

YAPILAN ÇALIŞMALAR HAKKINDA BİLGİLER

Ben **MUHAMMET KOÇ**, bu dosyada yaptığım çalışmalar sonucu yazdığım raporları veya ilgili bilgileri paylaşacağım.

İÇİNDEKİLER

1. G-TÜRK İHA TAKIMI- Elektronik sorumlusu
2. Website Tasarlama
3. TWİTTER VERİLERİ ÜZERİNDEN DUYGU ANALİZİ
- 4 THE MATLAB PROJECT REPORT: DSB-LC-AM
- 5 THE MATLAB PROJECT REPORT: FM MODULATION AND DEMODULATION
- 6 DİJİTALSİNYALİŞLEME: DRONE İLE ÇEKİLEN VİDEOLarda KANAT SESİNİ AYIKLAMA
- 7 THE MATLAB PROJECT REPORT :BASEBAND DIGITAL COMMUNICATION SIMULATION
- 8 EXPERIMENT Sampling and Reconstruction
- 9 EXPERIMENT BPSK AND QPSK
- 10 CONTROL SYSTEM THEORY COURSE PROJECT: THE ANTENNA POSITION CONTROL SYSTEM

1. G-TÜRK İHA TAKIMI- Elektronik sorumlusu

SABİT KANAT İHA AR-GE'si 5 kişi ile yürüttüğümüz bir Teknofest projesidir. Ben bu takımda elektronik sistemden sorumluydum. Yarışma sürecinde kavramsal tasarım raporu ve detaylı tasarım videoları aşamalarını geçerek finale katılmaya hak kazandık. Aşağıda kavramsal tasarım raporunun benim yazdığım kısmını ekleyeceğim. Detaylı tasarım videosu [link](#).

| | | |
|--|--|--|
|  G-TÜRK G-TÜRK İHA Takımı KAPTAN-Muhammet Kılınç | | |
| YAZILIM | ELEKTRONİK | MEKANİK |
| ABDULSAMED ASLAN Bilgisayar Müh. 2.sınıf Görüntü İşleme | Muhammet Koç Elektronik Müh. 2.sınıf Elektronik Sistemler | Mehmet Haluk Metin Makine Müh. 2.Sınıf Tasarım ve İmalat |
| BERKCAN NARMOT Makine Müh. 2.sınıf Otonom Sistemler | | Muhammet Kılınç Makine Müh. 2.sınıf Analiz ve Uçuş Mekaniği |

1.2.4. Elektrik Elektronik Kontrol ve Güç Sistemleri

Elektronik aksamlar seçilirken önceki sistemler incelemiş ve kullanılabilecek malzemeler hakkında fikir sahibi edinilmiştir. İHA'ya gerekli olan itkiyi sağlama için **fırçasız DC motor** kullanılmıştır. Sistemde, fırçasız DC motoru kontrol etmek için **Elektronik hız kontrol (ESC)** devresi bulunmaktadır. İstikamet dümeni, irtifa dümeni ve aileronların kontrolü için 4 tane **servo motor** vardır. Aynı zamanda bu hareketli kısımların kontrolü ve kullanımı için otonom uçuşa uygun **uçuş kontrol kartı** ve çeşitli görevler için ise bir **yardımcı bilgisayar** yer alır.

Uçuş kontrol kartını seçerken kullanacak olduğumuz Ardupilot oto pilot yazılımı için kullanılabilen en uygun, açık kaynak yazılıma sahip, ayı zamanda mikrokontrolcü tipinde kartlar arasında öne çıkan Pixhawk ailesi olmuştur. Pixhawk ailesi arasında sensörleri ve sahip olduğu işlemci karşılaştırılarak STM32F765 işlemcisi bulunan, 2 adet 6 eksenli ivmeölçerlere, barometreye sahip olan, ekstradan farklı tiplerde iletişim portlarını destekleyen Pixhawk 4 modelinde karar kılınmıştır. GPS modülü olarak pulsusu dahili olan Ublox Neo-m8n çipe sahip, güvenlik anahtarı ve buzzer bulunan Pixhawk 4 GPS modülü kullanılacaktır. Telemetri modülü için uçtan uca şifreleme özelliğine sahip, programlanabilen Digi Xbee modülü kullanılacaktır. Yer istasyonu olarak RF kumanda ve bilgisayar kullanılacaktır. Bilgisayar üzerinden Mission Planner yazılımı ile görevler idame ettirilecektir.

Yardımcı bilgisayar, muadilleri arasında uygun fiyat ve güvenilirlik açısından, görüntü işleme için yeterli işlemci gücü sağlayan Raspberry Pi 4 model b seçilmiştir. Raspberry Pi 4 diğer Raspberry modellerine göre daha yüksek işlemci gücü sunmaktadır.

Fırçasız DC motorun seçiminde ağırlık (Şekil-18) göz önünde bulundurulmuştur. Elektronik kısımlar ve batarya kütlesi 674 gram olarak hesaplanmıştır. İtki bölümünde (1.2.5.) belirtilen seyir şartlarına uygun 600W güce ulaşabilen SunnySky X2814 Fırçasız DC motora karar verilmiştir. Motorun data sheet dosyasına bakılıp maksimum güte 40.3A çekeceği görülmüştür. Bu bilgiye dayanarak hız kontrol devresi Hobbyking Skywalker 50A ESC tercih edilmiştir. Skywalker 50A ESC 5V/5A ubec devresine sahiptir. Buradan sağlanan güç ile servo motorlar beslenecektir.

Batarya seçiminde gerekli uçuş süresi önemli bir kriterdir. Sistemin dakikada ortalama kaç amper çektığı, bataryanın dakikada verdiği elektriksel güce bölünerek uçuş süresi hesaplanabilir.

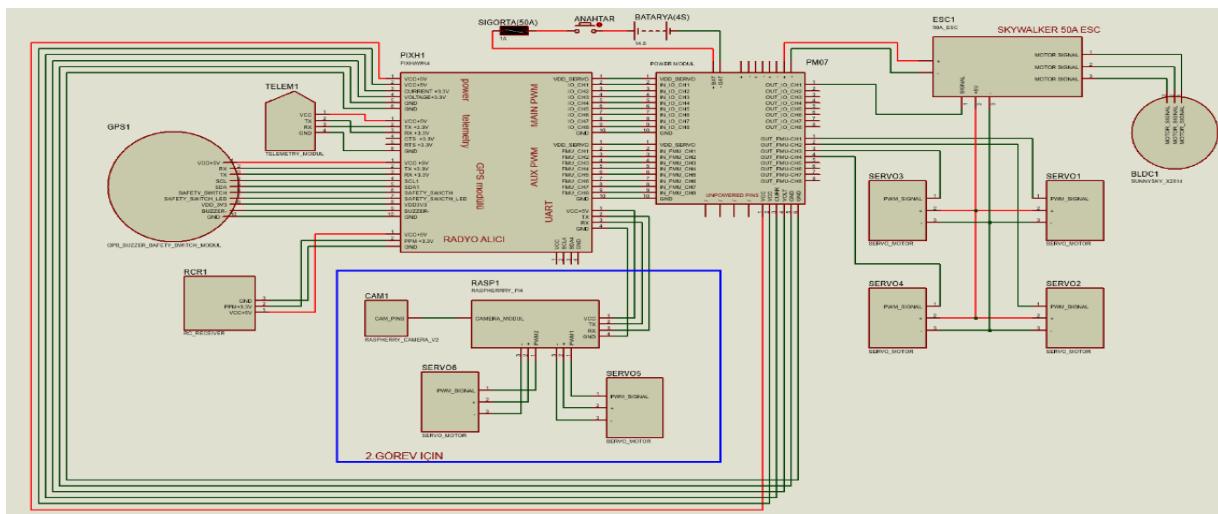
$$4m = 0.066h = \frac{2.2 Ah}{30 A}$$

Motorun ortalama 30A tüketeceği göz önünde bulundurularak yapılan hesaplarda 4 dakikadan fazla uçuş süresi sunması görevler tekrar durumunda yeterli görünen Gens Ace 4S 2200 mAh (30C) batarya tercih edilecektir. Hücre sayısını belirleyen faktör motorun çalışma gerilimidir. Motor 14.8V ile çalıştırılacağı için 4s seçilmiştir. C katsayı ile anlık çekilebilen akımı miktarını hesaplayabiliyoruz. Bu batarya için $30C \times 2.2A = 66A$ anlık maksimum akım alınaması beklenmektedir.

$$\text{uçuş süresi (saat)} = \frac{\text{bataryanın verdiği elektriksel yük (amper · saat)}}{\text{ihanin çektiği akım (amper)}}$$

görmüştür. Batarya güç modülüne dahildir. Güç modülünü açıklamak

gerekirse batarya, anahtar ve sigortadan oluşur. Öngörülemeyen fazla akım çekmelerinden sistemi korumak için 60A bıçak sigorta kullanılacaktır. Anahtar, İHA kullanılmayıken akımı kesmesi için eklenmiştir. Sistemin devre şeması Proteus programıyla çizilmiş, Şekil 14'te gösterilmiştir.



Şekil 14: Devre Şeması

2. Website Tasarlama

Pandemi döneminde web site tasarlamaya merak saldım. Kendi web sitemi yaptıktan sonra İHA takımıma rahatça tanıtım amacıyla web sitemizin olmasının iyi olacağını düşündük. Bu görevi ben üstlendim. İHA takımıtan, ekipteki kişiler, amacımız ve iletişim adreslerini içeren sadece frontend(Html,css,javascript) temelli tek sayfadan oluşan bir siteyi tasarladım ve yayınladık. Kodlar github [mhtkoc/gturiha](https://github.com/mhtkoc/gturiha) repomda bulunmaktadır. Daha sonra okulumuzda AR-GE Teknoloji ve İnovasyon Topluluğu adında bir topluluk oluşturduk. Bu topluluk içinde sadece frontend(Html,css,javascript) temelli bir site tasarladım. GTUATİK sitesi anasayfa, hakkımızda, iletişim ve etkinlikler sayfalarından oluşmaktadır. Kodlar github [mhtkoc/gtuatik](https://github.com/mhtkoc/gtuatik) repomda bulunmaktadır.

Site Linkleri:

İHA Takımı: <https://mhtkoc.github.io/gturiha/>

Topluluk sitesi: <https://mhtkoc.github.io/gtuatik>

TWİTTER VERİLERİ ÜZERİNDEN DUYGU ANALİZİ

Muhammet Koç

muhammetkoc2019@gtu.edu.tr

İsmail Akkoyun

i.akkoyun2019@gtu.ed.tr

²Elektronik Mühendisliği Bölümü, GTÜ, Kocaeli, Türkiye

I. GİRİŞ

Duygu Analizi, verilen bir metnin duyu durumunu sınıflandırma görevidir. Örneğin, metin tabanlı bir tweet, "olumlu", "olumsuz" veya "tarafsız" olarak kategorize edilebilir. Metin ve eşlik eden etiketler verildiğinde, bir model doğru duyu durumunu tahmin etmek üzere eğitilebilir.

Duygu Analizi teknikleri, makine öğrenimi yaklaşımı, sözlük tabanlı yaklaşım ve hibrit yöntemler olmak üzere çeşitli kategorilere ayrılabilir. Duygu analizi araştırmasının alt kategorileri arasında multimodal duyu analizi, yön bazlı duyu analizi, ince detaylı görüş analizi, dil özel duyu analizi bulunmaktadır.

1) Makine Öğrenimi Tabanlı Yöntemler:

- Doğal Dil İşleme (NLP) Modelleri (CNN, LSTM)
- Destek Vektör Makineleri (SVM)
- Karar Ağaçları ve Random Forest

2) Sözlük Tabanlı Yöntemler:

- Duygu Sözlükleri
- Duygu Puanlama Yöntemleri.

