EEE431 - Digital Communications

MATLAB Assignment 2 Report

Efe Tarhan 22002840

Introduction and Outline

In the digital world data transmission still requires the mediation of analog devices and channels. Due to the necessity of using analog waveforms to transmit the data, digital modulation is one of the important concepts of digital communication systems. Modulating a digital signal using an analog waveform requires the knowledge of signal spaces, orthogonal projection, best approximation of a signal to a given space and modulation techniques like PAM, PSM, FSM, etc.

In this homework report, the first part covers finding a proper set of orthonormal basis functions that can be used for representing the given signal. Then the proper coefficients for each basis function have been found using the projections. After that, the process has been repeated using noisy input signals and coefficients have been obtained again. In the second part of the homework, a random binary sequence with length 1e5 has been sent using pulse amplitude modulation and the error probability has been observed for the received sequence.

Part I: Signal Spaces

a.

Simple sinusoidals are orthogonal signals where they can be used for creating a basis to represent a signal space. The Discrete Cosine Transform (DCT) exactly uses this principle. Proof of the orthogonality of different sinusoidal signals is given below.

$$Let\ x_i(t) = \sin(2\pi i t)\ \ then,$$

$$\langle x_k(t), x_l(t)\rangle = \int_0^1 \sin\left(2\pi \frac{k}{T} t\right) \cdot \sin\left(2\pi \frac{l}{T} t\right) dt$$

$$= \frac{1}{2}\int_0^1 \cos\left(2\pi \frac{k-l}{T} t\right) dt - \frac{1}{2}\int_0^1 \cos\left(2\pi \frac{k+l}{T} t\right) dt$$

$$= \begin{cases} 0.5, & k = l \\ 0, & k \neq l \end{cases}$$

Therefore, for this task the following set can be used as an orthonormal basis for the given signal:

$$\boxed{\boldsymbol{S = \left\{\phi_i(t)\ |\ \phi_i(t) = \sqrt{2}\sin(10\pi i t)\,, i = 1, 2, 3 \ldots 30\right\}}}$$

To ensure that all signals in the following form can be represented with this basis each sinusoidal for each possible f_k has been included in the set.

$$x(t) = \sum_{k=1}^{10} a_k \sin(2\pi f_k t)$$

$$where\ f_k \in \{30k - 20, 30k - 10, 30k\}$$

Therefore, it can be said that the orthonormal basis is **30 dimensional**.

b.

The values of f_k and a_k can be extracted from the given signal using the following technique. Determining the values of f_k is a natural cause of finding the proper values for a_k. After finding the constellation coefficients of the signal the f_k values are the frequencies of the signals with nonzero a_k. The algorithm can be seen below.

- Obtain $\Phi_i = \langle x(t), \Phi_i(t) \rangle,\ i = 1,2,3 \dots 30$

Selecting the nonzero Φ_i can be changed with selecting the maximum of the absolute values of $\Phi_k, \Phi_{k+1}, \Phi_{k+2}$ for each k. Since the signal $x(t)$ is already in a signal reconstruction form from its basis functions the values of a_k can be found in the following way.

- $a_k = \sqrt{2}(\max(\{ |\Phi_k|, |\Phi_{k+1}|, |\Phi_{k+2}|\}))$

Then select the f_k such that,

- $f_k = \{30k - 20,\ 30k - 10, 30k\}$ which corresponds to the $a_k s$

c.

This method has been implemented in MATLAB using the file “signaldata0.mat”, since the signal in the .mat file is the sampled version of the original $x(t)$, the integral operation has changed with the following summation operation for calculating the inner products. The sampling frequency $f_s = 1200\ Hz$ which is nearly the 4 times of the maximum possible frequency in the signal $x(t)$.

$$\Phi_i = \int_0^1 x(t) \cdot \Phi_i(t)dt \cong \frac{1}{1200}\sum_{n=0}^{1199} x[n] \cdot \Phi_i[n]$$

The obtained Φ_i's can be seen below in Figure 1.

```
phi_list =

  Columns 1 through 7

         0    4.2426         0    4.2426         0         0         0

  Columns 8 through 14

    6.3640         0    4.2426         0         0         0    2.8284

  Columns 15 through 21

         0    0.7071         0         0         0         0    4.2426

  Columns 22 through 28

         0         0    2.1213         0         0    4.2426    2.1213

  Columns 29 through 30

         0         0
```

Fig.1 The values of obtained Φ_i's on the command window of MATLAB.

Then the values of the a_ks determined by selecting nonzero coefficients in each triple can be seen below in Figure 2.

```
ak_list =

  Columns 1 through 9

    6.0000    6.0000    9.0000    6.0000    4.0000    1.0000    6.0000    3.0000    6.0000

  Column 10

    3.0000
```

Fig.2 The values of obtained a_ks on the command window of MATLAB.

Then the frequencies that correspond to the determined a_ks can be seen below in Figure 3.

```
fk_list =

    10    60    80   100   150   180   190   220   270   280
```

Fig.3 The values of obtained f_ks on the command window of MATLAB.

This concludes the extraction of the values of a_k and f_k from the given signal $x(t)$.

d.

In this part 3 new signals have been generated by adding white Gaussian noise on top of the uncorrupted signal $x(t)$. The generated signals can be represented mathematically as the following:

$$\tilde{x}(t) = x(t) + \mathcal{N}(0, \sigma^2) \text{ for } \sigma = \{1,5,10\}$$

The resultant signals can be seen in Figures 4, 5, and 6.

