Αναφορά 2ης Εργασίας για το μάθημα Ψηφιακές Τηλεπικοινωνίες

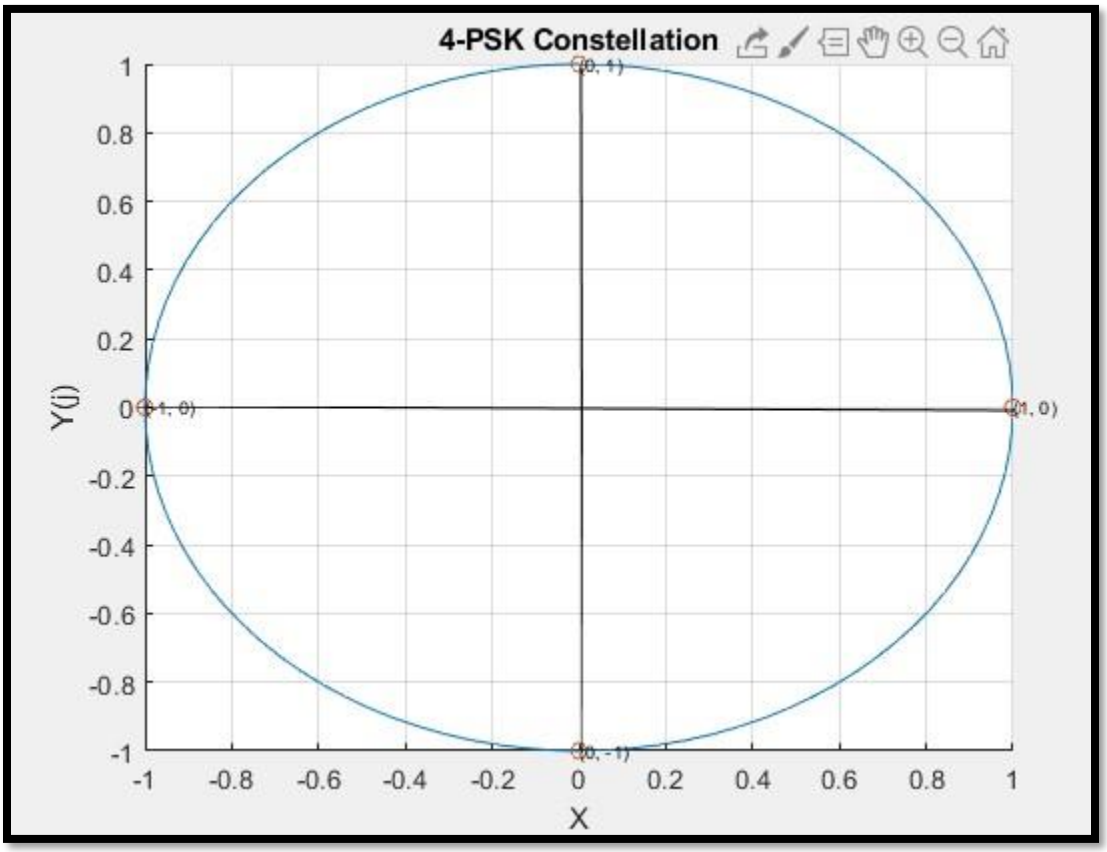
Στοιχεία Φοιτητή:

Ονοματεπώνυμο: Δημήτριος Κωστορρίζος

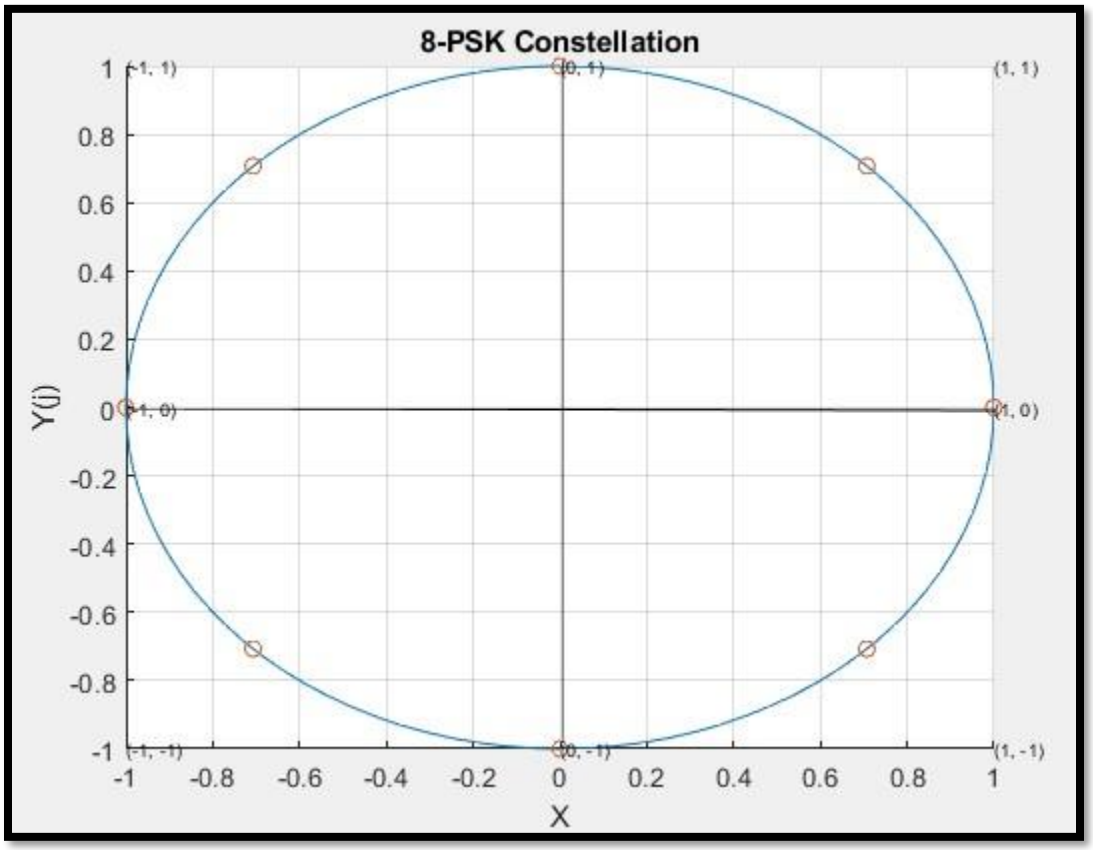
ΑΜ: 1054419

Έτος Σπουδών: Δ’

4-PSK Constellation:

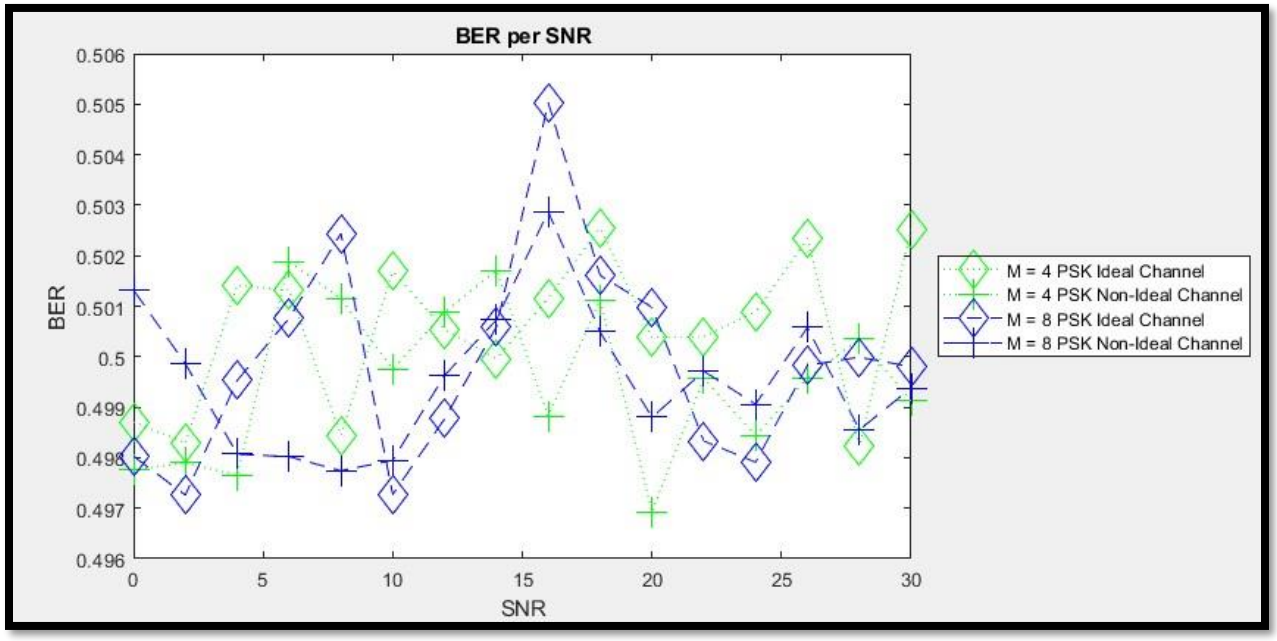


8-PSK Constellation:

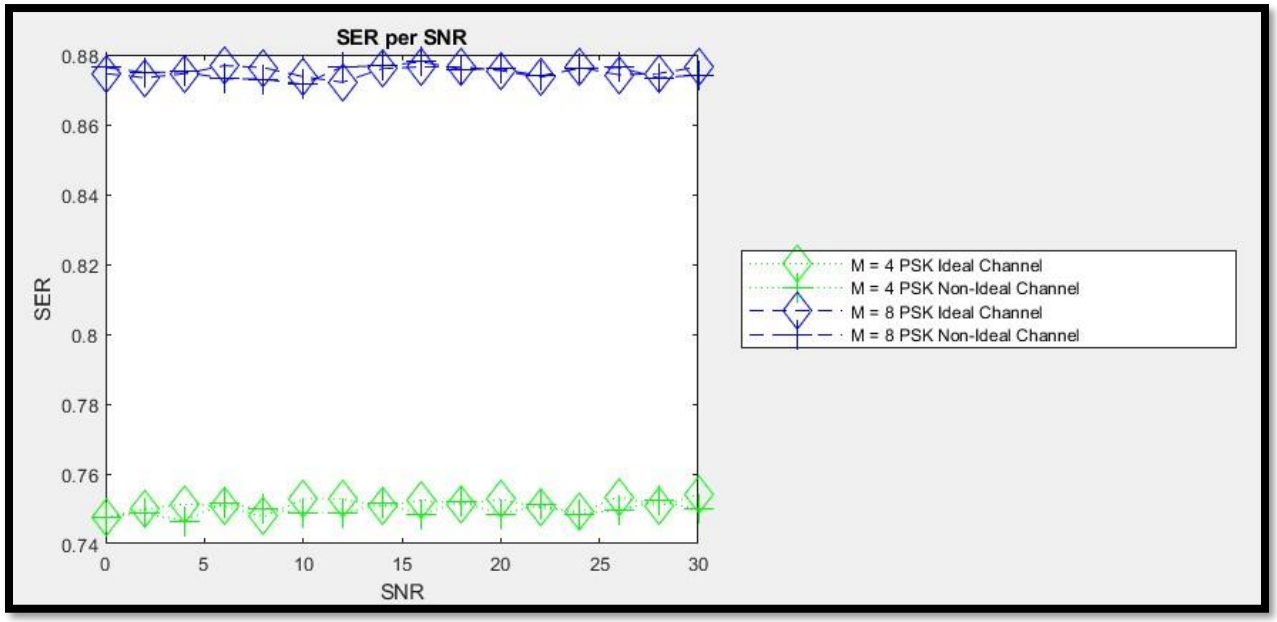


Οι παρακάτω καμπύλες BER, SER καθώς και οι πιθανότητες σφάλματος ανά SNR, έχουν υπολογιστεί για δεκαδική κωδικοποίηση των συμβόλων και μέγεθος ακολουθίας bit, N = 100000. Οι πιθανότητες σφάλματος ανά SNR έχουν στρογγυλοποιηθεί στα 4 σημαντικά ψηφία.