3) Hibrit Yöntemler:

- Makine Öğrenimi ve Sözlük Birleşimi

Ayrıca yöntemleri gözetimli gözetimsiz olarak listelersek:

1. Gözetimli Modeller:

- Sınıflandırma Modelleri: Bu modeller, belirli bir metni önceden tanımlanmış duyu kategorilerine sınıflandırmak için kullanılır. Örneğin, olumlu, olumsuz veya tarafsız.

Örnek algoritmalar: Destek Vektör Makineleri (SVM), Karar Ağaçları, Derin Sinir Ağları (CNN, LSTM).

- Regresyon Modelleri: Bu modeller, metindeki duygusal yoğunluğu belirlemek için kullanılır.

Metnin içeriği duyu düzeyini sürekli bir değer olarak tahmin eder.

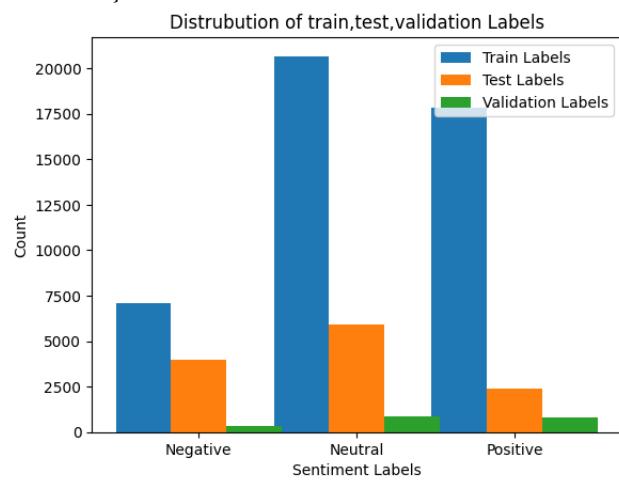
Örnek algoritmalar: Lineer Regresyon, Destek Vektör Regresyonu.

Gözetimli öğrenme yöntemleri kullanılarak uygulama gerçekleştirilmiştir. Kullanılan algoritmalar SVM, K-Nearest-Neighbor ve LSTM ile sonuçlar elde edilmiştir. K-means algoritması ile duyu analizi yapılamaz. K-Means Algoritması gözetimsiz bir yöntemdir. K-means ile yazılan makaleler bulunsa da itibar edilmemesi gereken bir yaklaşımındır.

II. METEDOLOJİ

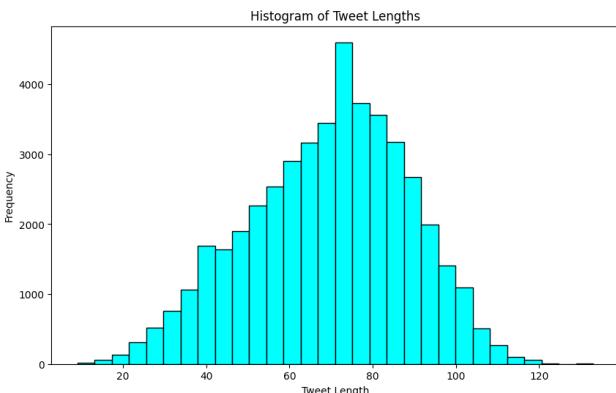
A. Veriye ön işlem uygulamak

Kullanılan veri seti TweetEval veri setinin Sentiment alt setidir. Veriyi yükleyip, Metin içinde sayı filtresi, noktalama işaretleri filtresi, N-karakter filtresi (N karakterden küçük tokenlar), durak kelimeleri(stop words) filtresi uygulanmıştır. Tokenization gerekli olduğu yerlerde Bert modelinin tokenizer modülü kullanılmıştır. Şekil 1'de veri seti içinde verilerin dağılımı gösterilmektedir. Train veri seti içinde 45615 etiketli tweet bulunmaktadır.



Şekil 1: verilerin dağılımı

ELM472 Makine Öğrenmesinin Temelleri



Şekil 2: Tweetlerin sahip olduğu karakterlere sayılarına göre dağılımı

Şekil 2'de Tweetlerin sahip olduğu karakterlere sayılarına göre dağılımı gösterilmektedir.



Şekil 3: Veri seti için kelime bulutu

Şekil 3’te veri seti içinde bulunan kelimelerin sıklıklarına göre oluşturulan kelime bulutu gösterilmektedir. Buradan en sık geçen kelimenin ‘user’ olduğu anlaşılmaktadır.

B. Özellik Çıkarımı(Feature Extraction)

Özellik çıkarımı için Metinleri numerik olarak ifade etme yöntemlerinden TfIDF, Word2vec ve Countvector yöntemleri yanı sıra Sentiwordnet gibi sözlüklerden oluşturulan tokenlardan elde skorlar ile özellikler çıkarılmıştır.

C. SVM Algoritmasinin uygulanmasi

SVM (Support Vector Machine), özellikle sınıflandırma ve regresyon problemleri için kullanılan bir makine öğrenimi algoritmasıdır. SVM, veri noktalarını sınıflarına ayırmak için bir hiper düzlemler oluşturur. Bu hiper düzlemler, veri noktalarını sınıflarına göre iki farklı tarafına böler.

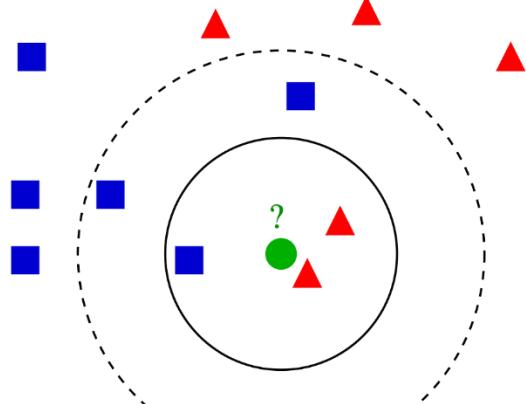
SVM'nin temelde iki ana türü vardır: doğrusal SVM ve çekirdek SVM. Doğrusal SVM, veri noktalarını doğrusal bir hiper düzlemlle ayıırken, çekirdek SVM, veriler arasında daha karmaşık ilişkileri modellemek için bir çekirdek fonksiyonu kullanır. Uygulamada Doğrusal SVM kullanılmıştır. Şekil 4'te SVM sonuçları gösterilmiştir

Şekil 4: SVM sonuçları

D. KNN Algoritmasinin uygulanmasi

KNN (K-Nearest Neighbors) Algoritması iki temel değer üzerinden tahmin yapar;

- Distance (Uzaklık): Tahmin edilecek noktanın diğer noktalara uzaklıği hesaplanır. Bunun için Minkowski uzaklık hesaplama fonksiyonu kullanılır.
 - K (komuşuluk sayısı): En yakın kaç komşu üzerinden hesaplama yapılacağını belirtir. K değeri sonucu direkt etkileyecektir. $K=1$ olursa overfit etme olasılığı çok yüksek olacaktır. Çok büyük olursa da çok genel sonuçlar verecektir. Bu sebeple optimum K değerini tahmin etmek problemin asıl konusu olarak karşımızda durmaktadır. K değerinin önemini aşağıdaki Şekil 5 güzel bir şekilde göstermektedir. Eğer $K=3$ (düz çizginin olduğu yer) seçersek sınıflandırma algoritması ? işaretü ile gösterilen noktayı, kırmızı üçgen sınıfı olarak tanımlayacaktır. Fakat $K=5$ (kesikli çizginin olduğu alan) seçersek sınıflandırma algoritması, aynı noktayı mavi kare sınıfı olarak tanımlayacaktır.



Sekil 5:KNN k parametresi için gösterim

ELM472 Makine Öğrenmesinin Temelleri

KNN (K-Nearest Neighbors) Algoritması ile üretilmiş bir modelin başarısını ölçmek için genel olarak kullanılan 3 adet indikatör vardır.

- **Jaccard Index:** Doğru tahmin kümesi ile gerçek değer kümesinin kesişim kümesinin bunların birleşim kümesine oranıdır. 1 ile 0 arası değer alır. 1 en iyi başarım anlamına gelir.
- **F1-Score:** Confusion Matriks üzerinden hesaplanan Precision ve Recall değerlerinden hesaplanır. $\text{Pre}=\text{TP}/(\text{TP}+\text{FP})$ $\text{Rec}=\text{TP}/(\text{TP}+\text{FN})$ $\text{F1-Score}=2((\text{Pre}\text{Rec})/(\text{Pre}+\text{Rec}))$ 1 ile 0 arası değer alır. 1 en iyi başarım anlamına gelir.
- **LogLoss:** Logistic Regresyon sonunda tahminlerin olasılıkları üzerinden LogLoss değeri hesaplanır. 1 ile 0 arası değer alır. Yukarıdaki iki değerden farklı olarak 0 en iyi başarım anlamına gelir.

Sonuçlarda F1 skor parametresi kullanılmıştır.

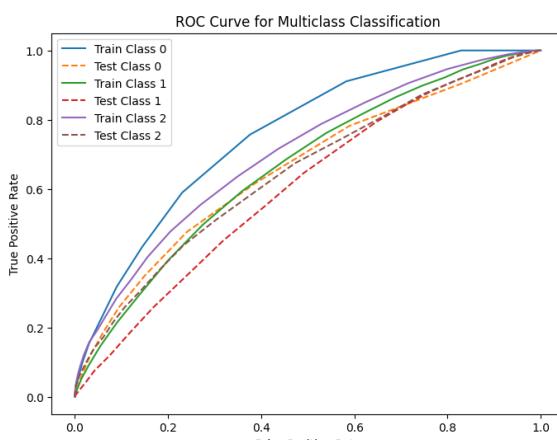
Uygulama yapılrken Feature vektörü için hem TFIDF hemde CountVector yöntemleri ile vektörler elde edilmiştir.

| Classification Report: | | | | |
|------------------------|-----------|--------|----------|---------|
| | precision | recall | f1-score | support |
| 0 | 0.53 | 0.09 | 0.15 | 3972 |
| 1 | 0.50 | 0.84 | 0.63 | 5937 |
| 2 | 0.39 | 0.27 | 0.32 | 2375 |
| accuracy | | | 0.49 | 12284 |
| macro avg | 0.48 | 0.40 | 0.37 | 12284 |
| weighted avg | 0.49 | 0.49 | 0.41 | 12284 |

Şekil 6: TFIDF feature vektörü ile elde edilen KNN sonuçları

```
Best Parameters: {'n_neighbors': 40, 'metric': 'manhattan'}
Best Accuracy: 0.4992475588459236
```

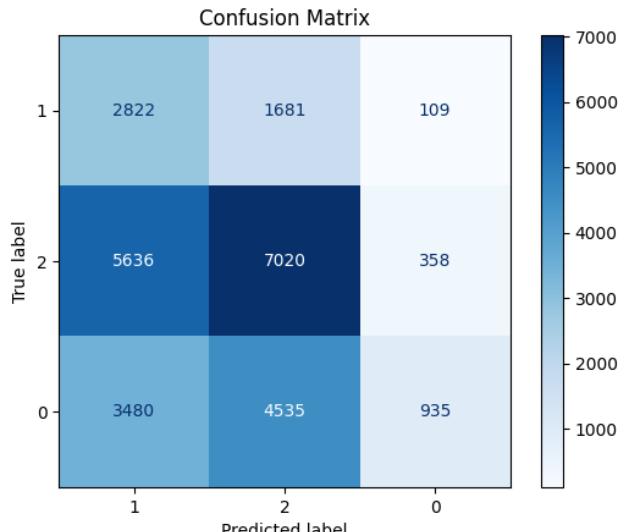
Şekil 7: Countvector ile elde edilen KNN algoritması sonuçları