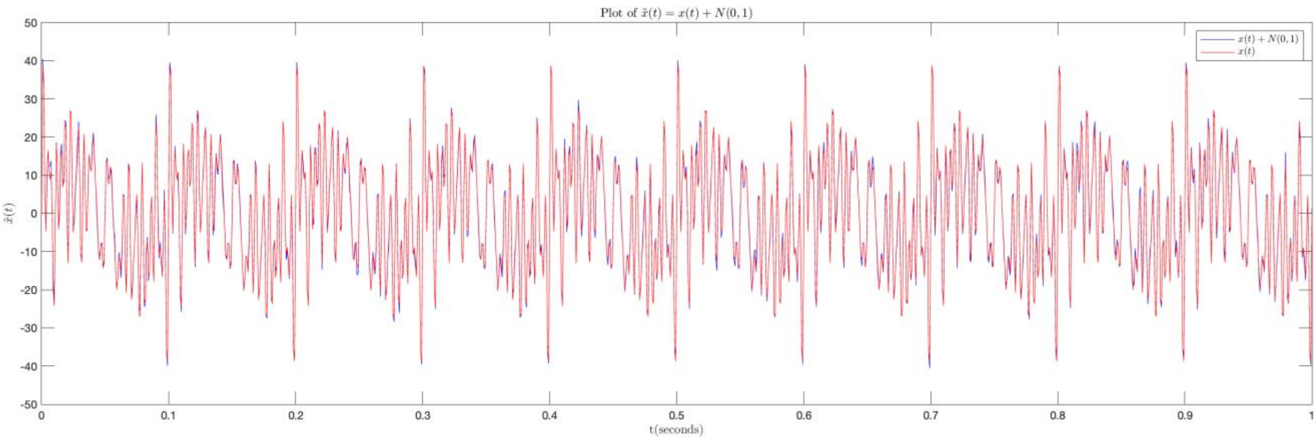

Fig.4 Plots of the signals $x(t)$ (red) and $\tilde{x}(t) = x(t) + \mathcal{N}(0,1)$ $(blue)$.

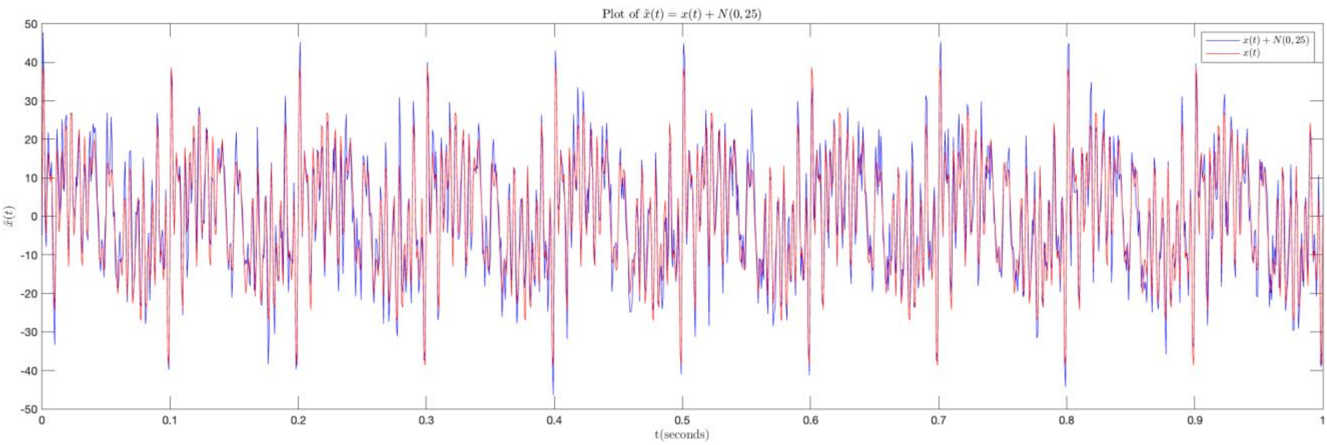

Fig.5 Plots of the signals $x(t)$ (red) and $\tilde{x}(t) = x(t) + \mathcal{N}(0,25)$ $(blue)$.

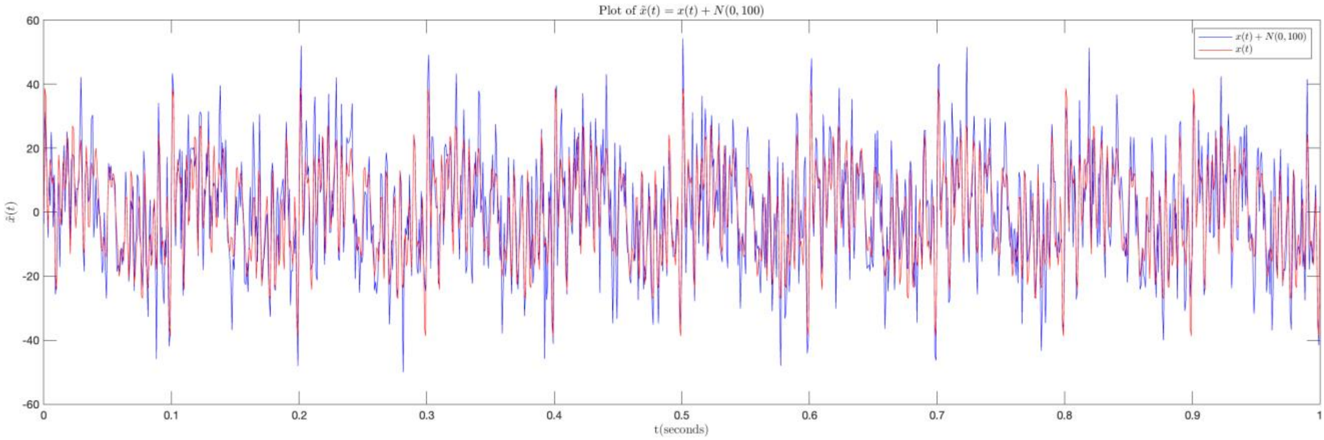

Fig.6 Plots of the signals $x(t)$ (red) and $\tilde{x}(t) = x(t) + \mathcal{N}(0,100)$ $(blue)$.

e.

