CAIRO UNIVERSITY
FACULTY OF ENGINEERING
ELECTRONICS & ELECT. COMM. DEPT

COMMUNICATIONS
FOURTH YEAR
FIRST SEMESTER 2021-2022

INFORMATION THEORY REPORT

- This report can be done in groups with a maximum number of 5 students per group.
- Matlab, or similar programs, is to be used in this report.

Source Coding

1. Write a program to find the binary Huffman code of an independent discrete random variable. Test your code on the case when there is a source that produces 7 symbols with the following probabilities: 0.35, 0.30, 0.20, 0.10, 0.04, 0.005, and 0.005.

2. Write a program to find the binary Fano code of an independent discrete random variable. Test your code on the case when there is a source that produces 7 symbols with the following probabilities: 0.35, 0.30, 0.20, 0.10, 0.04, 0.005, and 0.005.

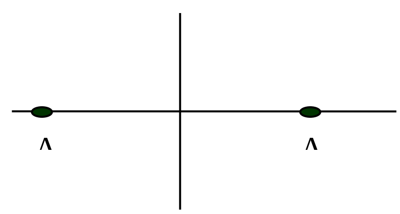
Channel Coding

Assume you want to transmit 110000 information bits.

3. Use BPSK with no coding and plot a graph showing probability of error versus Eb/No for Eb/No=-3 to 10 dB where Eb=A^2.
   Note: Use

   segma = sqrt((Eb/2)./10^(ebno/10));

   where segma is noise power per dimension, Eb is energy per bit and ebno is Eb/No.



BPSK Constellation

4. Repeat 1, but use BPSK with repetition 3 coding, and use hard decision decoding:
   (a) With same energy per transmitted bit.
   (b) With same energy per information bit.

Plot each of the curves of (a) and (b) with the curve of 1 on the same graph.

If you are designing a communication system, can you think of a scenario where you will use such an error correction coding scheme as in (a) or (b) ?

---

5. Repeat 2, but use soft decision decoding. Again, think of a scenario where you will use such an error correction coding scheme.

---

6. Repeat 2 but use BPSK with a (7,4) Hamming code. You can use Matlab “encode” and “decode” functions.
   (c) What is the minimum distance of this Hamming code?
   (d) Would you recommend such a code if the purpose is to decrease the bit error rate using the same energy, but disregarding transmit time?

---

7. Repeat 2 but use BPSK with a (15,11) Hamming code.
   (e) Would you recommend such a code if the purpose is to decrease the bit error rate using the same energy, but disregarding transmit time?
   (f) If you want to keep transmission time equal to or less than the transmission time of 1, what would you propose?
   (g) Plot a curve for the performance of your proposal in (b). Comment.

---

8. Now, assume you want to transmit 26200000 bit. Compare, on the same curve, at Eb/No= 5 to 15 dB, between
   (h) Using QPSK with no coding.
   (i) Using 16 QAM but utilizing a (255,131) BCH code. Use Matlab “encode” and “decode” functions. Also, you can use the following functions for 16 QAM modulation and demodulation.

<u>Modulation</u>

```
function [rxsig]=mod16(txbits)
psk16mod=[1+j*1 3+j*1 1+j*3 3+j*3 1-j*1 3-j*1 1-j*3 3-j*3 -1+j*1 -3+j*1
-1+j*3 -3+j*3 -1-j*1 -3-j*1 -1-j*3 -3-j*3];
sigham=txbits;
m=4;
sigqam16=reshape(sigham,m,length(sigham)/m);
rxsig=(psk16mod(bi2de(sigqam16')+1));
```

<u>Demodulation:</u>

```
function [rxbits]=demod16(rxsig)
m=4;
psk16demod=[15 14 6 7 13 12 4 5 9 8 0 1 11 10 2 3];
```

```
rxsig(find(real(rxsig)>3))=3+j*imag(rxsig(find(real(rxsig)>3)));
rxsig(find(imag(rxsig)>3))=real(rxsig(find(imag(rxsig)>3)))+j*3;
rxsig(find(real(rxsig)<-3))=-3+j*imag(rxsig(find(real(rxsig)<-3)));
rxsig(find(imag(rxsig)<-3))=real(rxsig(find(imag(rxsig)<-3)))-j*3;
rxdemod=round(real((rxsig+3+j*3)/2))+j*round(imag((rxsig+3+j*3)/2));
rxdebi=real(rxdemod)+4*(imag(rxdemod));
sigbits=de2bi(psk16demod(rxdebi+1));
rxbits=reshape(sigbits',1,length(sigbits)*m);
```

Notes:

(i) Use Eb=2.5.

(ii) Simulation time: To speed up simulation time, at EbNo=5 to 8 you can only do 2620000 bits. You can, for example, do a loop 100 times for 26200 bits each, or whatever is faster on your computer. For EbNo=9 you will have to do 26200000 bits at least. For EbNo=10 to 15 you can assume linear interpolation between 8 and 15. Plot the curve from 5 to 15 dB.

---

9. Write a Matlab code to convolutionally encode 1000 bits by the (2,3,K) code described by:

$g_1^{(1)}$=[0 1]
$g_2^{(1)}$=[1 1]
$g_1^{(2)}$=[1 1]
$g_2^{(2)}$=[1 0]
$g_1^{(3)}$=[0 0]
$g_2^{(3)}$=[1 1]