Şekil 8: Countvector ile KNN Roc Curve

E. Sentiwordnet Duygu Sözlüğü Kullanımı

Sentiwordnet ile tokenların duygu skoruna göre pozitif skor pozitif, negatif skor negatif olarak değerlendirildiğinde; sentiwordnet ile doğruluk Oranı: 0.41



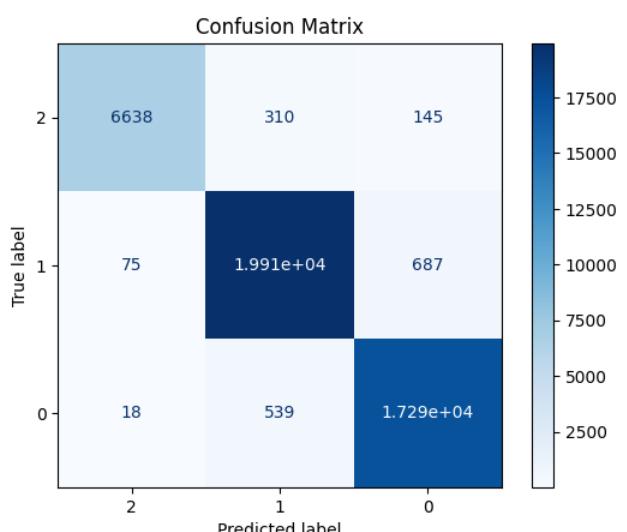
Şekil 9: sentiwordnet ile elde edilen matris

F. LSTM Algoritmasının Uygulanması

LSTM: Eğitim sırasında model, metin verilerindeki karmaşık ilişkileri öğrenir.

LSTM metin verilerindeki belirgin özellikleri öğrenme yeteneği ile bilinir ve duygu analizi için genellikle iyi performans gösterir.

LSTM ile Doğruluk Oranı: 0.96



Şekil 10:LSTM etiketleri ile elde edilen etiketlerin karmaşılık matrisi

ELM472 Makine Öğrenmesinin Temelleri

III. SONUÇLAR

A. Küme Sayısı

Duygu analizi için kümeler pozitif, negatif ve nötr olarak seçildiği için küme sayısı parametresi $k=3$ olarak kullanılmıştır.

B. Küme görselleştirmesi

Train veri setinde bulunan gerçek etiketler Şekil 2'de gösterilmiştir. Veri setinde 45615 tweet bulunmaktadır.

C. Modellerin Karşılaştırılması

Duygu analizi için kullanılabilen birçok yöntem bulunmaktadır. En iyi model olarak söylenebileceğim bir yöntem yoktur. Uygulamanın gereksinimlerine göre modeller seçilmelidir.

Ayrıca K-means gibi gözetimsiz modeller duygusal analizi için kullanılmaz

KAYNAKÇA

- [1] E. Alpaydin, Introduction to Machine Learning, 3. bs. Cambridge, MA, USA: MIT Press, 20[1] E. Alpaydin, Introduction to Machine Learning, 3. bs. Cambridge, MA, USA: MIT Press, 2014.14.

GEBZE TECHNICAL UNIVERSITY

ELECTRONIC ENGINEERING DEPARTMENT

**ELEC 361
ANALOG COMMUNICATIONS SYSTEMS**

**A MATLAB PROJECT REPORT
DSB-LC-AM**

MUHAMMET KOÇ

1901022017

1. Table of contents

Contents

| | |
|---|----|
| 1. Table of contents | 1 |
| 2. Abstract | 2 |
| 3. Matlab Solution | 3 |
| 3.a. Plotting $m(t)$, $x(t)$, $z(t)$ | 3 |
| 3.b.1 Plotting $M(f)$, $X(f)$, $Z(f)$ | 5 |
| 3.b.2 Plotting $Y(f)$ and $y(t)$ | 7 |
| 3.c. Demodulating $y(t)$ by using envelope detector | 9 |
| 4. Analytic Solution | 11 |
| 5. Conclusion..... | 11 |
| 6. References | 11 |

2. Abstract

The message signal $m(t) = 5\cos(200\pi t) + 10\cos(400\pi t)$ is transmitted by using the DSB-Large Carrier (LC)-AM modulator given below. The nonlinear device has the input-output relationship $z(t) = 60x(t) + x^2(t)$. The bandpass filter (BPF) has a gain of 1, $BW = 2B = 400 \text{ Hz}$, center frequency $f_c = 1 \text{ kHz}$.

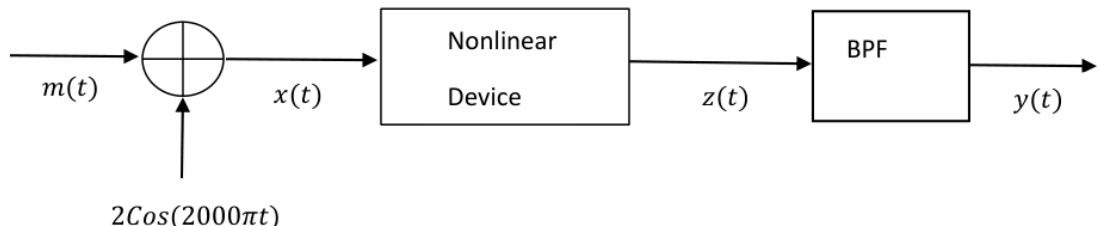


Figure 1:modulation of $m(t)$

Problems;

- Plotting $m(t)$, $x(t)$, $z(t)$, $y(t)$.
- Finding and plotting $M(f)$, $X(f)$, $Z(f)$, $Y(f)$. (the filtering in frequency domain.)
- Demodulating $y(t)$ by using envelope detector. Plotting the demodulated signal $\tilde{m}(t)$ and its spectra. (Using envelope command of MATLAB or the circuit of the envelope detector to demodulate)

3. Matlab Solution

3.a. Plotting $m(t)$, $x(t)$, $z(t)$

According to Figure 1, signals $m(t)$, $x(t)$, and $z(t)$ were defined and plotted using the plot function. The signal $y(t)$ was then plotted after obtaining it in the frequency domain. A sampling frequency of 4100 Hz was chosen. The graphs are displayed in Figure 2.

```

Fs = 4100; % Sampling frequency
t = 0:1/Fs:1-1/Fs; % Time vector
m=5*cos(2*pi*100*t)+10*cos(2*pi*200*t); %m(t)
x= m + 2*cos(2*pi*1000*t); %x(t)
z=60*x+ x.^2; %z(t)

figure(1)
subplot(3,1,1) % 3 rows, 1 column, first position
plot(t,m);
title('graphic of m(t) at time domain','muhammet koç 1901022017')
xlabel("t(s)")
ylabel("m(t)")

subplot(3,1,2) % 3 rows, 1 column, second position
plot(t,x)
title('graphic of x(t) at time domain')
xlabel("t(s)")
ylabel("x(t)")

subplot(3,1,3) % 3 rows, 1 column, third position
plot(t,z)
title('graphic of z(t) at time domain')
xlabel("t(s)")
ylabel("z(t)")

```

ELEC 361- ANALOG COMMUNICATIONS SYSTEMS
A MATLAB PROJECT REPORT: DSB-LC-AM

Muhammet Koç -1901022017

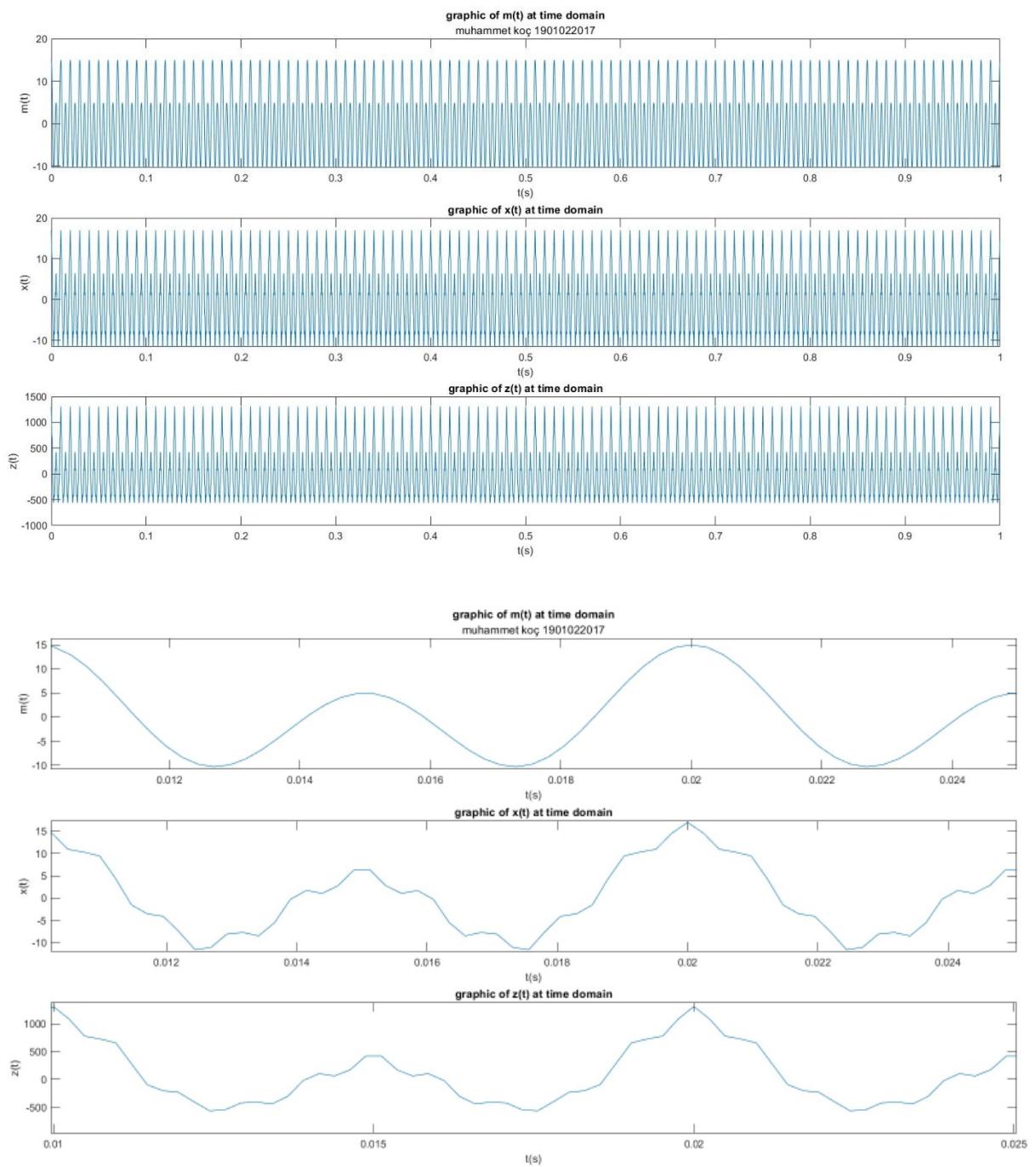


Figure 2: plots of $m(t)$, $x(t)$ and $z(t)$

3.b.1 Plotting $M(f)$, $X(f)$, $Z(f)$

Frequency domain values for the signals $m(t)$, $x(t)$, and $z(t)$ were obtained using the `fft()` function after obtaining them in the time domain. The resulting values were then shifted from the range $(0, T_0)$ to the range $(-T_0/2, T_0/2)$ using the `fftshift()` function. These shifted values are plotted using the `stem()` function, and the graphs are displayed in Figure 3

```

M=fft(m); % forier transform of m(t)
Mshift = fftshift(M); %fftshift() function allows us to bring the
center to zero
lm = length(M);
fm = (-lm/2:lm/2-1)/lm*Fs;

X=fft(x); % forier transform of x(t)
Xshift = fftshift(X);
lx = length(X);
fx = (-lx/2:lx/2-1)/lx*Fs;

Z=fft(z); % forier transform of z(t)
Zshift = fftshift(Z);
lz = length(Z);
fz = (-lz/2:lz/2-1)/lz*Fs;
figure(2)
subplot(3,1,1) % 3 rows, 1 column, first position
stem(fm,abs(Mshift)/Fs)
title('graphic of M(f) at frequency domain','muhammet koç 1901022017')
xlabel("Frequency (Hz)")
ylabel("|M(f)|")
grid
subplot(3,1,2) % 3 rows, 1 column, second position
stem(fx,abs(Xshift)/Fs)
title('graphic of X(f) at frequency domain','muhammet koç 1901022017')
xlabel("Frequency (Hz)")
ylabel "|X(f)|"
grid
subplot(3,1,3) % 3 rows, 1 column, third position
stem(fz,abs(Zshift)/Fs)
title('graphic of Z(f) at frequency domain','muhammet koç 1901022017')
xlabel("Frequency (Hz)")
ylabel "|Z(f)|"
grid