At the last step of the first part of the MATLAB assignment, the coefficients of the noisy signals are also tried to be found by using the same algorithm proposed in the first part. This time because of the component of the noise on the subspace defined above the values have changed with the change of the size of variance. First case will be the addition of a white normal gaussian to the original signal. The Φ_i values can be observed in Figure 7.

```
phi_list_1 =

  Columns 1 through 9

    0.0312    4.2191    0.0409    4.1940   -0.0379   -0.0103    0.0437    6.4063   -0.0054

  Columns 10 through 18

    4.2482    0.0637   -0.0059   -0.0140    2.7870   -0.0120    0.7294   -0.0143   -0.0443

  Columns 19 through 27

    0.0100    0.0129    4.2067    0.0030   -0.0225    2.1388    0.0538   -0.0464    4.2420

  Columns 28 through 30

    2.1195   -0.0110   -0.0054
```

Fig.7 The values of obtained Φ_is on the command window of MATLAB.

After selecting the ones with the maximum absolute value, the a_ks become the following in Figure 8.

```
ak_list_1 =

  Columns 1 through 9

    5.9667    5.9312    9.0598    6.0078    3.9415    1.0316    5.9491    3.0248    5.9992

  Column 10

    2.9974
```

Fig.8 The values of obtained a_ks on the command window of MATLAB.

The frequencies have found out to be the same with the frequencies in the Figure 3 which can be seen in Figure 9. As it can be seen values of the a_ks has changed with the order of ∓ 0.1 because of the component of the noise on the defined signal space.

```
fk_list_1 =

    10    60    80   100   150   180   190   220   270   280
```

Fig.9 The values of obtained f_ks on the command window of MATLAB.

After completing the case for the noise with variance one, the second case is the noise with variance 25. The same procedure has been applied and the values of the Φ_is and a_ks can be seen from Figures 10 and 11.

```
phi_list_5 =

  Columns 1 through 9

   -0.1940    4.1298    0.2202    4.4964    0.0022   -0.0930    0.0184    6.4311    0.1534

  Columns 10 through 18

    4.0972    0.0446   -0.3730   -0.0376    2.6435   -0.3674    0.9197    0.0029   -0.0682

  Columns 19 through 27

   -0.0878    0.1347    4.2342   -0.2721   -0.0310    2.1617   -0.1268   -0.0376    4.1241

  Columns 28 through 30

    2.3004   -0.2110    0.0067
```

Fig.10 The values of obtained Φ_i's on the command window of MATLAB.

```
ak_list_5 =

  Columns 1 through 9

    5.8404    6.3589    9.0950    5.7943    3.7385    1.3006    5.9880    3.0571    5.8324

  Column 10

    3.2532
```

Fig.11 The values of obtained a_ks on the command window of MATLAB.

As it can be seen the amount of the error has changed from ∓ 0.1 to ∓ 0.5. This is logical since the variance of the channel noise has increased. The f_ks which can be seen in Figure 12 have remained constant.

```
fk_list_5 =

    10    60    80   100   150   180   190   220   270   280
```

Fig.12 The values of obtained f_ks on the command window of MATLAB.

Lastly the signal with gaussian noise with variance 100 has again been processed using the same algorithm and the results can be seen from the Figures 13 and 14.

```
phi_list_10 =

  Columns 1 through 9

    0.2064    4.4521    0.1608    4.1545   -0.0147   -0.2659    0.4118    6.0356    0.1401

  Columns 10 through 18

    4.6330    0.4438    0.2018   -0.2260    2.8669   -0.1365    0.9764    0.0154    0.0128

  Columns 19 through 27

   -0.0759    0.3910    4.4494    0.2356   -0.4902    2.0956   -0.0900    0.2615    3.9420

  Columns 28 through 30

    1.7729    0.1359    0.1757
```

Fig.13 The values of obtained Φ_i's on the command window of MATLAB.

```
ak_list_10 =

  Columns 1 through 9

    6.2962    5.8754    8.5356    6.5520    4.0544    1.3809    6.2925    2.9637    5.5749

  Column 10

    2.5073
```

Fig.14 The values of obtained a_ks on the command window of MATLAB.

The error rate has changed to ∓ 1 because of the noise variance. Again, the frequencies have been remained the same since the noise is not strong enough to distort the coefficients such that the wrong frequencies are selected. Frequencies can be seen from Figure 15.

```
fk_list_10 =

    10    60    80   100   150   180   190   220   270   280
```

Fig.15 The values of obtained f_ks on the command window of MATLAB.

Part II: Binary Modulation

For the part of the binary modulation, I will be using the signal $\Lambda_0(t)$. Plot of this signal can be seen in Figure 16.

