Tasks

It is suggested that you use the Communications Toolbox (which provides various MATLAB® functions for the design and analysis of communication systems) for carrying out this task. You may find all or some of the following MATLAB functions useful: randi, comm.GeneralQAMModulator, comm.GeneralQAMDemodulator, awgn, reshape, bi2de, de2bi, comm.RSEncoder, comm.RSDecoder, biterr. A detailed explanation of how to call and use each of these functions is given in ref. [2]. A shorter explanation may also be obtained by typing help followed by function name in the MATLAB command window.

Figure 1 shows the 16-APSK constellation with 3 different arrangements of transmitted carrier states on two rings with a ratio of 2.1756 between outer and inner rings radii. You are to write a MATLAB® program (m-file) to implement each of these M-ary APSK transmissions (where M = 16) through an AWGN channel, including modulation (symbol generation) at the transmitter, addition of white noise in the channel, and symbol detection and determination of bit error ratio (BER) at the receiver. Your message signal will be a random sequence consisting of **403200** integers that have values between 0 and M−1 with equal probability. For each value of E_b/N_o in the range 4 dB to 16 dB in steps of 0.5 dB, where E_b is the average energy per bit and N_o is the noise power per unit bandwidth, run your simulation of the system repeatedly until at least 101 bit errors occur before calculating BER.

Submit a report of your work, which should include the following:

(a) A printout of your m-files. [10 marks]

(b) A graph of BER versus E_b/N_o (dB) featuring three BER curves, one for the simulation results obtained above for each constellation in Fig. 1. Use a

logarithmic scale for the y-axis and limit its range to 1×10^{-8} to 0.1, and a linear scale for the x-axis with range 4 to 16 dB. [20 marks]

(c) A discussion of your simulation results. Discuss how BER varies with E_b/N_o and give reasons for any observed differences in performance of the 3 constellations. Furthermore, explain how each of the three systems compares with the stipulation of Shannon's information capacity theorem. [25 marks]

(d) If Figure 1 is the constellation diagram of a transmission system with

$$\alpha_0(t) = 4000\sqrt{2}\cos\left(8\pi \times 10^9 t\right)\text{rect}\left(\frac{t - 3.125\times10^{-8}}{6.25\times10^{-8}}\right);$$

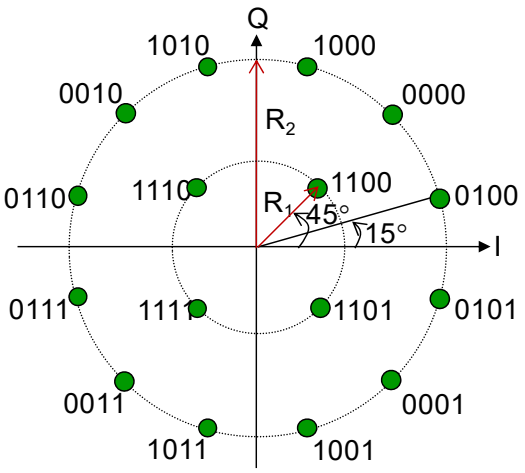
$$\alpha_1(t) = -4000\sqrt{2}\sin\left(8\pi \times 10^9 t\right)\text{rect}\left(\frac{t - 3.125\times10^{-8}}{6.25\times10^{-8}}\right);$$

$$R_1 = 10^{-3} \ \ (\text{joules})^{1/2}$$

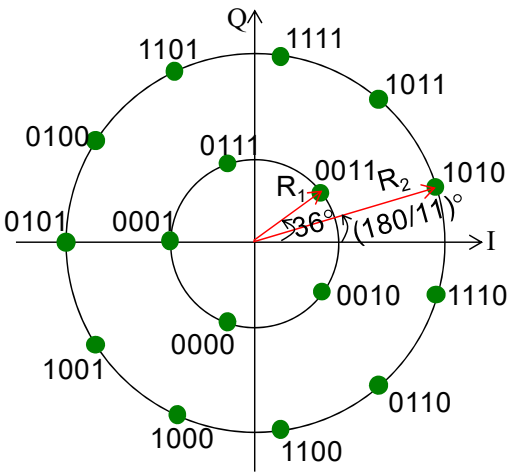
and the symbols are filtered prior to transmission using a raised cosine filter of roll-off factor α = 0.25, calculate the average power in the transmitted carrier of each system and the frequency range occupied by the transmitted signal.

