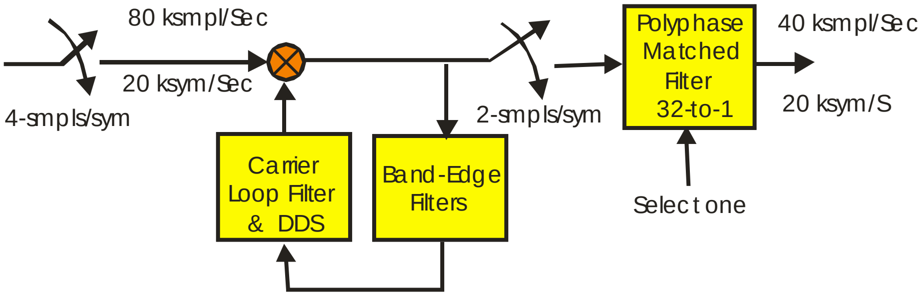
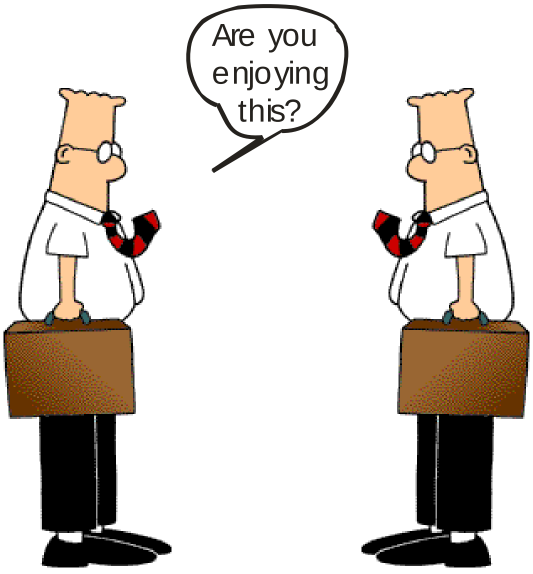
EE-655 MODEM DESIGN
Fall 2015

We are going to simulate the components of a BPSK modem receiver. To exercise the receiver we have to first form a modulator.

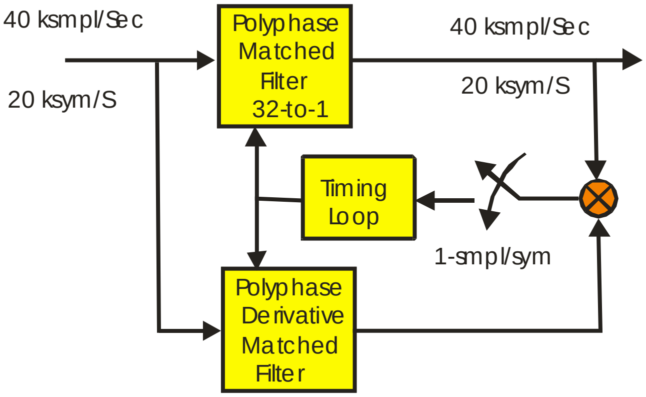
A. Design and implement a BPSK modulator with a SQRT Nyquist shaping filter with 40% excess BW operating at 8 samples per symbol. Form 2000 symbols. Symbol rate is 20 k-symbols per sec. program and form figures showing:
   - i) 200-symbols of the real part of the baseband time series.
   - ii) Windowed Spectrum
   - iii) Eye diagram
   - iv) Constellation diagram

B. Simulate a frequency offset between modulator and demodulator by spinning the modulated signal with a complex heterodyne spinning $2\pi/100$ radians per sample. Down sample the received and spinning sequence to 4-samples per symbol staring at sample 2 (to insert a time offset). Deliver this sequence to the frequency lock loop block diagram shown below. Program the band edge filters and the poly-phase matched filter. Form figures showing the spectra of the input signal and the two band edge filters carrier loop filter output with the loop disabled. Show the 4-to-1 down sampled spinning constellation from the polyphase matched filter when the correct phase is selected to align output with offset time series. Show the phase profile of the input and output phase accumulators as well as the loop filter input and output during the frequency acquisition. Also show the signal and band edge spectra after acquisition.