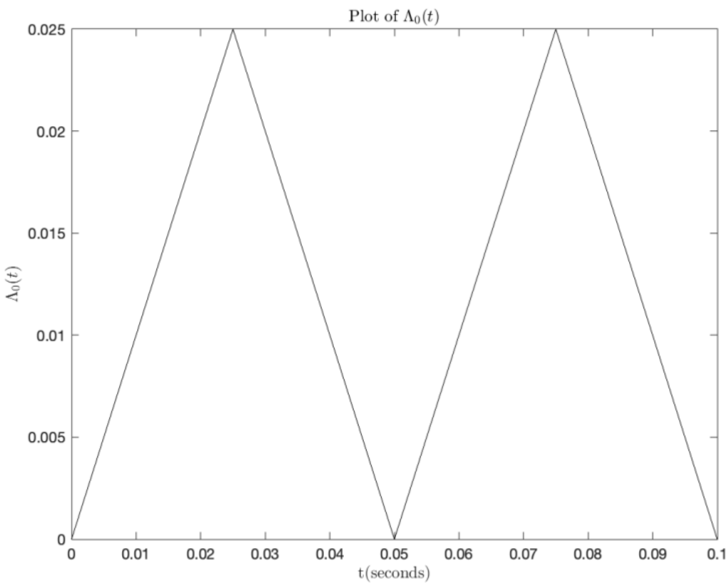

Fig.16 Plot of the signal $\Lambda_0(t)$ which is sampled with 1000 Hz sampling rate.

a.

5 random bits has been generated following the homework guide; the bit sequence is "01110". The modulated version of this signal with sampling frequency of 1000 Hz can be seen from Figure 17.

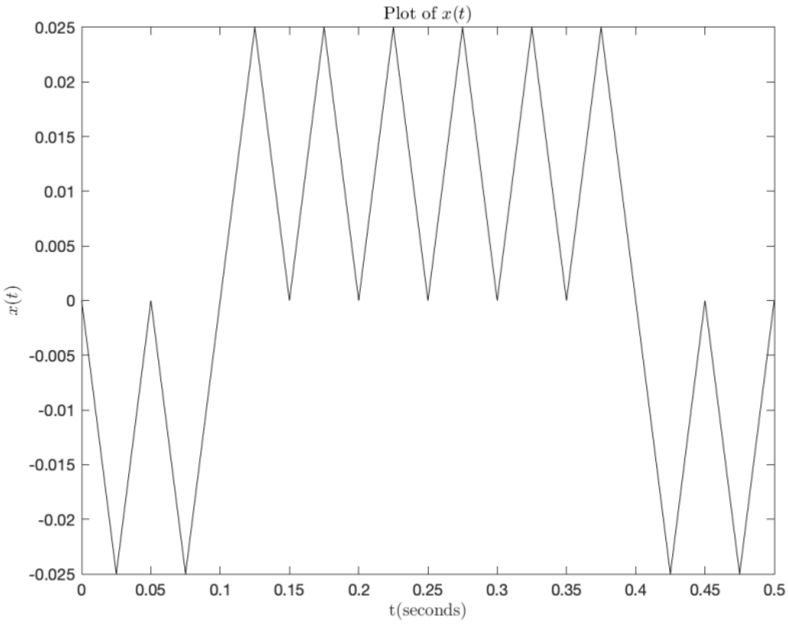

Fig.17 Plot of the modulated digital signal "01110" which is sampled with 1000 Hz sampling rate.

b.

For this signal space one basis signal will be enough to generate all possible signals therefore the orthonormal basis consists of only one signal which will be the normalized version of the signal $\Lambda_0(t)$. The signal that creates the basis can be found by the following operations:

$$\Psi_0(t) = \frac{\Lambda_0(t)}{||\Lambda_0(t)||} = \frac{\Lambda_0(t)}{\sqrt{\int_0^{0.1} \Lambda_0(t)^2 \mathrm{dt}}} = \frac{\Lambda_0(t)}{\sqrt{\frac{0.1^3}{48}}}$$

Plot of this basis function can be seen in Figure 18.

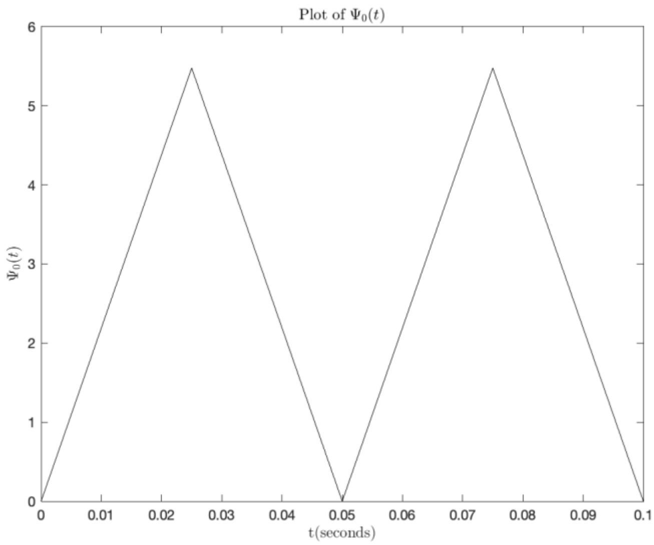

Fig.18 Plot of the signal $\Psi_0(t)$ which is sampled with 1000 Hz sampling rate.

The representation of the two modulated signals $\Lambda_0(t)$ and $-\Lambda_0(t)$ in this signal space can be seen in Figure 19.