Καμπύλη BER:



Καμπύλη SER:



Πιθανότητες Σφάλματος ανά τιμή του SNR ε [0,30], για τις μεθόδους 4-PSK, 8-PSK σε ιδανικό και μη ιδανικό κανάλι.

SNR	BER				SER			
	4-PSK Ideal	4-PSK Non-Ideal	8-PSK Ideal	8-PSK Non-Ideal	4-PSK Ideal	4-PSK Non-Ideal	8-PSK Ideal	8-PSK Non-Ideal
0	0.5011	0.4989	0.5001	0.5019	0.7517	0.7496	0.8766	0.8759
2	0.4992	0.5044	0.4994	0.4999	0.7477	0.7538	0.8748	0.8743
4	0.4998	0.5010	0.4995	0.4986	0.7496	0.7518	0.8739	0.8761
6	0.5016	0.4994	0.4997	0.5035	0.7511	0.7469	0.8743	0.8769
8	0.4984	0.4989	0.5015	0.5025	0.7484	0.7490	0.8764	0.8766
10	0.4980	0.4980	0.4987	0.4996	0.7478	0.7470	0.8757	0.8770
12	0.4988	0.4988	0.5009	0.5003	0.7479	0.7488	0.8723	0.8741
14	0.5001	0.4984	0.5013	0.5000	0.7494	0.7493	0.8780	0.8749
16	0.5006	0.5013	0.5025	0.5012	0.7502	0.7513	0.8787	0.8772
18	0.4997	0.4980	0.5022	0.4988	0.7487	0.7477	0.8773	0.8745
20	0.5029	0.5007	0.5007	0.4984	0.7533	0.7513	0.8745	0.8754
22	0.4992	0.4995	0.5008	0.4997	0.7478	0.7480	0.8707	0.8781
24	0.5003	0.5039	0.4997	0.4997	0.7500	0.7541	0.8752	0.8743
26	0.5010	0.4990	0.4973	0.4990	0.7509	0.7493	0.8734	0.8749
28	0.4989	0.5017	0.4994	0.4986	0.7479	0.7542	0.8730	0.8722
30	0.4980	0.4983	0.5011	0.5013	0.7472	0.7473	0.8749	0.8774

Με βάση τις παραπάνω πιθανότητες, σε συνδυασμό με τα γραφήματα των καμπυλών BER, SER για την μετάδοση 4-PSK, 8-PSK σε ιδανικό και μη ιδανικό κανάλι, καταλήγω στα εξής συμπεράσματα:

- Στην περίπτωση όπου στοχεύουμε στην ελαχιστοποίηση του δείκτη BER, η καλύτερη επιλογή είναι η μέθοδος 4-PSK σε ιδανικό κανάλι μετάδοσης.

- Στην περίπτωση όπου στοχεύουμε στην ελαχιστοποίηση του δείκτη SER, η καλύτερη επιλογή είναι η μέθοδος 4-PSK σε μη ιδανικό κανάλι μετάδοσης.

- Στην περίπτωση όπου στοχεύουμε στην ταυτόχρονη ελαχιστοποίηση των δεκτών BER και SER, η καλύτερη επιλογή είναι η μέθοδος 4-PSK σε μη ιδανικό κανάλι μετάδοσης.

