

## EE/TE 4385

### Lab 7: PAM Receiver

Lab Report Due: Monday, 10/30/06, 2PM

### PAM Receiver with Carrier Tracking: C6713 DSK Implementation Using CC

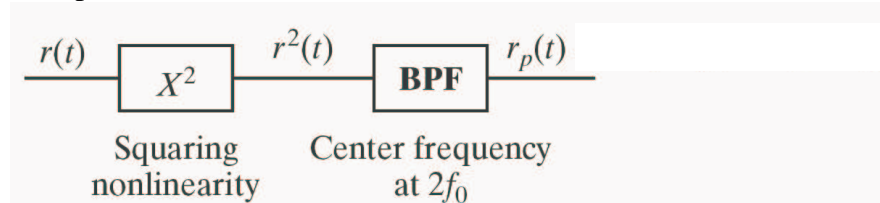
This is a follow up lab to ideal PAM receiver. This lab builds a carrier tracking function using the Phase Lock Loop (PLL). PLL scheme uses a squaring nonlinearity followed by a narrowband bandpass filter at 4KHz (since the carrier frequency is 2KHz).

- Use your project files from PAM receiver and transmitter labs. We will add additional functions to build the PLL to the PAM receiver program.
- Again, you need the help of your neighboring group to generate the PAM signal for you.
- Create a lab7 project directory under myprojects subdirectory in c:\ti folder.
- Copy your PAM receiver files into this directory. Make sure that your program still compiles without any problem. Keep the sampling rate 16 KHz.
- Ask you neighboring group to generate the PAM signal.
- Make sure your PAM receiver works as before.
- Your carrier tracking function needs two new FIR filters: BPF and LPF.
- You can use fdatool in MATLAB to generate the coefficients of these filters.
  - BPF: bandpass, equiripple, specify order:48, fs=16KHz
    - fstop1:3.5KHz, fpass1:3.8KHz, fpass2:4.2KHz, fstop2:4.5KHz
  - LPF: lowpass, equiripple, specify order:48, fs=16KHz
    - Fpass:100Hz, Fstop:500Hz
- Export the filter coefficients into your receiver code. Define them as arrays with length of 49. You can use `bpf` and `lpf` as the array names.
- You still need to demodulate the received signal with both sin and cos waves. However, this time you need to use built-in sin and cosine functions to generate sine and cosine samples to include the phase and carrier offset estimates. Be careful with incrementing the carrier phase since it will increase without bound unless it is checked against  $2\pi$ .

```
#include <math.h>
#define fo 2
#define fs 16
#define deltaf 2*PI*fo/fs
Float Phi=0;
...
Phi=Phi+deltaf;
If (Phi>=2*PI)
Phi=Phi-2*PI;

Sinsample=sin(Phi+theta1+theta2);
Cosinesample=cos(Phi+theta1+theta2);
```

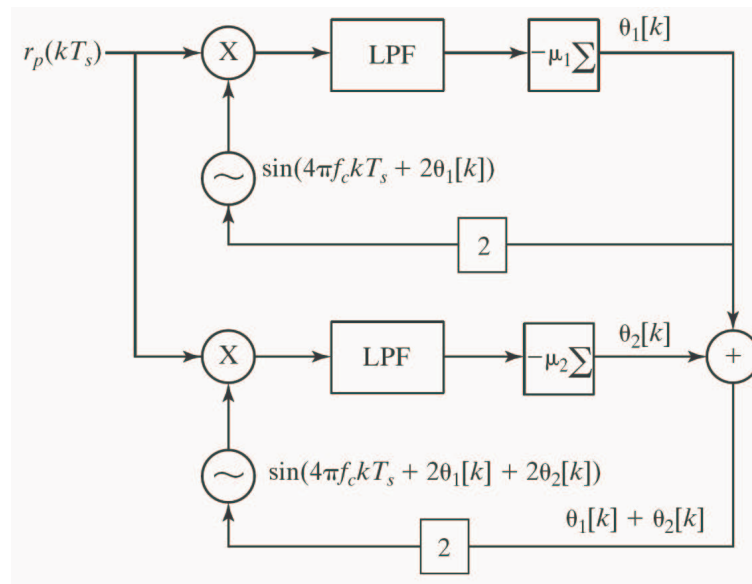
- Note that  $\theta_1$  and  $\theta_2$  will be generated by the carrier tracking loops to be explained next. Initially, they should be set to zero. Please compile your program at this point to verify that your PAM receiver still works as before without any problem. Set the time knob of oscilloscope at least to 1 sec to see that the amplitude of the signal changes at the rate of the carrier offset.
- The below is the block diagram of the first step of the carrier tracking loop. This step takes square of the input samples and applies bandpass filter to the squared input samples.



- The second step is the phase and carrier tracking loops: The first loop tracks the phase and the second loop tracks the frequency offset. Select  $\mu_1 = 0.02$  and  $\mu_2 = 0.0001$ .  $\theta_1$  and  $\theta_2$  which are obtained from these loops at each sampling instant should be used in the demodulation block of PAM receiver. Note that both  $\theta_1$  and  $\theta_2$  should also be checked against  $2\pi$  and  $-2\pi$  to keep them increasing without bound. The same applies to the original carrier phases. You can use  $\Phi$  generated above as the main phase increment due to the carrier frequency. But, you need to double it, i.e.,  $\sin(2*\Phi + \theta_1)$  or  $\sin(2*\Phi + \theta_1 + \theta_2)$ ; Output of the BPF above is used below. After mixing with both sin waves, the mixed signals are processed by the same filter separately. Update equations are given below:

$$\theta_1 = \theta_1 - \mu_1 LPF(r_p * \sin(4\pi f_c k T_s + 2\theta_1))$$

$$\theta_2 = \theta_2 - \mu_2 LPF(r_p * \sin(4\pi f_c k T_s + 2\theta_1 + 2\theta_2))$$



- Inspect your demodulated baseband PAM signal and compare it to the ideal baseband PAM signal. This time, the signals on both channels should not oscillate if your PLL works. Get a snapshot of the eye diagram on the screen. Set the time knob of oscilloscope to 1 sec to see that the amplitude of the signal remains the same after PLL locking.