**2DPSK和2FSK实验原理说明**

1. DPSK原理说明
2. 发送端：DpskSysTx.m

% mapping 0 to +1; 1 to -1

SendBpsk=1-2\*SendBpsk;

注意：SendBpsk是由SendBpsk调相（2PSK）而成，也可以写成

SendBpsk=exp(1i\*SendBpsk\*pi); 注意载波为0，故是基带调制。

经过信道之后会改变幅度、频率和相位，故收到的是复数（RecvSig = A\*exp(j\*(2\*pi\*F\*t+P)).\*SendSig（DpskSysChannel.m）

SendSig是SendBpsk经过升采样及滤波得到，但不影响运算关系）。

2、% upsampling

SendBpskUp=zeros(1,length(SendBpsk)\*UpSampleRate);

for iBits=1:length(SendBpsk)

SendBpskUp(UpSampleRate\*iBits)=SendBpsk(iBits);

end

这里：SendBpskUp(UpSampleRate\*iBits)=SendBpsk(iBits)相当于基带信号：



下图SendBpskUp前300点数据打印的图形 

*Ts*在这里就是UpSampleRate=20，是用采样的点数表示的。真实值应该是UpSampleRate/Fs或=1/Rs

3、 % RRC filtering

filterDef=fdesign.pulseshaping(UpSampleRate,'Square Root Raised Cosine','Nsym,Beta',FilterSymbolLen,Rolloff);

myFilter = design(filterDef);

myFilter.Numerator=myFilter.Numerator\*UpSampleRate;

SendSig = conv(myFilter.Numerator,SendBpskUp);

这里：SendSig 就是基带信号：



其前300点图形如下：



注意：基带波形用的是根升余弦，波形由myFilter.Numerator给出，其图形为：



这种波形主瓣就跨了两个码元周期，故波形是重叠的。

另外滤波器的长度为120，滤波（conv）后引入了120/2的延时，加上SendBpskUp的数据是从20（可以从一开始）开始的，故第一个采样点在80。

这些引入的延时不用管它，在接收端通过搜索前导符可以去掉。

1. 根升余弦波形是对称的，根据匹配滤波器原理，匹配滤波和发送滤波器是一样的。
2. SendSig是由SendBpsk升采样后滤波得到，每个码元周期幅度变化了，但相位没有变，和1）中的描述是一样的。
3. SendSig经过信道传输后，只增加频偏和相偏，没有载波，故仍然是基带信号，见1），在本实验中，这些偏差不用纠正。
4. 前导符的搜索算法参考2BPSK代码。
5. 由于SendBpsk=exp(1i\*SendBpsk\*pi)，差分想干解调就是前一个码元的共轭乘以当前码元（或者相反）。

1. 2FSK原理说明
2. 2FSK的调制是两路2ASK信号的叠加，其原理是。其中：f0和f1分别是0和1的频率，但这个频率是相对于载波的相对频率（即去除载波），比如可选Rs和-Rs或其倍数。由于没有载波，故s(t)是基带信号。
3. 1）中的调制可参考信道中的代码。
4. 指数调制的解调只要乘以载波的复共轭就可以。