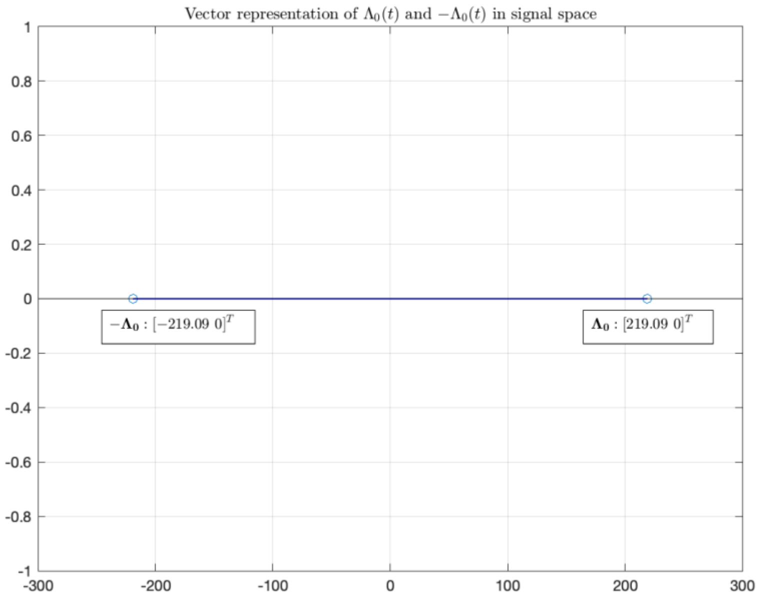

Fig.19 Vector representation of the $\Lambda_0(t)$ and $-\Lambda_0(t)$ in the proposed signal space

c.

In this part a white gaussian noise with three different variances (1e-4, 1e-2 and 1) has been added on top of the generated signal in Figure 17. The corresponding plots of these signals on top of the original signal can be seen in the Figures 20, 21, and 22 respectively.

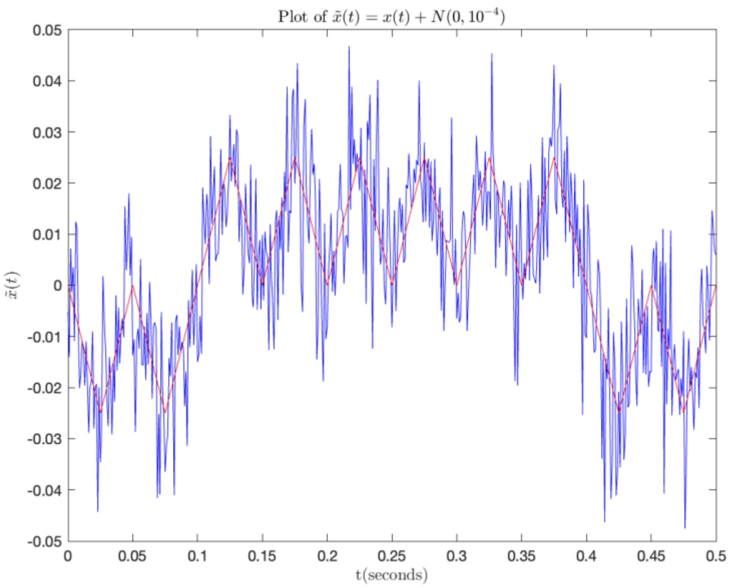

Fig.20 Plot of the signal $\tilde{x}(t) = x(t) + \mathcal{N}(0,10^{-4})$ (blue) and $x(t)$ (red).

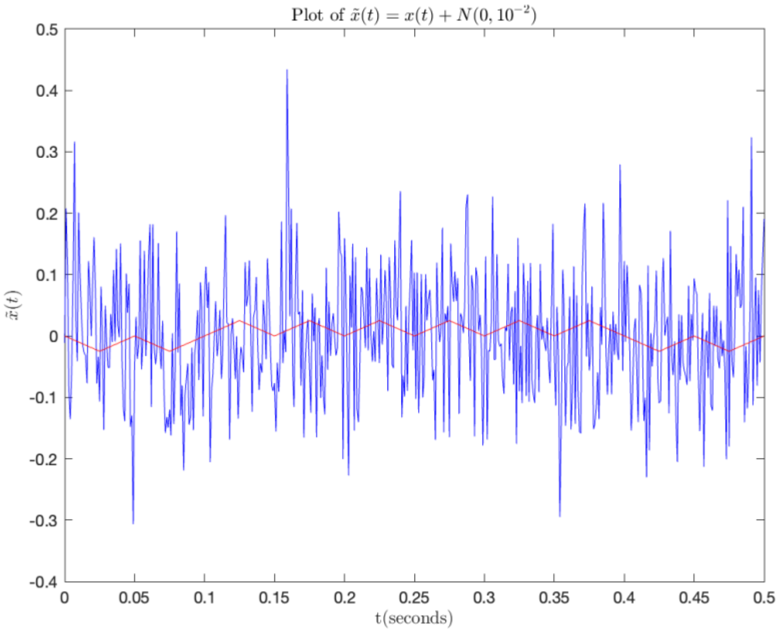

Fig.21 Plot of the signal $\tilde{x}(t) = x(t) + \mathcal{N}(0,10^{-2})$ (blue) and $x(t)$ (red).

