

Software Radio Lab 3: PSK & Differential Encoding
4/15/07

This lab will continue the simulation trend by extending to Phase Shift Keying, and differential encoding of data. Phase Shift Keying is prominent in many real radios and even optical communications links. If you have an old cellular phone that uses TDMA, it uses 4-PSK. Also, variants of DBPSK & DQPSK are used in 802.11b, Bluetooth and ZigBee.

NOTE: We will be using almost the same signal flow graph as Lab 2, with only the addition of the Differential Encoder/Decoder. If you saved your signal flow graph from last time (or you found someone else's on the computer), then you can save yourself 5 or so minutes of clicking.

PreLab: Review Section 5.2.8 in the textbook regarding DPSK.

1. Repeat steps 1 through 7 of Lab 2.
2. Add the "Differential Encoder" block located in the Coders category. The "Modulus" is equal to the number of symbols or $2^{\text{(number of bits/symbol)}}$.

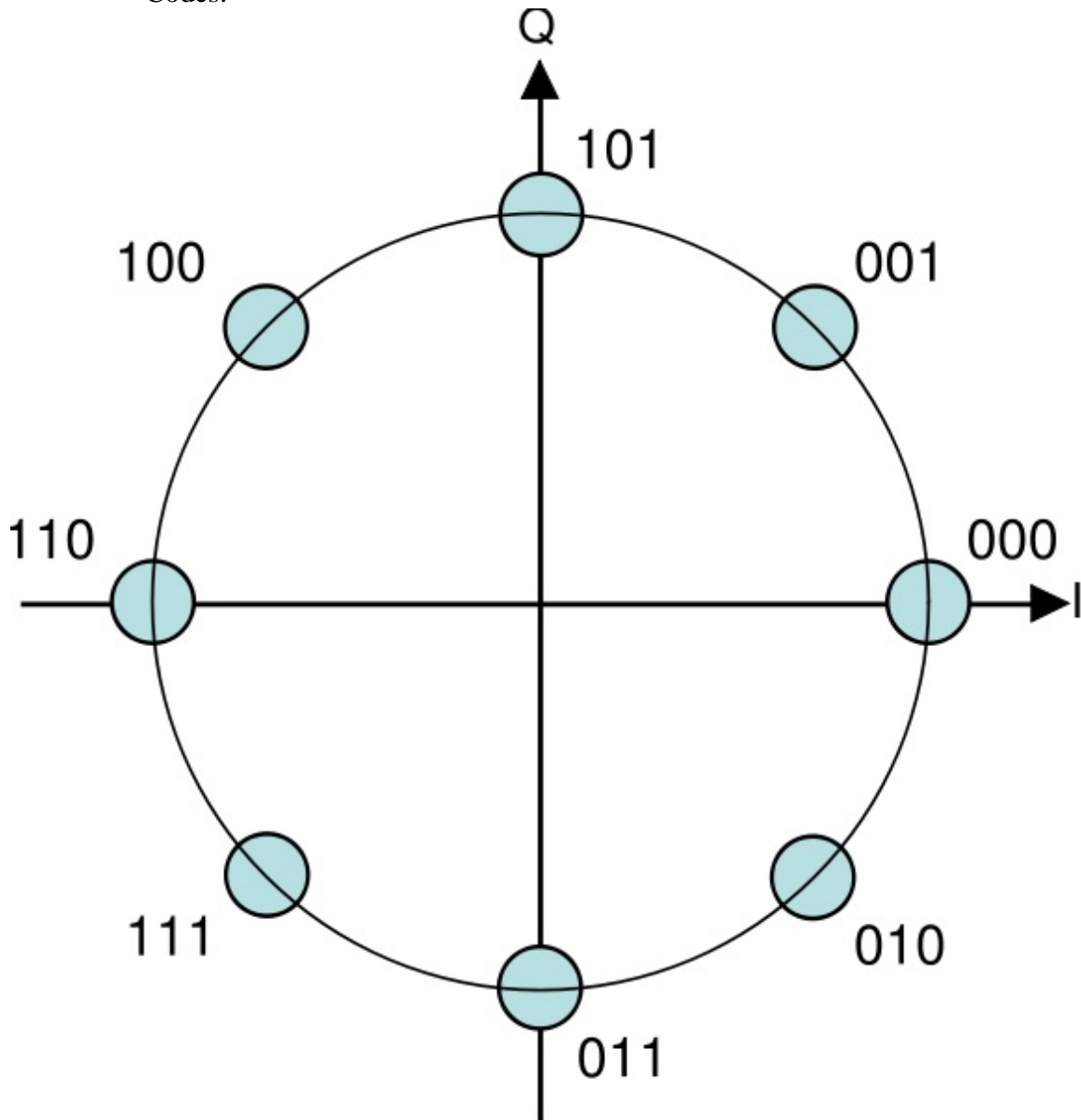
Differential Encoder:

