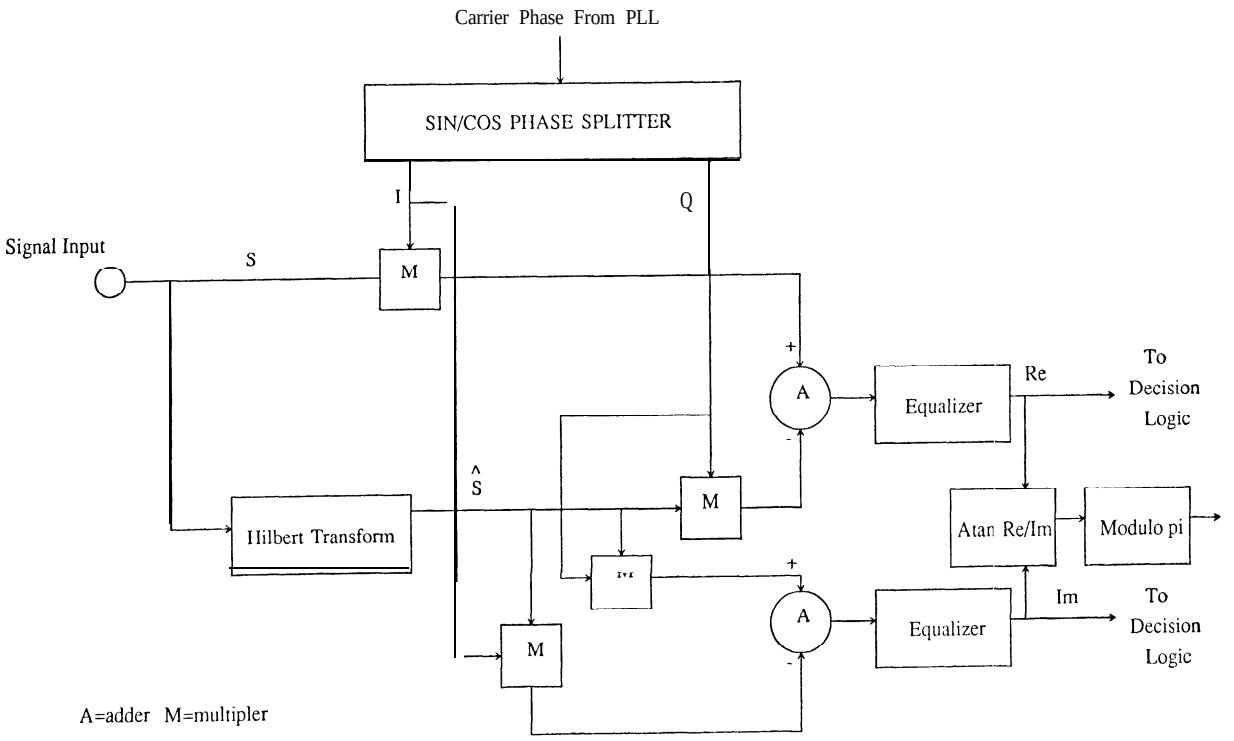


Figure 3