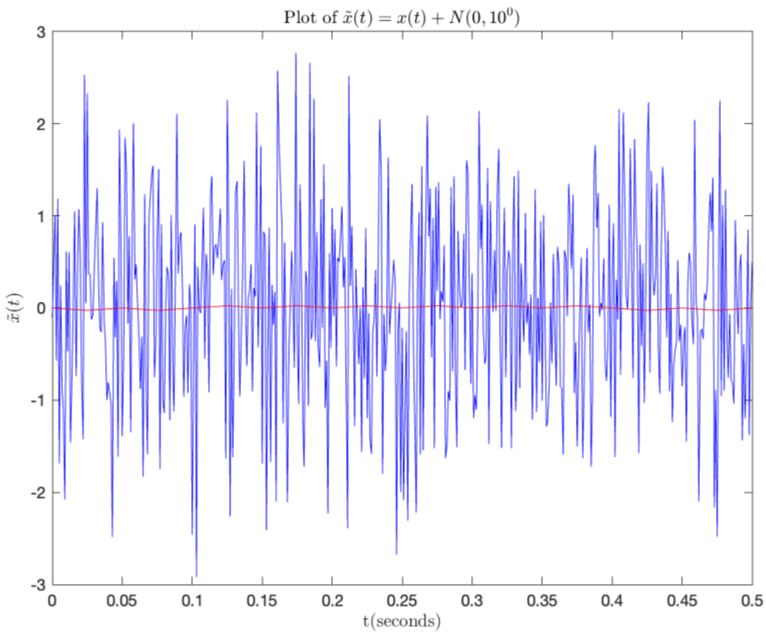

Fig.22 Plot of the signal $\tilde{x}(t) = x(t) + \mathcal{N}(0,1)$ (blue) and $x(t)$ (red).

SNR can be briefly explained as the measure of signal power to noise power of a given signal. It can be mathematically represented as the following:

$$\Upsilon_s = \frac{E_s}{N_0}$$

For a discrete signal sampled from a continuous signal, the average signal power can be calculated using the following formula:

$$E_s = \frac{1}{F_s}\sum_{n=1}^{F_s} x[n]^2$$

Using this formula, SNR can be expressed as the following:

$$SNR = \frac{\frac{1}{F_s}\sum_{n=1}^{F_s} x[n]^2}{\sigma^2}$$

Therefore, increasing or decreasing the sample size does not directly or correlatedly change the value of the SNR especially above the Nyquist frequency, where the signal can be represented fully in discrete time. Also from another perspective, the noise is an AWGN, therefore the noise for each infinitesimal time instance of the continuous time signal is independent from each other.

So, increasing the sample amount increases the total power of the discrete signal but it also increases the amount of independent noise components contributing to the discrete time signal. The formula from another perspective can be the following:

$$SNR = \frac{\sum_{n=1}^{F_s} x[n]^2}{\sum_{n=1}^{F_s} \tilde{n}\,[n]^2} \cong \frac{\sum_{n=1}^{F_s} x[n]^2}{F_s \cdot \sigma^2}$$

But it must be noted that sampling the signal effects the bandwidth of the signal, which eventually effects the gaussian noise that will add up on top of the signal. Since the noise is white gaussian, it has infinite full spectrum noise power. When adding this signal on top of the signal of interest the noise variance changes with bandwidth of the signal. Therefore, it can be argued that the noise power becomes T_s times larger after sampling the signal. Which must be noted during creating signals on MATLAB.

For sufficiently large F_s. The value of SNR can increase or decrease depending on the variance of the AWGN of the channel which can be seen from Figures 20, 21, and 22.

d.

For the case of the binary modulation the following block model in Figure 23 can be used as an optimal receiver to detect the messages sent. The boundary for the separation of the vectors in the orthonormal space can be selected as 1 since the modulated messages are symmetric which can be seen from Figure 19.

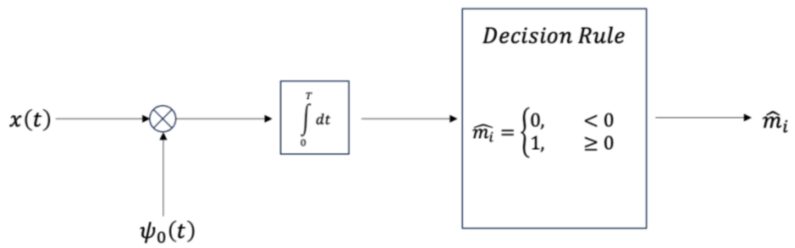

Fig.23 The optimal receiver design to demodulate the received signal.

Since the discrete signals being used in the MATLAB are sampled versions of the original continuous signal, the optimal receiver in Figure 23 will be changed with its discrete version which can be seen in Figure 24.

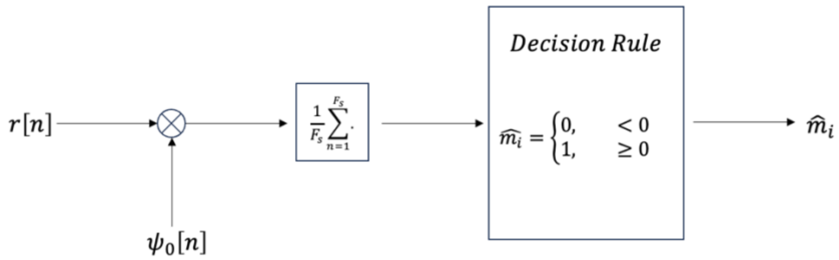

Fig.24 The optimal receiver design to demodulate the received signal in MATLAB.