```

ELEC 361- ANALOG COMMUNICATIONS SYSTEMS
A MATLAB PROJECT REPORT: DSB-LC-AM

Muhammet Koç -1901022017

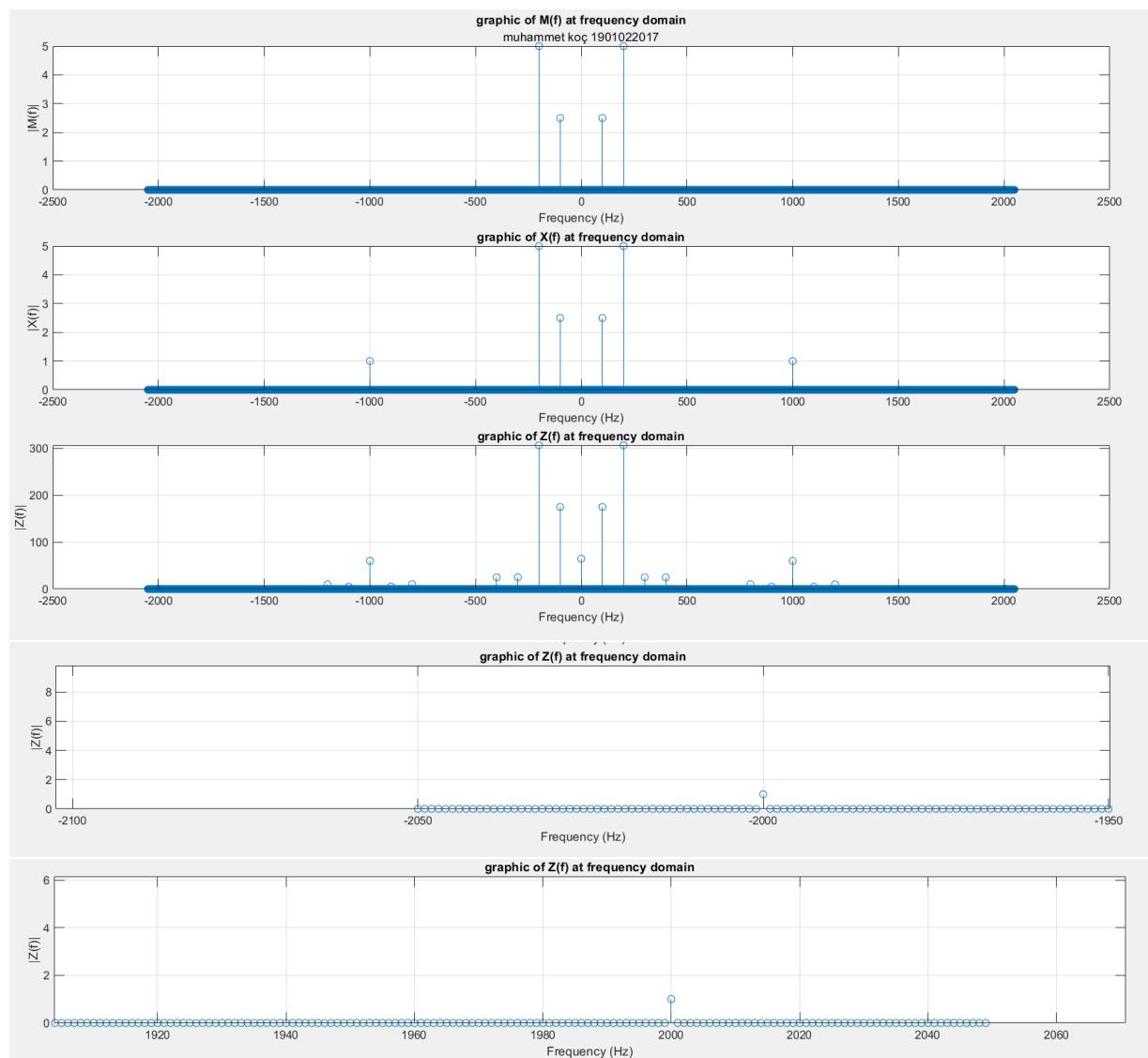


Figure 3:Plots $M(f)$, $X(f)$, $Z(f)$

3.b.2 Plotting $Y(f)$ and $y(t)$

To filter the $Z(f)$ signal obtained in the frequency domain, a for loop was used to filter the frequency values between -1200 and -800, as well as 800 and 1200. If-block values were created based on non-shifted indices. After filtering, $y(t)$ was obtained using the ifft() function, and the graph of $y(t)$ was drawn using the plot function. The graphs are shown in Figure 4.

```

Y=Z;
for i=1:1:4100 % band pass filter at frequency domain
    if i<801 ||( i>=1202 && i<2901) || i>=3302
        Y(i)=0*Z(i);
    end
end
Yshift = fftshift(Y);
y=ifft(Y); % inverse fourier transform
figure(3)
subplot(2,1,1) % 2 rows, 1 column, first position
stem(fz,abs(Yshift)/Fs)
title('graphic of Y(f) at frequency domain','muhammet koç 1901022017')
xlabel("Frequency (Hz)")
ylabel("|Y(f)|")
grid
subplot(2,1,2) % 2 rows, 1 column, second position
plot(t,y)
title('graphic of y(t) at time domain','muhammet koç 1901022017')
xlabel("t (s)")
ylabel("y(t)")
```

ELEC 361- ANALOG COMMUNICATIONS SYSTEMS
A MATLAB PROJECT REPORT: DSB-LC-AM

Muhammet Koç -1901022017

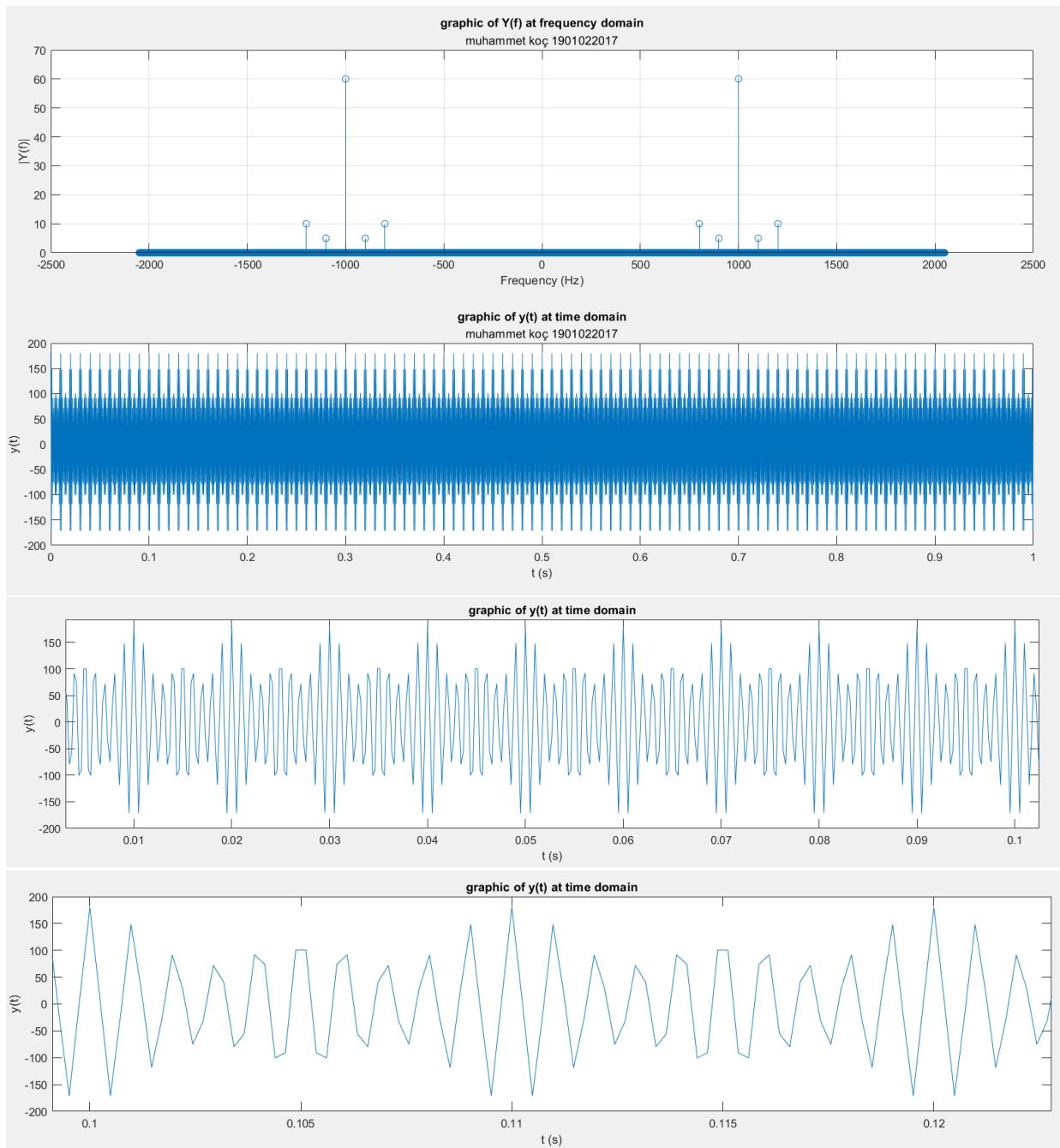


Figure 4: plots $Y(f)$ and $y(t)$

3.c. Demodulating $y(t)$ by using envelope detector

The envelope() function in Matlab was used to demodulate the $y(t)$ signal. The envelope() function returns the upper and lower envelopes of the signal. The signal $m(t)$ can be obtained by using the upper envelope, which is a signal with the form $a_1 + a_2m(t)$. In the analytical solution, it was determined that $a_1 = 120$ (DC offset) and $a_2 = 4$ (gain). To obtain the $m(t)$ signal, it is sufficient to zero the Dirac at $f = 0$ with the Fourier transform of the upper envelope and multiply it by $1/4$. Additionally, in the time domain, this is equivalent to subtracting the DC offset and multiplying it by $1/4$. The graphs are shown in Figure 5.

```

yupper=envelope(y); % envelope function allows us to get upper and lower
% envelope of signal
YUPPER=fft(yupper); % for dc filter at frequency domain
YUPPER(1)=0; % removed dc offset
MDEMOD=fftshift(YUPPER)
mdemod=(ifft(YUPPER)/4); %inverse fourier transform and gain is 1/4

figure(4)
subplot(3,1,1) % 3 rows, 1 column, first position
%easy way time domain filter after from envelope detector, (a1+a2m(t)=>
a1=120(dc offset), a2=4)
plot(t,(yupper-120))
title('graphic of (yupper(t)-120)= km(t),k=4 at time domain','muhammet koç
1901022017')

xlabel("t(s)")
ylabel("y(t)")

subplot(3,1,2) % 3 rows, 1 column, second position
plot(t,mdemod)
title('graphic of mdemod(t) filtered in frequency domain')
xlabel("t(s)")
ylabel("m(t)")

subplot(3,1,3) % 3 rows, 1 column, third position
stem(fm,abs(MDEMOD)/Fs)
title('graphic of MDEMOD(f) at frequency domain')
xlabel("Frequency (Hz)")
ylabel("|M'(f)|")

