EE/TE 4385

Lab 6: PAM Receiver

Lab Report Due: 10/18/06, 2PM,

Pulse Amplitude DeModulation (Ideal): LabVIEW Implementation

Programming:

The following steps describe how to build a VI which implements Ideal Pulse Amplitude Demodulation. Download PAM-DeModulationTemplate.vi from the course website. Inspect the front panel and block diagram that has already been created for you. When this VI is completed, you will be able to see the time and eye diagram representation of the demodulation PAM signal. The following front panel represents the operation of a completed VI:

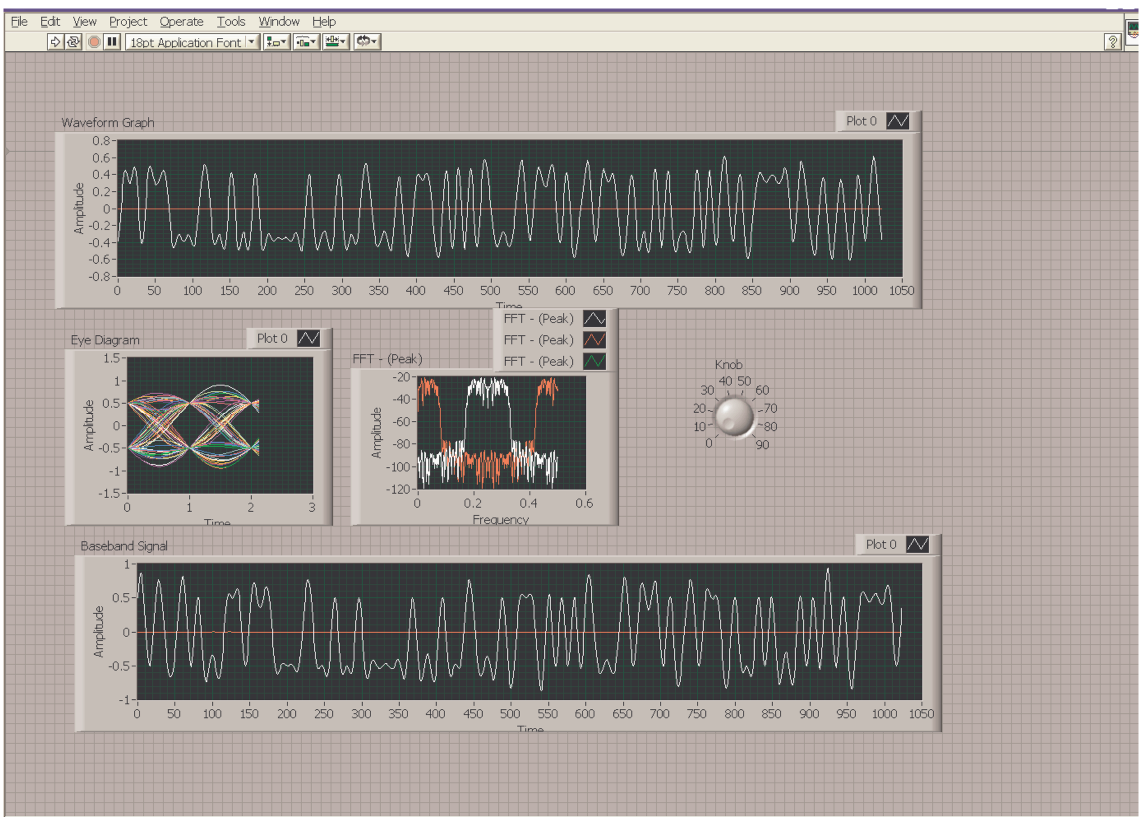

Figure 1 – Completed 2-PAM Transmitter/Receiver Front Panel

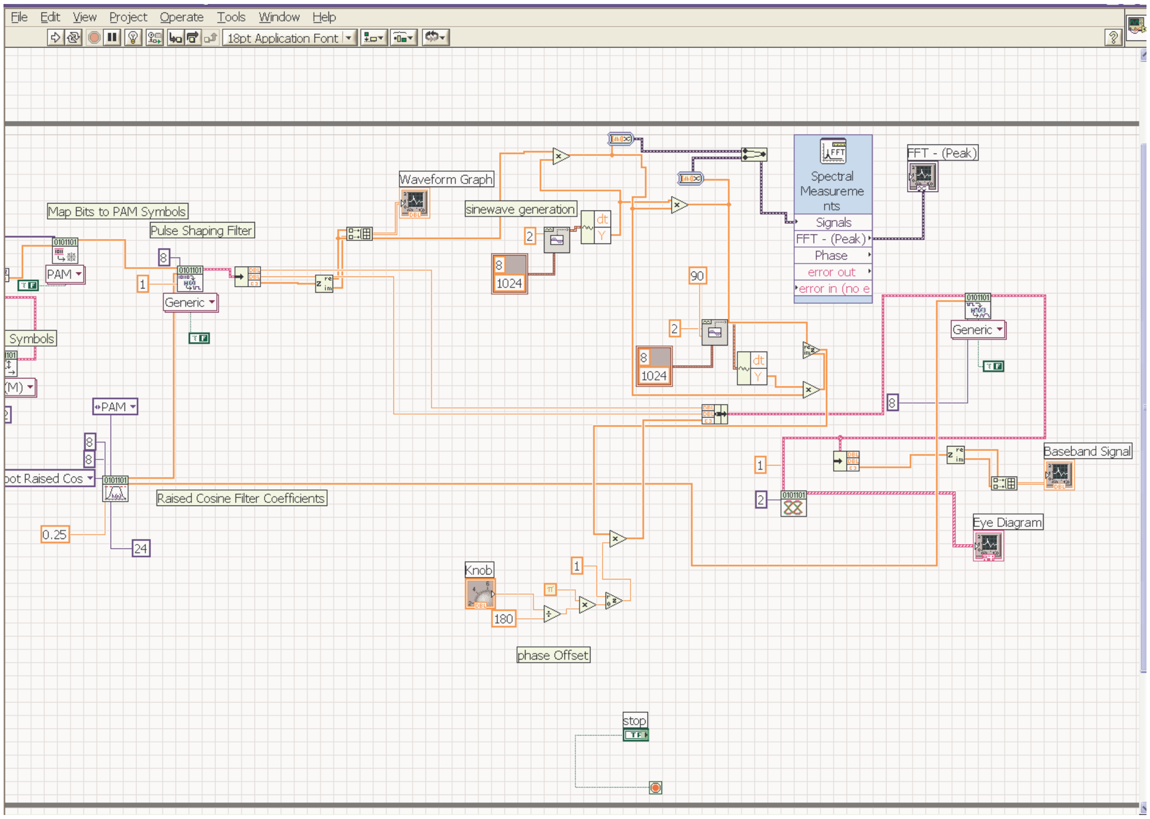

Figure 2 – Receiver Side of 2-PAM Transmitter/Receiver block diagram

- In addition to PAM transmitter VIs, you need Sine Waveform.vi, Spectral Measurements (Express VI), Apply Matched Filter.vi, and MT Format Eye Diagram(complex).vi.,
- Run your completed LabVIEW PAM Transmitter/Receiver. Stop and copy your block diagram and front panels
- Knob on the front panel introduces carrier phase shift to the received PAM signal. Dial the knob to observe the effect of phase offset on the demodulated signal.
- Change the alphapet size provided by the constant value (default is 2) wired to Generate System Parameters.vi to observe demodulated 4-PAM signal

PAM Receiver: C6713 DSK Implementation Using CC

For this part of the lab, you will demodulate the (upconverted) PAM signal generated by your neighboring group.

Modifications to PAM Transmitter

- Use your project files from your PAM transmitter lab. We will add additional functions to build the 2-PAM receiver.
- Create a lab6 project directory under myprojects subdirectory in c:\ti folder.
- Copy your PAM project files into this directory. Make sure that your program still compiles without any problem. Keep the sampling rate 8 KHz.
- Since we want to obtain raised cosine response from overall system, we need to design root raised cosine FIR filter with the function ‘rcosfir’ in MATLAB. Use

rolloff factor = 0.125, extent of the filter = 4, oversampling rate = 4 to get the FIR filter coefficients, and set the type of the filter to ‘sqrt’.

```
>> h = rcosfir(0.125, 4, 4, 1, ‘sqrt’)
```

- You can use `stem` function in MATLAB to display filter coefficients.
- Replace pulse shaping filter coefficients in your transmitter file with the new ones.