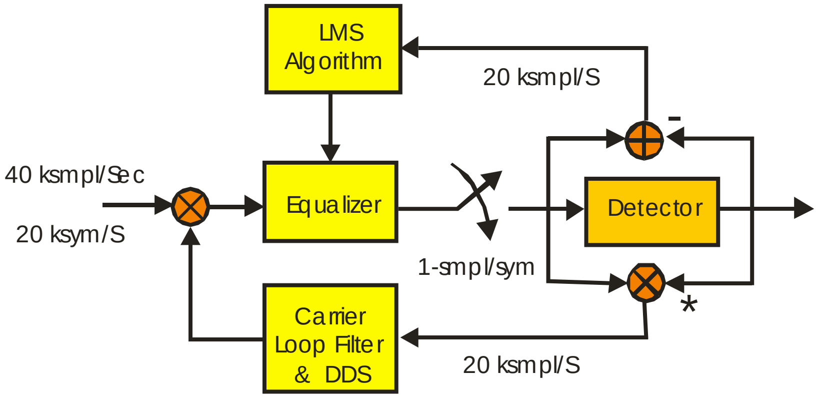




C. Simulate the timing recovery loop following the frequency lock loop. The block diagram for this loop is shown below. Design the polyphase matched filter and derivative matched filter. Show impulse response and frequency response of the two prototype filters. Show the group delay phase response of the polyphase matched filter arms (single figure). The input to this loop is the output from the previous (frequency lock) loop. Program the timing recovery loop and form figures showing the phase accumulator and integer part of phase accumulator time responses during timing acquisition. Show the input and output time series of the loop filter during timing acquisition. Show the constellation prior to and after the timing acquisition.


40 ksmpl/Sec
20 ksym/S
Polyphase Matched Filter 32-to-1
40 ksmpl/Sec
20 ksym/S
Timing Loop
1-smpl/sym
Polyphase Derivative Matched Filter

D. Simulate the equalizer and phase locked loop of the figure shown below. The equalizer will contain 40 taps.

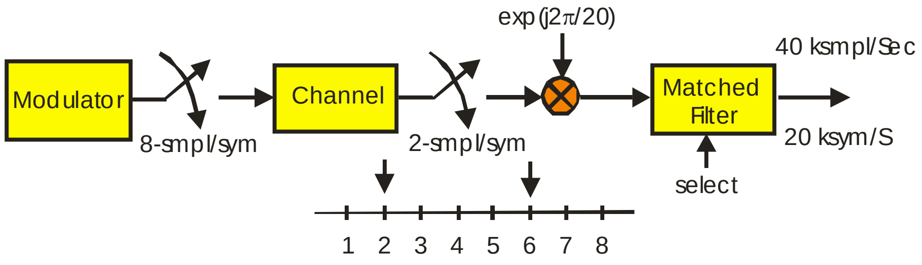


To test this loop we form its input signal from the flow diagram below. Pass the modulator output trough the channel model with impulse response shown.

$$H(z) = 1 + 0.2Z^{-5} + j0.05Z^{-12}$$

The output of the channel is down sampled, as shown, 4-to-1 starting on sample 2, to simulate the pair of down sample operations in the frequency locked loop and

the timing loop. The down sampled output is rotated 2π/20 and passed through the matched filter with filter arm selected for best timing alignment. This pseudo channel simulates channel distortion, a time offset, and a phase offset. The equalizer and phase PLL will process this signal.
Show the learning curve of the equalizer. Show the time series at input and output of the PLL loop filter. Show the phase accumulator time series. Show the constellation prior to and after operating the two loops.



E. By now we know how each component in the processing chain operates. We now operate the full modem. Have the modulator deliver samples through the channel, spin the channel output at 2π /100 rad/sample, down sample 2-to-1 starting at index 2, and deliver the channel distorted, time offset, and spinning sequence to the frequency locked loop. This output is down sampled 2-to-1 and fed to the timing recovery loop which in turn is presented to the equalizer and phase-locked loop. Show the signal progression and the loop control time responses for each loop in the sequence.

```
function [hh,gg,g]=
band_edge_harris(m_smpl,alpha,m_dly)
% band_edge_harris(m_smpl,alpha,m_dly),
% hh is sqrt Nyquist matched filter
& gg     positive freq band edge filter
% g      baseband band edge filter
% m_smpl samples/symbol, alpha excess bw
% m_dly is delay in symbols to filter center
% n_len=2*m_dly+1

hh=rcosine(1,m_smpl,'sqrt',alpha,m_dly);
hh=hh/sum(hh);
nn=length(hh);
tt= (-m_dly*m_smpl:1:m_dly*m_smpl)
g=sinc(2*alpha/m_smpl*tt-0.5)...
 +sinc(2*alpha/m_smpl*tt+0.5);
g=g/sum(g);
mm=nn/2;

% now translate center to (1+alpha)*fs/2
phi=(1+alpha)*(-mm:0.5:mm)/(2*m_smpl);
phi=phi(2:2:length(phi));
gg=exp(j*2*pi*phi).*g;
```

0