```

ELEC 361- ANALOG COMMUNICATIONS SYSTEMS
 A MATLAB PROJECT REPORT: DSB-LC-AM

Muhammet Koç -1901022017

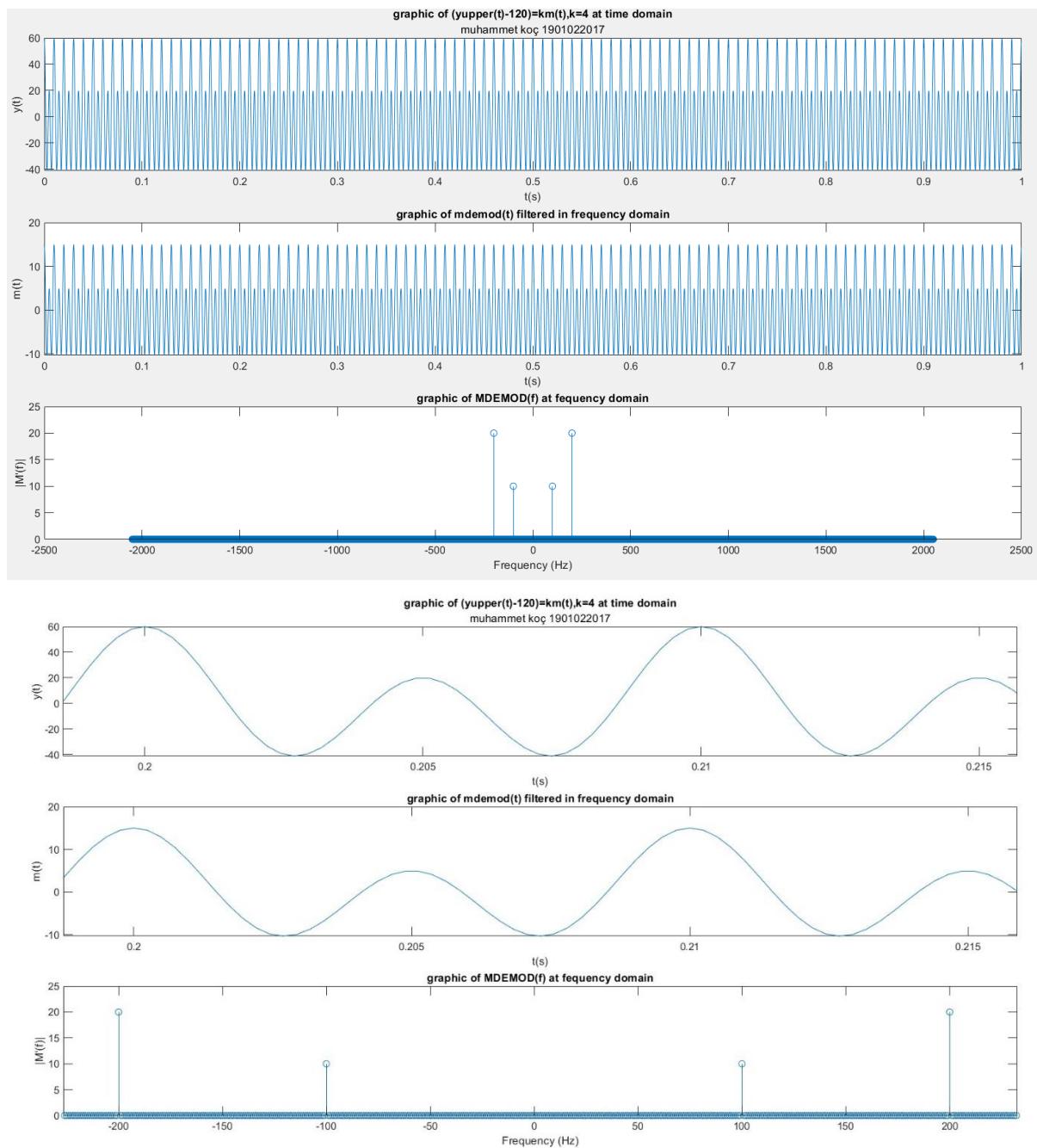


Figure 5: demodulated $m'(t)$ and $M'(f)$

4. Analytic Solution

Signals were obtained sequentially in both time and frequency domains, and those that can be drawn by hand were sketched.

5. Conclusion

As a result of this project, DSB-LC-AM has been realized in all its stages. We initially modulated our message signal and then demodulated it at the receiver to retrieve the original message signal. Consequently, our operations have led to the complete restoration of the message signal. When comparing the analytical solution with the MATLAB solution, it appears that the results align. It's worth noting that the analytical solution may not always be feasible, as demonstrated in this project. Conducting modulation and demodulation steps in a computer environment using analysis programs such as Matlab can save time and simplify complex projects.

6. References

1. Alan V. Oppenheim and Ronald W. Schafer, Discrete-Time Signal Processing, 2nd ed., Prentice-Hall, Upper Saddle River, New Jersey, 1998.
2. John G. Proakis & Masoud Salehi, Fundamentals of Communication Systems, 2nd ed., Prentice-Hall, Upper Saddle River
3. https://www.mathworks.com/help/matlab/ref/fft.html?s_tid=doc_ta#mw_b6c10d38-03f2-4c5e-80af-7a2930956b5b

GEBZE TECHNICAL UNIVERSITY

ELECTRONIC ENGINEERING DEPARTMENT

**ELEC 361
ANALOG COMMUNICATIONS SYSTEMS**

**THE MATLAB PROJECT REPORT:
Frequency Modulation and Demodulation**

MUHAMMET KOÇ

1901022017

1. Table of contents

Contents

| | |
|--|----|
| 1. Table of contents | 1 |
| 2. Abstract | 2 |
| 3. Matlab Solution | 3 |
| 3.a. Plotting $m(t)$ and $M(f)$ | 3 |
| 3.b. Finding and plotting $\phi(t)$ and $\phi(f)$ | 4 |
| 3.c Finding and plotting $y(t)$ and $Y(f)$ | 5 |
| 3.d. Demodulating $y(t)$ by using envelope detector step by step | 7 |
| 4. Analytic Solution | 14 |
| 5. Conclusion..... | 15 |
| 6. References | 15 |

2. Abstract

The message signal $m(t) = \Pi\left(\frac{t-1}{2}\right) - \Pi\left(\frac{t-3}{2}\right)$ t in seconds, modulates the carrier signal $c(t) = 5\cos(500\pi t)$ to obtain FM signal $y(t) = 5\cos(500\pi t + \phi(t))$, where $\phi(t) = 2\pi kf \int m(\tau)d\tau$ with $kf = 50$. The demodulator consists of a differentiator, a diode (assume ideal), a low pass filter (LPF) and a capacitor for DC blocking, successively.

Problems;

- a) Plotting $m(t)$ and $M(f)$
- b) Finding and plotting $\phi(t)$ and $\phi(f)$
- c) Finding and plotting $y(t)$ and $Y(f)$
- d) Obtaining and plotting the time-domain representation and magnitude spectrum of the signal at each step of the demodulator. (Do the filtering in frequency domain.)

3. Matlab Solution

3.a. Plotting $m(t)$ and $M(f)$

Signal $m(t) = \prod\left(\frac{t-1}{2}\right) - \prod\left(\frac{t-3}{2}\right)$ defined and plotted using the plot function. The signal $M(f)$ was then plotted after obtaining it in the frequency domain using stem function. A sampling frequency of 1000 Hz was chosen. The graphs are displayed in Figure 1.

```
% Generate a range of values for t from -1 to 5 with a step size of 1/Fs
Fs = 1000;
time = -5:1/Fs:5-1/Fs;

% Initialize an empty array to store the results of m(t)
mresult = zeros(size(time));

% Define the rectangular pulse for m(t)
m_pulse = @(t) 1 * (t >= 0 & t < 2) - 1 * (t >= 2 & t < 4);

% Evaluate the rectangular pulse for each value of t
for i = 1:length(time)
    mresult(i) = m_pulse(time(i));
end

% Compute the FFT of the mresult array
M = fft(mresult);
Mshift = fftshift(M); % fftshift() function centers the frequencies around
zero
lm = length(M);

% Generate the frequency axis in Hz for the original signal m(t)
fm = (-lm/2:lm/2-1)/lm*Fs;

% Plotting
figure(1)
subplot(2, 1, 1);
plot(time, mresult);
title('m(t)');
xlabel('Time (s)');
ylabel('Amplitude');

subplot(2, 1, 2);
stem(fm, abs(Mshift)/Fs);
title('FFT of m(t)');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
xlim([-50 50]);
```