[20 marks]

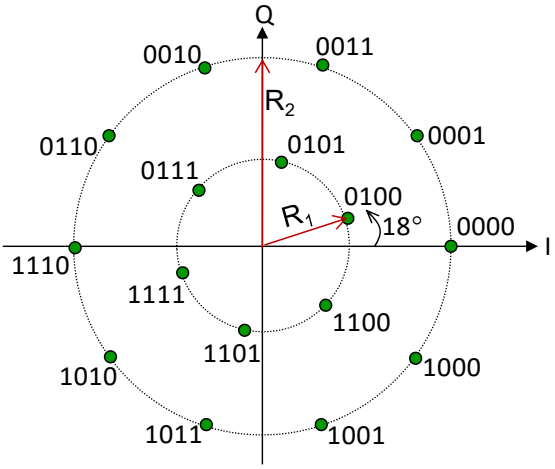
(e) You may reduce the value of E_b/N_o required to achieve a specified BER in the simulated system of Fig. 1 by introducing error correction coding. Investigate this possibility for the constellation of <u>Fig. 1(b) only</u> using a Reed-Solomon (RS) block code (n, k) = (255, 223). Determine (by simulation following the guidance given in the appendix after Fig. 1) the bit error ratio of the coded system for values of E_b/N_o between 4 dB and 10.5 dB in steps of 0.5 dB. Include in your report your m-file for this section and a graph of BER versus E_b/N_o featuring the simulated BER curves for the coded system (in this section) and the uncoded 5+11-APSK system (obtained earlier). By comparing the two simulation results, what is your estimate of coding gain? Does coding gain depend on bit error ratio? Explain fully. [25 marks]



(a) 4+12-APSK: $R_2 = 2.1756R_1$; $R_1 \equiv 1$ (normalised)



(b) 5+11-APSK: $R_2 = 2.1756R_1$; $R_1 \equiv 1$ (normalised)



(c) 6+10-APSK: $R_2 = 2.1756R_1$; $R_1 \equiv 1$ (normalised)

Figure 1: 16-APSK constellations featuring different arrangements of the 16 states on 2 rings

Appendix: Guidance for RS Coding simulation in Task (e)

1. Since the modulator input must be integers in the range 0 to M−1 whereas your RS encoder will produce coded integers outside this range, you must use a random sequence of bits (i.e. 0 and 1) as the message.
2. Set the message length N to **214080**, which is carefully chosen to be a multiple of k, $\log_2(n+1)$ and $\log_2 M$. Set the 'BitInput' parameter of the RS encoder and RS Decoder to the MATLAB keyword true. This tells the coder that the input is bits, rather than integers (the default setting). The coder will therefore take k bits at a time from the stream of N bits and produce n coded bits, so that the message of length N bits produces n*N/k bits at the encoder output.
3. Before presenting the coded message to the modulator, you need to convert the coded bit stream to integers by taking $\log_2 M$ bits at a time and replacing with their integer value. You may find the MATLAB functions reshape and bi2de useful. Use help to learn how these functions work and then, to further deepen your understanding, try out the following from within the MATLAB command window (i.e. by typing the following and then pressing the enter key) to see how it converts the bitstream to integers in the range 0 to M-1:

M=8, bitstream=[1,0,1,0,1,1,0,0,1,1,1,1,0,0,0,1,1,0,1,0,1,1,0,0],
bitstream = reshape(bitstream,log2(M),length(bitstream)/log2(M))',
integerstream = bi2de(bitstream,'left-msb')

4. You will also need to convert the detected symbols (which are integers from 0 to M−1) at the receiver back to bits before presenting to the RS decoder (since this decoder is expecting a stream of bits, rather than integers). To do this you will find the MATLAB function de2bi useful. Now follow up the exercise in (3) above by trying the following:
   bits = de2bi(integerstream,log2(M),'left-msb')', restoredbitstream = bits(:)
5. The RS decoder output is a bitstream of length N. This is your recovered message which you compare to the originally generated message to compute bit error ratio.