Following this the probability of error can be calculated using the following formula:

$$P_e = \frac{1}{2}\left(P_{e,1} + P_{e,2}\right)$$

Because of symmetry,

$$P_{e,1} = P(\widehat{m}_1 \neq m_1 \,|\, m_1\ sent) = P_{e,2} = P(\widehat{m}_2 \neq m_2 \,|\, m_2\ sent)$$

Therefore,

$$P_e = P_{e,1}$$

One of the error probabilities can be found using the following operations:

$$P_{e,1} = P(r > \ d \,|\, m_1\ sent)$$

$$P_e = P\left(\mathcal{N}\left(0, \frac{N_0}{2}\right) > d\right) = Q\left(\frac{d}{\sqrt{\frac{N_0}{2}}}\right) = \boxed{Q\left(\sqrt{\frac{2d^2}{N_0}}\right)}$$

e.

To compare the theoretical and empirical values of the probability of error, a random bit sequence of length 10^5 has been generated and a white gaussian noise has been applied on the signal. The noise variance has been selected with an amount that will cause a probability of error between 0.5 and 10^{-4}. Intuitively, a probability of error of 0.5 in the case of two events that has equal priors correspond to totally randomized message predictions which might occur due to high noise power. Mathematically,

$$P_e = 0.5 = Q\left(\frac{d}{\sqrt{\frac{N_0}{2}}}\right)\Bigg|_{\frac{d}{\sqrt{\frac{N_0}{2}}}=0}$$

$$\lim_{N_0 \to \infty} \frac{d}{\sqrt{\frac{N_0}{2}}} = 0$$

The second case is the probability of error equal to 10^{-4},

$$P_e = 10^{-4} = Q\left(\frac{d}{\sqrt{\frac{N_0}{2}}}\right)\Bigg|_{\frac{d}{\sqrt{\frac{N_0}{2}}} \cong 3.72}$$

$$N_0 = 2 * \left(\frac{d}{Q^{-1}(10^{-5})}\right)^2$$

Then,

$$d = \sqrt{E} = \sqrt{\frac{0.1^3}{48}}$$

Then,

$$N_0 = 3 \cdot 10^{-6}$$

Therefore, the probability of error has been calculated for values of noise variance which is $0.5N_0$ and between $1.5 \cdot 10^{-6}$ *and* 1. Plot of the probability of error vs the value of SNR can be seen from the Figure 25.

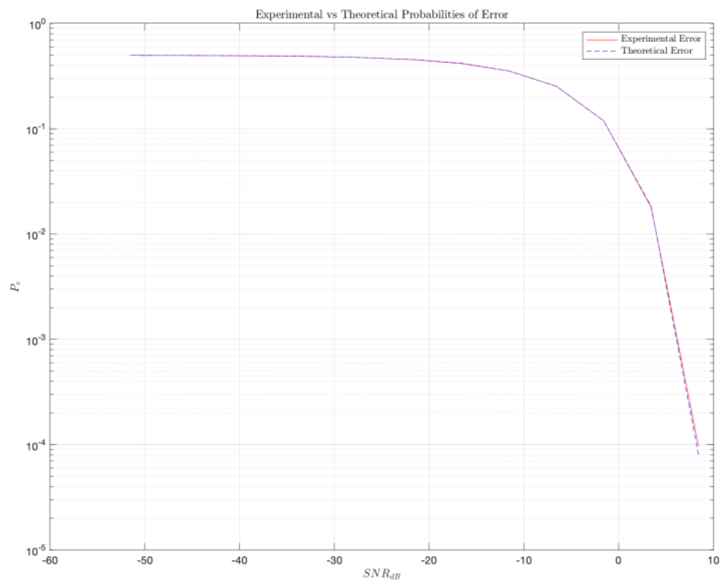

Fig.25 Probability of error vs SNR plot for theoretical and experimental results

f.

For the case where the priors of the messages are not equal the optimal detection boundary changes since the probability of making a mistake in detecting each message changes. The optimal boundary can be derived using the following formula by using the inverse priors of the probability of the posterior distributions:

$$f(m_1\ sent) = \frac{9}{10}\frac{1}{\sqrt{\pi N_0}}\ e^{\frac{(x-\sqrt{E})^2}{N_0}} = \frac{1}{10}\frac{1}{\sqrt{\pi N_0}}\ e^{\frac{(x+\sqrt{E})^2}{N_0}}$$

Take log of both sides

$$\frac{(x-\sqrt{E})^2}{N_0} + \log 9 = \frac{(x+\sqrt{E})^2}{N_0}$$

$$4x\sqrt{E} = N_0 * \log 9$$

$$x = \frac{N_0 \cdot \log 9}{4\sqrt{E}}$$

So, the optimal receiver structure for the given messages can be seen in Figure 26:

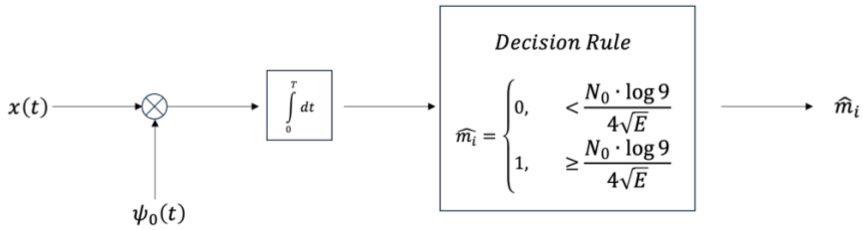

Fig.26 The optimal receiver design to demodulate the received signal in MATLAB.

Result of using this receiver model can be seen from the Figure 27 also with using the optimal receiver for equal priors in the case of nonequal priors.