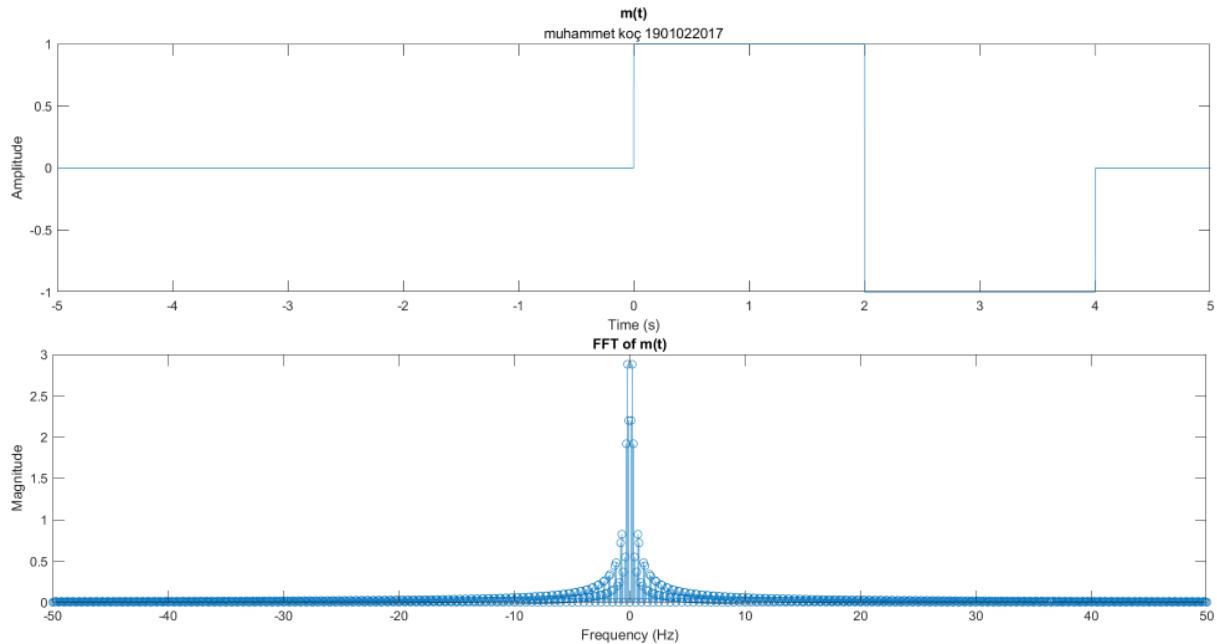


Figure 1: plots of $m(t)$ and $M(f)$

3.b. Finding and plotting $\phi(t)$ and $\phi(f)$

Signal $\phi(t) = 2\pi kf \int m(\tau)d\tau$ with $kf = 50$ defined and plotted using the plot function. Numerical calculations were tried to integrate the $m(t)$ function, but since the `cumtrapz()` function in Matlab was more reliable, so `cumtrapz()` was used. The signal $\phi(f)$ was then plotted after obtaining it in the frequency domain using `stem` function. A sampling frequency of 1000 Hz was chosen. The graphs are displayed in Figure 2.

```
% Calculate fi(t) as a separate function using trapezoidal cumulative
% sum (better than integrated_m)
fi_t = 2 * pi * kf * cumtrapz(time, mresult);
% Calculate the Fourier Transform of fi(t)
Fi_t = fft(fi_t);
Fi_t_shift = fftshift(Fi_t); % fftshift() function centers the frequencies around
zero
lf = length(Fi_t);
f_Fi_t = (-lf/2:lf/2-1)/lf*.2;

figure(2)
subplot(2, 1, 1);
plot(time, fi_t);
title('fi(t)', 'muhammet koç 1901022017');
xlabel('Time (s)');
ylabel('Phase');

subplot(2, 1, 2);
stem(f_Fi_t, abs(Fi_t_shift)/Fs);
title('Fourier Transform of fi(t)');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
```

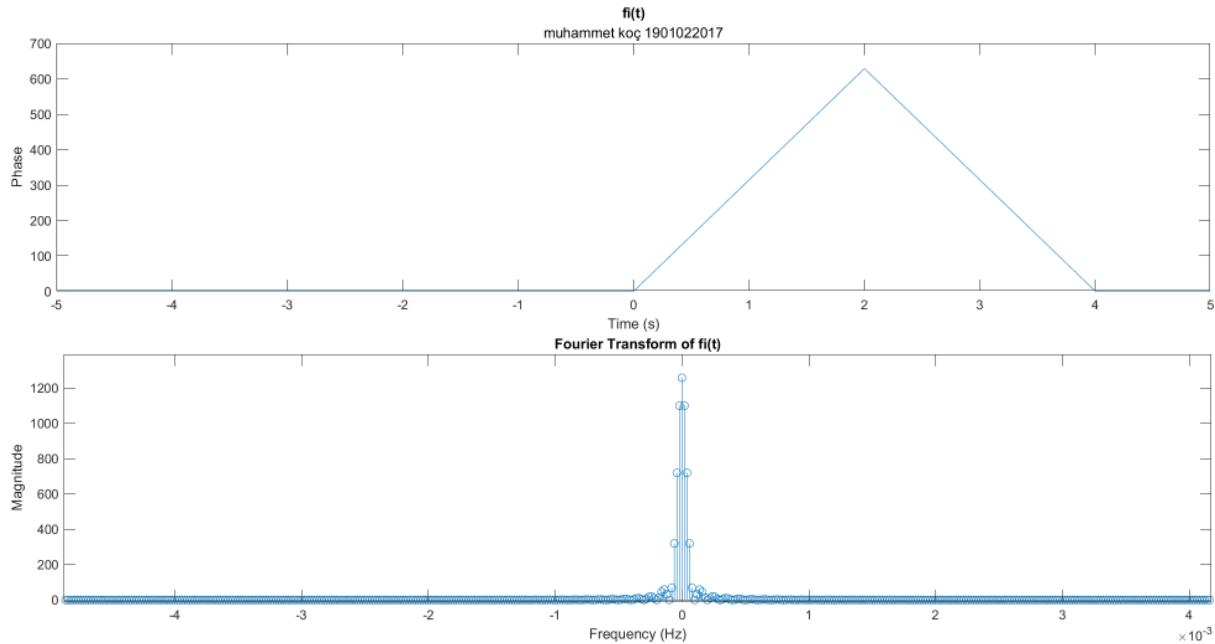


Figure 2: plotting $\phi(t)$ and $\phi(f)$

3.c Finding and plotting $y(t)$ and $Y(f)$

Signal $y(t) = 5\cos(500\pi t + \phi(t))$ defined and plotted using the plot function. The signal $Y(f)$ was then plotted after obtaining it in the frequency domain using stem function. A sampling frequency of 1000 Hz was chosen. The graphs are displayed in Figure 1.

```
% Set the carrier frequency for the frequency-modulated signal
carrier_frequency = 250;

% Compute the frequency-modulated signal y
y = 5 * cos(2 * pi * carrier_frequency * time + fi_t);

% Compute the FFT of the frequency-modulated signal
Y = fft(y);
Yshift = fftshift(Y); % fftshift() function centers the frequencies around
zero
ly = length(Y);
fy = (-ly/2:ly/2-1)/ly*Fs;

figure(3)
subplot(2, 1, 1);
plot(time, y);
title('Frequency-Modulated Signal y(t)', 'muhammet koç 1901022017');
xlabel('Time (s)');
ylabel('Amplitude');

subplot(2, 1, 2);
stem(fy, abs(Yshift)/Fs);
title('FFT of Frequency-Modulated Signal y(t)');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
```

ELEC 361- ANALOG COMMUNICATIONS SYSTEMS
THE MATLAB PROJECT REPORT: FM MODULATION AND DEMODULATION
Muhammet Koç -1901022017

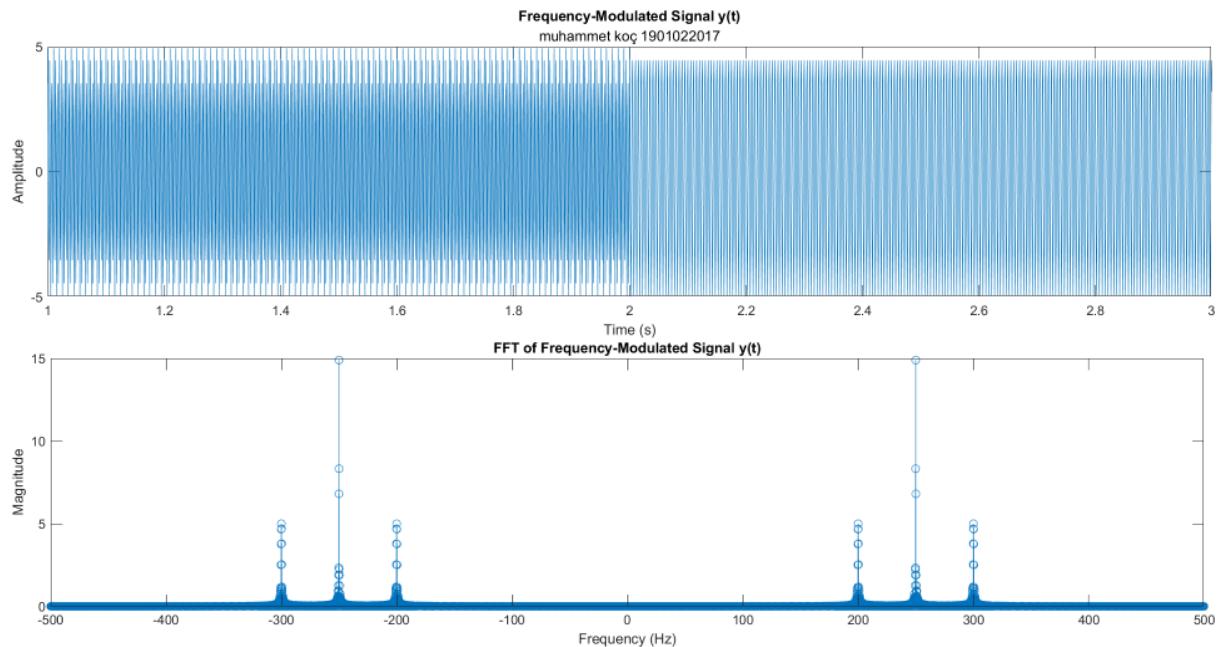


Figure 3: plots $Y(f)$ and $y(t)$

3.d. Demodulating $y(t)$ by using envelope detector step by step

Derivation, diode, RC filter and low-pass filter were used to demodulate the $y(t)$ signal.

3.d.1 Derivation of $y(t)$

Codes used to derive $y(t)$:

```
% Obtain the derivative of y and apply an ideal diode
ydiff = numerical_derivative(y,1/Fs);
Ydiff = fft(ydiff);
Ydiff_shift = fftshift(Ydiff);

figure(4);
subplot(2,1,1)
plot(time,ydiff)
title("differentiated y(t)",'muhammet koç 1901022017')
xlabel('Time (s)');
ylabel('Amplitude');

subplot(2, 1, 2);
stem(fy, abs(Ydiff_shift)/Fs);
title('FFT of Differentiated y(t)');
xlabel('Frequency (Hz)');
ylabel('Magnitude');

function dVin_dt = numerical_derivative(Vin, dt)
% Vin: Input voltage signal
% dt: Time step (inverse of the sampling frequency)

% Initialize the derivative vector
dVin_dt = zeros(size(Vin));

% Calculate the numerical derivative
for n = 1:length(Vin)-1
    dVin_dt(n) = (Vin(n+1) - Vin(n)) / dt;
end

% Use forward difference for the last point
dVin_dt(end) = dVin_dt(end-1);
end
```

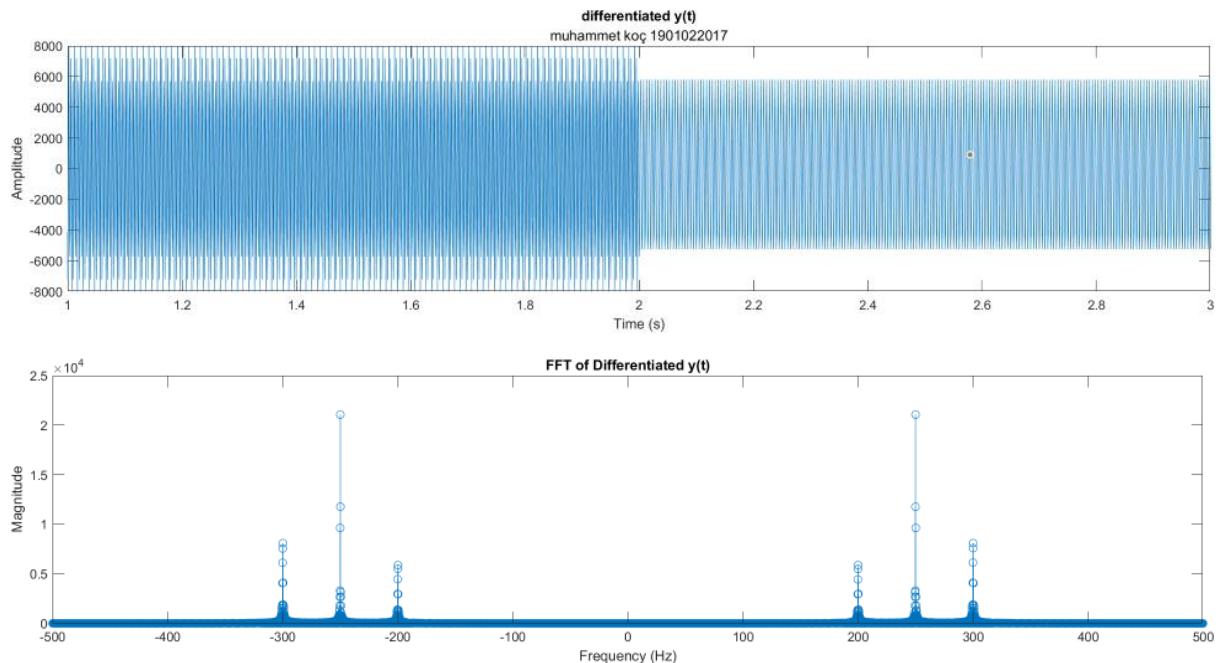


Figure 4: Derivation of $y(t)$ and $Y(f)$

3.d.2 Applying ideal Diode to $y(t)$

Codes used to apply to $y(t)$:

```
% Apply ideal diode the signal
ydiode = ideal_diode(ydiff);
Ydiode = fft(ydiode);
Ydiode_shift = fftshift(Ydiode);

figure(5);
subplot(2,1,1)
plot(time,ydiode)
title("Applied diode y(t)",'muhammet koç 1901022017')
xlabel('Time (s)');
ylabel('Amplitude');

subplot(2, 1, 2);
stem(fy, abs(Ydiode_shift)/Fs);
title('FFT of Applied diode y(t)');
xlabel('Frequency (Hz)');
ylabel('Magnitude');

function voltage= ideal_diode(V_in)
voltage=zeros(size(V_in));
for i=1:length(V_in)
    if 0< V_in(i)
        voltage(i)=V_in(i);
    end
end
end
```