- Make sure you can still generate a PAM signal. This time, though, eye diagram will not be perfect due to root raised cosine filtering.
- Modulate the PAM samples with a sinewave at 2 KHz. You can generate a sine wave using the look up table approach by pregenerating samples for complete period of sin wave, i.e., sin(2*pi*fc/fs*n) for n=0,1,2,3.

```
float sine_table[4]={0,1,0,-1};
```

- Again, observe the modulated PAM signal on oscilloscope using the FFT screen to see if the carrier shifts the spectrum of the baseband PAM signal. Push math button on the front panel of oscilloscope to bring the FFT screen. Select the source as the channel that is connected to PAM signal coming out of DSK. Record this screen for your lab report.

Building Ideal PAM Receiver

- Download PAM receiver (lab6R.zip) file from the course web site
- Unzip it in your myprojects directory under c6713 directory. Open the project files.
- Since the received sampling rate is set to 16 KHz for more accurate receiption, you need to redesign root raised cosine FIR filter with the function ‘rcosfir’ in MATLAB. Use the same rolloff factor = 0.125, extent of the filter = 4, oversampling rate = 8 to get the FIR filter coefficients, and set the type of the filter to ‘sqrt’.

```
>> h = rcosfir(0.125, 4, 8, 1, ‘sqrt’)
```

- PAM signal is modulated with a sinewave at 2 KHz. Since the sampling rate is increased to 16 KHz, the number of samples needed in the look up table is increased to 8 samples. Again construct your look up table by pregenerating samples for complete period of sin wave, i.e., sin(2*pi*fc/fs*n) for n=0,..7, The sine table is given in the template file. Construct the cosine table by yourself.

```
float sine_table[8]={0,0.7071,1.0000,0.7071,0.0000,-
0.7071,-1.0000,-0.7071};
```

- Demodulate the received signal with both sin and cos waves by using the sin and cosine samples in the look up table. Then, you will have two outputs. Apply root raised cosine FIR filtering to both signals.
- Outputs from both filters form a complex baseband signal. We use this approach to observe the effect of the unknown phase offset between transmitter and receiver.
- Store one (or both) of the filter output to an array (512 long) to examine in the Graphical Display of CCStudio. Inspect your demodulated baseband PAM signal on

the DSK output and compare it to the ideal baseband PAM signal. Get screenshots for your report.