Κώδικας M-PSK συστήματος μετάδοσης σε δύο είδη καναλιού:

```
function [OutputBitMatrix, OutputSymbolVector, InputSymbolVector, InputBitMatrix] = PSK(N,M,ChannelType,SNR)
  %Filter
  RollOffFactor = 0.3;
  HyperSamplingFactor = 4;
  HyperSamplingPeriods = 6;
  Filter = rcosdesign(RollOffFactor,HyperSamplingPeriods,HyperSamplingFactor); % Hyper sampled filter

  %Bit Sequence Creation
  SymbolLength = log2(M);
  NumberOfSymbols = ceil(N/SymbolLength);
  BitSequence = randi([0 1],1,NumberOfSymbols * SymbolLength);
  BitMatrix = zeros(NumberOfSymbols, SymbolLength); %Symbol Matrix, every row has bit representation of the symbol

  VectorIndex = 1;
  for index = 1:NumberOfSymbols % Convert the Bit Sequence to Bit Matrix
    for counter = 1:SymbolLength
      BitMatrix(index, counter) = BitSequence(VectorIndex);
      VectorIndex = VectorIndex + 1;
    end
  end
  InputBitMatrix = BitMatrix; % Every row has the bit representation of each symbol

  SymbolVector = bi2de(BitMatrix,"left-msb"); % Encode the Bit Matrix to Decimal Symbol Vector
  InputSymbolVector = SymbolVector';

  HyperSampledSymbolVector = upsample(SymbolVector,HyperSamplingFactor); %Hyper Sampled Symbol Vector

  PhaseVector = zeros(1, M); %Symbol Phase Vector
  AmplitudeVector = zeros(1, M); %Symbol Amplitude Vector
  for index = 1:length(AmplitudeVector)
    PhaseVector(index) = (2 * pi * (index - 1)) / M;
    AmplitudeVector(index) = complex(cos(PhaseVector(index)), sin(PhaseVector(index))); % Generate the complex symbol
  end

  ModulatedSymbols = zeros(1,length(HyperSampledSymbolVector));
  for index = 1: length(HyperSampledSymbolVector)
    ModulatedSymbols(index) = AmplitudeVector(HyperSampledSymbolVector(index) + 1);
  end



  FilteredModulatedSymbols = filter(Filter,1,ModulatedSymbols); %Apply Sender Filter

  %Channel
  if(ChannelType == "non-ideal")
    h = [0.04, -0.05, 0.07, -0.21, -0.5, 0.72, 0.36, 0, 0.210, 0.03, 0.07];
    HyperSampledh = upsample(h,HyperSamplingFactor);  % Hyper sample the Channel's Impulse
    FilteredModulatedSymbols = filter(HyperSampledh,1,FilteredModulatedSymbols); % Apply Channel's Impulse
  else
    if (ChannelType ~= "ideal")
      error("Channel Type has to be either ideal or non-ideal");
    end
  end
```

```
    %AWGN
    SignalPower = sum(abs(FilteredModulatedSymbols).^2)/length(FilteredModulatedSymbols); % Signal power
    Variance = SignalPower/(10 ^ (SNR/10)); % Noise variance
    AWGNoise = sqrt(Variance) * (randn(size(FilteredModulatedSymbols)) + 1j*randn(size(FilteredModulatedSymbols))); % Real
and Imaginary Noise Generation
    FilteredModulatedSymbols = FilteredModulatedSymbols + AWGNoise; % Add the complex noise to the modulated symbols

    FilteredModulatedSymbols = filter(Filter,1,FilteredModulatedSymbols);%Apply Receiver Filter

    FilteredModulatedSymbols = downsample(FilteredModulatedSymbols, HyperSamplingFactor); %Downsample the received
symbols

    % Euclidean Distance
    ReceivedSymbolVector = zeros(1,length(FilteredModulatedSymbols));
    EuclideanDistance = zeros(1,M);
    for counter = 1:length(FilteredModulatedSymbols)
        for index = 1:M
            SymbolDistanceVector = [real(FilteredModulatedSymbols(counter)) - real(AmplitudeVector(index)),
imag(FilteredModulatedSymbols(counter)) - imag(AmplitudeVector(index))];
            EuclideanDistance(index) = norm(SymbolDistanceVector); % Euclidean norm for each symbol
        end
        [~,MinimumPosition] = min(EuclideanDistance); % Find which symbol has the minimum distance from the received symbol
        ReceivedSymbolVector(counter) = MinimumPosition - 1; %Most Similar Symbol
    end

    OutputSymbolVector = ReceivedSymbolVector;
    OutputBitMatrix = de2bi(ReceivedSymbolVector', "left-msb"); % Decode the received symbols to bit
end
```