In order to send the difference of two symbols, this block keeps a record of the previously transmitted symbol. These two symbols are stored as integers, and then are simply added and a modulus is taken. The decoder reconstructs the symbol by also keeping a record of the previous symbol. In a perfect world, both the encoder and decoder agree on initial value of the memory element to be 0. GNURadio is far from perfect, and to make things simple the decoder does not explicitly set an initial value. This frequently results in the first symbol received being in error. So if you want to be REALLY accurate with your data, you should always subtract one from the total errors reported.

3. Repeat Step 8 in Lab 2 to get 10-20 points at various noise amplitudes and compare this with the theoretical BER for DBPSK (check online or in the textbook for the formula or derive it if you feel up to it). In your report, justify any discrepancy between the BER plots. Also, plot the BER for B-PAM from last week's lab if you think there is an interesting relationship between the two transmission schemes.
(Hint: plot B-PAM, DBPSK, & theoretical DBPSK on the same graph).
4. Now we will implement the famous 8-DPSK (Bluetooth uses this). Change the parameters in the "Constellation Decoder", "Chunks to Symbols", "Differential Encoder", and "Differential Decoder" to support the following constellation:
 $(1+0j), (0.707+0.707j), (0+1j), (-0.707+0.707j),$
 $(-1+0j), (-0.707-0.707j), (0-1j), (0.707-0.707j)$
5. Repeat Step 8 in Lab 2 to get 10-20 points and plot the BER for 8-DPSK. Include a theoretical plot (or your data for 8-QAM from last week) of 16-

QAM on the same graph. If given the choice, which scheme would you choose?

- By changing the order of values in the “Symbol Table” parameter, you can map different binary values to different symbols (you will need to change the “Constellation Decoder” too). Modify the encoding scheme to use Gray Codes.



- As you might have guessed, 8-DPSK isn't used very much. Now we will look at DQPSK, which is very widely used, and trivial to make. Use the following constellation:
 $(1+0j)$, $(0-1j)$, $(-1+0j)$, $(0+1j)$
- Plot the BER for DQPSK, and compare it to QPSK. Are you surprised by the results? Differential Encoding of bits introduces a lot of headaches, especially at higher constellations. Why do you think 256-QAM is used in Cable Modems vs 256-DPSK?

Questions:

1. In a DBPSK, assume that the first few bits received are in error. If the probability of subsequent bit errors is 0, how many bits will be received in error if N bits are transmitted?
2. If PSK is modulated with a carrier and subsequently demodulated, what is a source of error that is not caused by the channel? (Assuming AWGN channel)
3. This lab is still “cheating” in regard to one of the major problems with communication systems. Clock Recovery, or a lack thereof, can quickly kill even the most robust systems. However, DPSK & DQPSK aren't that effected if you don't sample exactly at $t=T_b$, and that is primarily why they are used. Justify this statement with words and/or drawings.