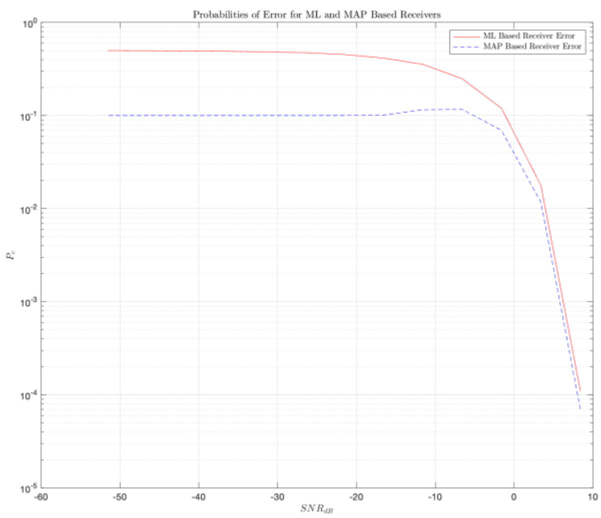

Fig.27 Probability of error for source with nonequal priors using optimal and nonoptimal receivers.

Since there is a noise variance term in the formula of the boundary, as the variance decreases which means the SNR increases the boundary gets closer to zero which will cause the plots to get closer to each other and converge eventually.

Lastly the effect of the prior distribution has been inspected. The value of the prior has been swiped through 0 to 0.5 to see the effect. The proposed MAP estimator is the following given a prior for the message $P(m_1 = 1\ sent) = \alpha$.

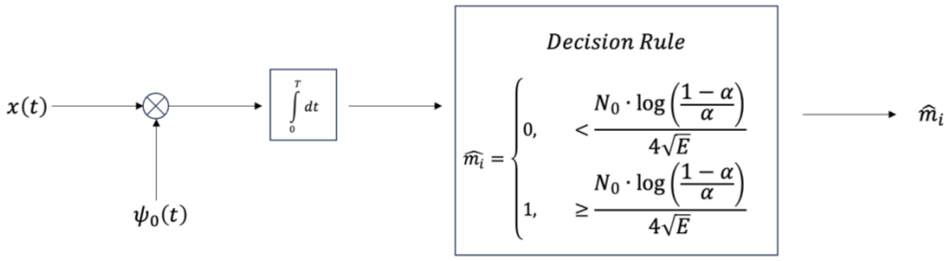

Fig.28 Proposed optimal receiver for selecting different priors to send messages.

As it can be seen from the experimental results which are displayed in Figure 29, the ML based estimator performs in its worst case no matter what value the alpha takes. This is because the ML estimator already performs with 0.5 probability of error in the equal prior case and the change of prior does not affect the decision boundary of the receiver in part e. But the receiver proposed in the last part has a boundary that changes with the value of alpha, where the receiver predicts all values as 0 if the prior is 0, which leads to 0 error probability no matter what the noise is. Therefore the MAP estimator performs better than the MLE based receiver in noisy situations where the prior is known to the model.

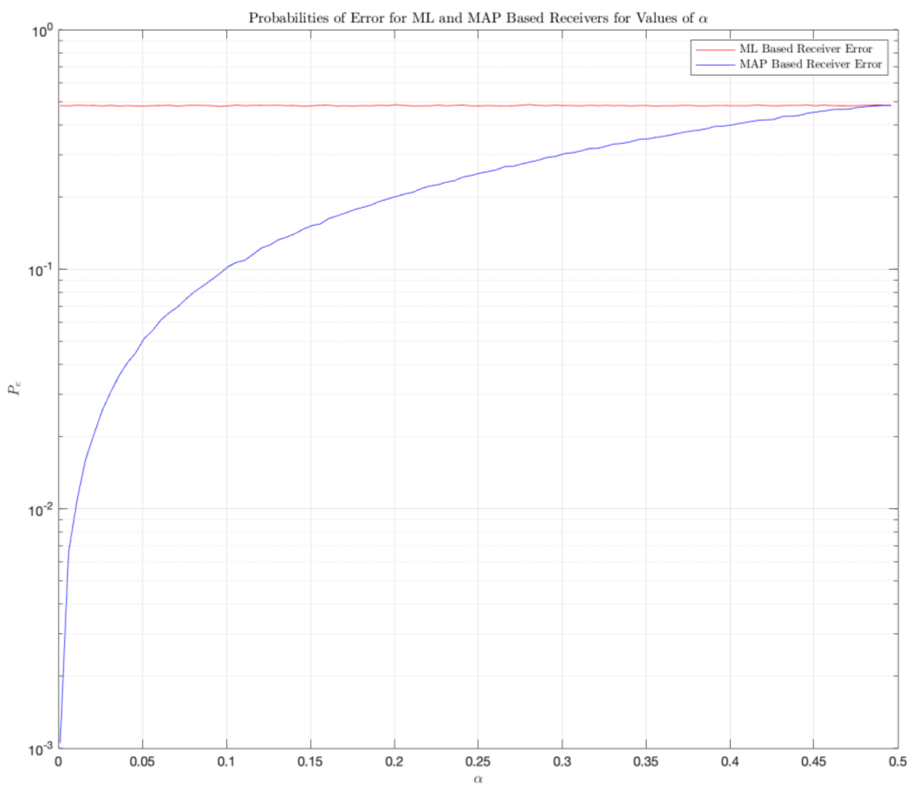

Fig.27 Probability of error for source with nonequal priors α , using optimal (MAP) and non-optimal (MLE) based receivers.