ELEC 361- ANALOG COMMUNICATIONS SYSTEMS
THE MATLAB PROJECT REPORT: FM MODULATION AND DEMODULATION
Muhammet Koç -1901022017

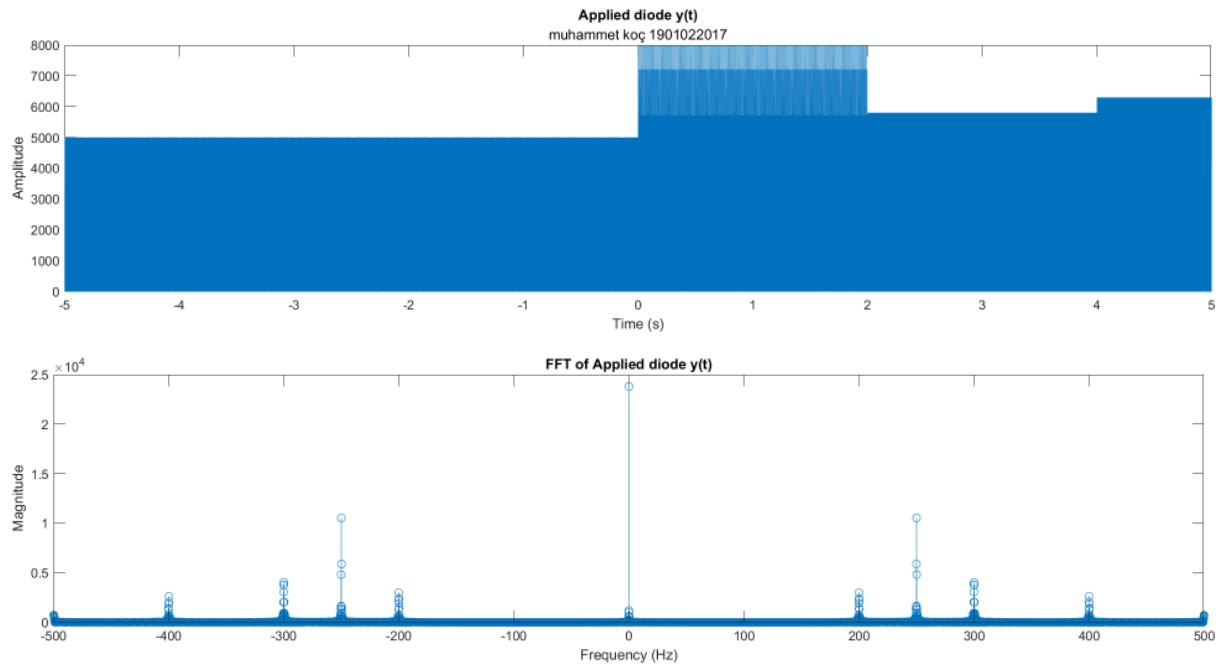


Figure 5: Apply diode to $y(t)$ and $Y(f)$

3.d.3 Applying RC filter to $y(t)$

Codes used to apply to $y(t)$:

```
% Define RC filter parameters
R = 10000; % Resistance in ohms
C = 10*10^-8; % Capacitance in farads

% Demodulate using an RC filter
mdemod = envelope_detector(ydiode, R, C, Fs);
Mdemod =fft(mdemod);
Mdemod_shift=fftshift(Mdemod);
figure(6);
subplot(2,1,1)
plot(time,mdemod)
title("Applied RC filter y(t)",'muhammet koç 1901022017')
xlabel('Time (s)');
ylabel('Amplitude');

subplot(2, 1, 2);
stem(fy, abs(Mdemod_shift)/Fs);
title('FFT of RC filter diode y(t)');
xlabel('Frequency (Hz)');
ylabel('Magnitude');

function Vout = envelope_detector(Vin, R, C, fs)
    % Vin: Modulated input signal
    % R: Resistance in ohms
    % C: Capacitance in farads
    % fs: Sampling frequency in Hz

    % Time constant
    tau = R * C;

    % Filter coefficients
    alpha = exp(-1/(tau * fs));

    % Initialize output voltage
    Vout = zeros(size(Vin));

    % Apply the envelope detector
    for n = 2:length(Vin)
        % Envelope detector equation
        Vout(n) = alpha * Vout(n-1) + (1 - alpha) * Vin(n);
    end
end
```

ELEC 361- ANALOG COMMUNICATIONS SYSTEMS
THE MATLAB PROJECT REPORT: FM MODULATION AND DEMODULATION
Muhammet Koç -1901022017

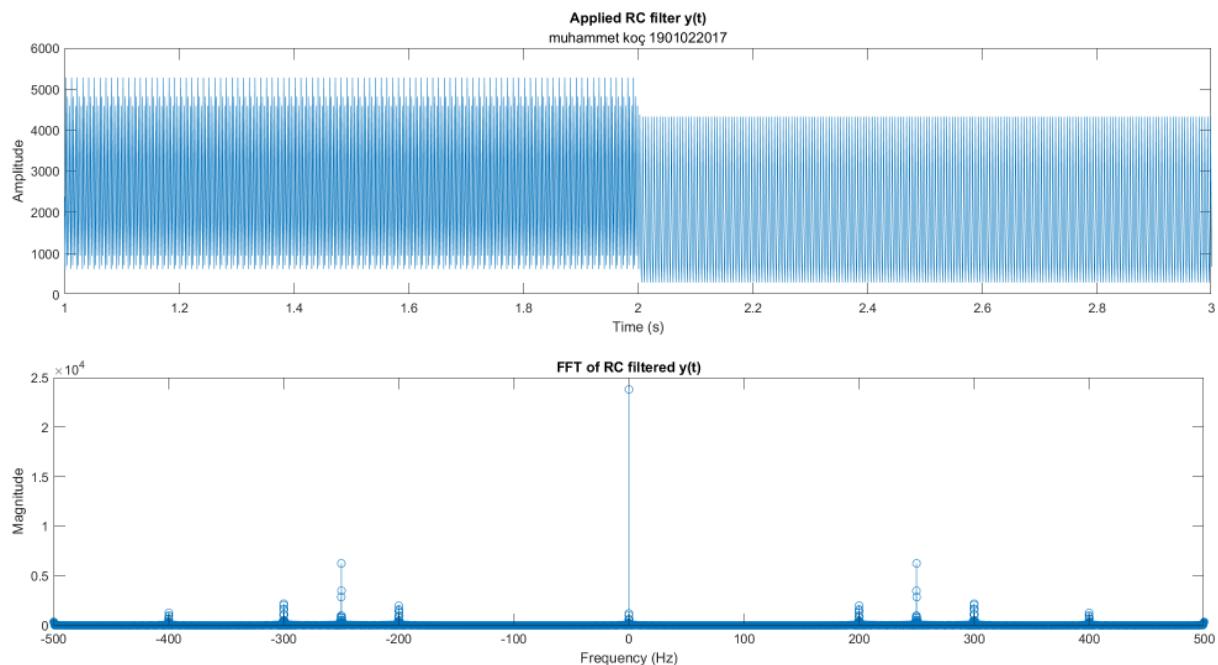


Figure 6: Apply RC filter to $y(t)$ and $Y(f)$

3.d.4 Dc blocking

Codes used to apply to $y(t)$:

```
% Compute the FFT of the demodulated signal
fftdemod = fft(mdemod);
fftdemod(1) = 0; % DC blocking
for i=1:1:10000 % LPF
    fftdemod(i)=fftdemod(i)/628;
    if(i>500 && i<9500)
        fftdemod(i)=0;
    end
end
mdemod_dc = ifft(fftdemod);
fftdemod_shift = fftshift(fftdemod); % fftshift() function centers the
frequencies around zero

figure(8);
subplot(2, 1, 1);
plot(fm, abs(Mshift)/Fs , 'r--');
title('FFT of m(t)', 'muhammet koç 1901022017');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
xlim([-20, 20]); % x-axis limit
ylim([-0.5 3.5]); % y-axis limit
hold on

plot(fm, abs(fftdemod_shift)/Fs, 'b-');
title('FFT of m(t) and FFT of mdemod(t)/628');
hold off
legend("M(f)", "Demodulated M(f)");
xlabel('Frequency (Hz)');
ylabel('Magnitude');

subplot(2, 1, 2);
plot(time, mresult , 'r--+');
title('FFT of m(t)');
xlim([-0.5 4.5]); % x-axis limit
ylim([-1.5 1.5]); % y-axis limit
hold on

plot(time, mdemod_dc, 'b-');
title('m(t) and mdemod(t)/628');
hold off
legend("m(t)", "Demodulated m(t)");
xlabel('Time (s)');
ylabel('Amplitude');
```

ELEC 361- ANALOG COMMUNICATIONS SYSTEMS
THE MATLAB PROJECT REPORT: FM MODULATION AND DEMODULATION
Muhammet Koç -1901022017

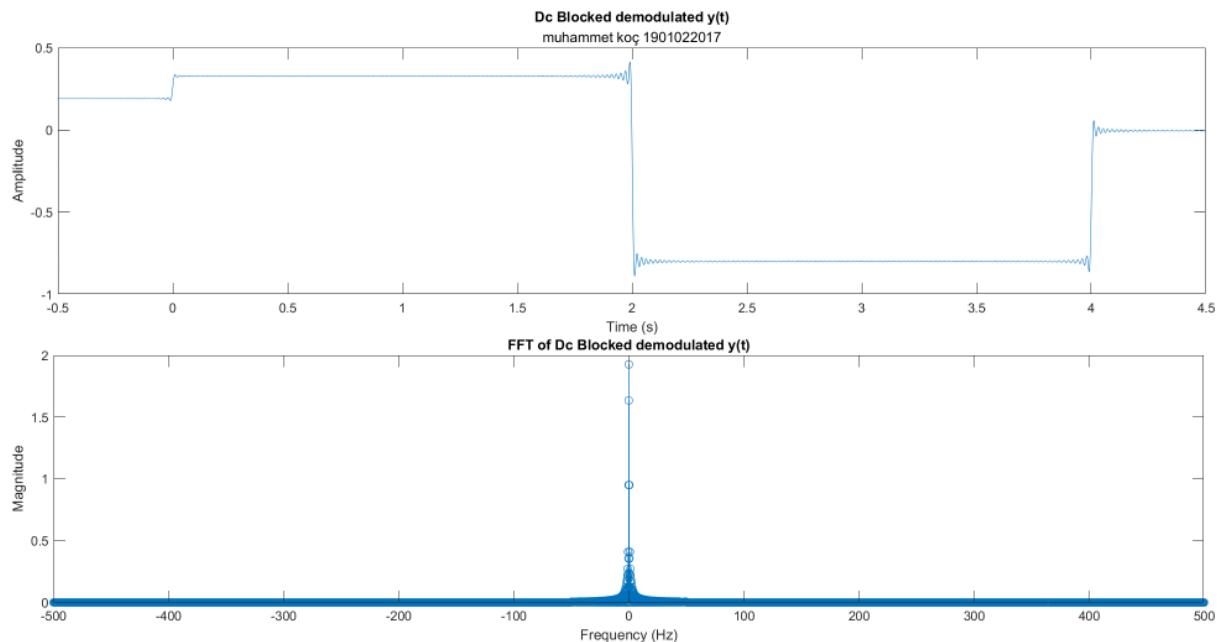


Figure 7: DC blocked demodulated $m'(t)$ and $M'(f)$

3.d.5 Comparing Results

Finally, the graphs obtained to compare the results obtained with the original plots $m(t)$ and $M(f)$ are given in figure 8.

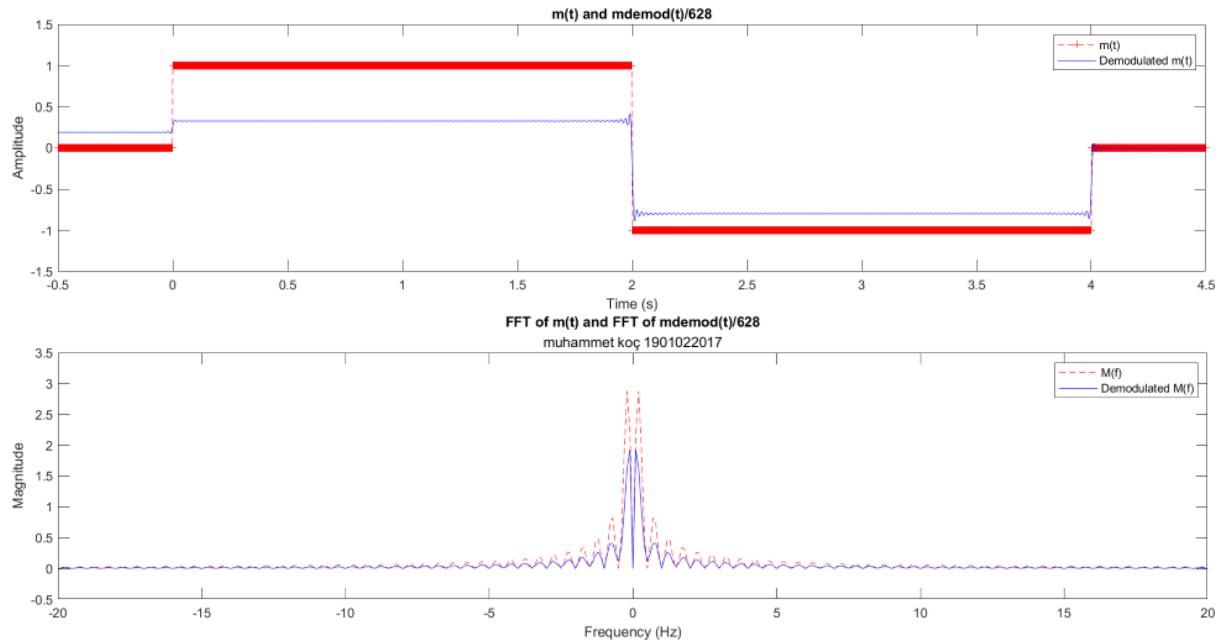


Figure 8: Comparing results

4. Analytic Solution

Signals were obtained sequentially in both time and frequency domains, and those that can be drawn by hand were sketched.

5. Conclusion

As a result of this project, FM modulation and demodulation have been implemented in all its stages. We initially modulated our message signal and then demodulated it at the receiver to get the original message signal. As a result, our operations did not achieve complete restoration of the message signal, but close results were achieved. When comparing the analytical solution with the MATLAB solution, the results appear to be compatible. It is worth noting that analytical solution may not always be possible, as demonstrated in this project. Performing modulation and demodulation steps on a computer using analysis programs such as Matlab can save time and simplify complex projects.

6. References

1. Alan V. Oppenheim and Ronald W. Schafer, Discrete-Time Signal Processing, 2nd ed., Prentice-Hall, Upper Saddle River, New Jersey, 1998.
2. John G. Proakis & Masoud Salehi, Fundamentals of Communication Systems, 2nd ed., Prentice-Hall, Upper Saddle River

ELM368 – DÖNEM PROJESİ

DRONE İLE ÇEKİLEN VİDEOLARDA KANAT SESİNİ AYIKLAMA

Elif Ergür, Muhammet Koç, Duygu Zeynep Tuncel
200102002024, 1901022017, 200102002082
e.ergur2020@gtu.edu.tr, muhammetkoc2019@gtu.edu.tr, d.tuncel2020@gtu.edu.tr

Projenin Amacı

Günümüzde drone teknolojisi, hava fotoğrafçılığı, haritalama ve gözlem gibi çeşitli alanlarda geniş çapta kullanılmaktadır. Ancak, döner kanat drone'ların çalışma sırasında çikardıkları yüksek ses, özellikle video çekimlerinde istenmeyen bir gürültü kaynağı oluşturur. Bu gürültü, genellikle drone'un pervanelerinin dönme hareketinden kaynaklanan kanat sesidir ve çekilen videoların kalitesini düşürerek izleyici deneyimini olumsuz etkiler.

Bu projenin amacı, drone ile çekilen videolarda yer alan kanat sesini dijital sinyal işleme (DSP) teknikleri kullanarak ayıklamaktır. Bu sayede, videoların ses kalitesi artırılacak ve izleyicilere daha temiz ve net bir ses deneyimi sunulacaktır. Projede, farklı filtreleme yöntemleri ve algoritmalar kullanılarak kanat sesinin ayıklanması hedeflenmektedir. Bu amaçla, frekans alanında analiz ve filtreleme teknikleri uygulanarak pervane sesinin belirlenmesi ve elimine edilmesi sağlanacaktır.

Sonuç olarak, bu proje ile drone videolarının kalitesinin artırılması ve daha profesyonel ses kayıtlarının elde edilmesi amaçlanmaktadır. Bu teknoloji, özellikle film yapımı, gazetecilik, doğa belgeselleri ve spor etkinlikleri gibi alanlarda önemli bir değer katacaktır.

ANAHTAR KELİMELER

Gürültü ayıklama, ses işleme, dijital filtreleme, genlik spektrumu, FFT (Fast Fourier Transform), FIR filtre.

1. Giriş

Drone ile çekilen videolarda pervane sesini bastırarak konuşma sesini netleştirebilme amacıyla çeşitli filtreler tasarlanmıştır. Bu çalışmayı gerçekleştirmek için, ses kayıtları "Audacity" programı yardımıyla FPV Drone'un

pervane sesi ile insan konuşma sesi birleştirilerek gerçek video çekimlerinde karşılaşılabilcek türden senaryolar oluşturulmuştur. Kayıtlar .wav formatında elde edilmiştir. Bu formatın tercih edilmesinin sebebi, Python dili ile bu formattaki dosyaları rahatça işleyebilmektir.

Elde edilen dosyalar Python'da okunarak, örnekleme frekansı ve ses dosyasına ait diğer bilgiler elde edilmiştir. Bu verilerin frekans cevapları analiz edilerek yorumlanmıştır. Yapılan yorumlar sonucunda, uygun filtreler tasarlanmış ve ses işaretleri filtrelenmiştir. Filtreleme işlemi sonucunda elde edilen ses işaretleri ile orijinal ses işaretleri Python üzerinde dinlenerek, aradaki farklar duyulabilir şekilde ortaya konmuştur.

2. Deneyler ve Analiz

Bu çalışmada, pervane sesi içeren ses kayıtlarını elde etmek için Audacity programı kullanılmıştır. Farklı şiddetlerde gürültüye sahip üç ses kaydı elde edilmiştir.

Elde edilen ses verileri, Python'da wavfile modülü kullanılarak okunmuştur. Bu modül ile seslerin uzunluğu, kanal sayısı ve örnekleme frekansı gibi bilgiler elde edilmiştir. Veriler, hem zaman hem de frekans uzayında Python ile görselleştirilmiştir ve genlik spektrumları çıkarılmıştır. Frekans uzayındaki analizler, Fourier dönüşümleri kullanılarak yapılmıştır.

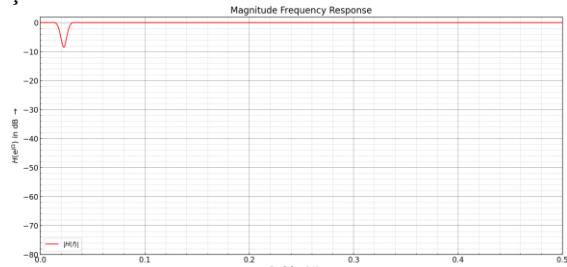
Bu veriler işliğinde, farklı kesim frekanslarına sahip filtreler Pyfda yazılımı kullanılarak tasarlanmıştır. Filtrelerin Python ortamına aktarılması için laboratuvara daha önce kullanılan kodlar fonksiyon haline getirilmiştir. Filtreler Python'a eklendikten sonra, üretilmiş ses işaretleri scipy kütüphanesinden lfilter fonksiyonu kullanılarak filtrelenmiş ve yeni ses işaretleri elde edilmiştir. Orijinal ve filtrelenmiş ses işaretleri, Audio kütüphanesinin display fonksiyonu ile Jupyter Notebook üzerinde dinlenerek aralarındaki fark değerlendirilmiştir.

Sonuçların Analizi:

- Filtrelenmiş ses işaretlerinde, gürültü işaretleri tamamen ayıklanamasa da belirli bir ölçüde ayıklama işlemi gerçekleştirilmiştir.
- 0.18π Yüksek geçiren filtre kullanıldığında, ses işaretinin kaybolduğu gözlemlenmiştir.
- Band geçiren ve alçak geçiren FIR filtreler kıyaslandığında, bu problem için en uygun filtre tipinin alçak geçiren FIR filtre olduğu anlaşılmıştır.

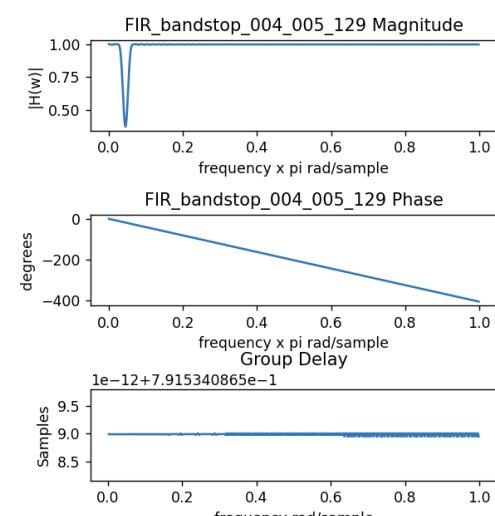
2.1 Filtre Tasarımları ve Filtrelenmiş İşaretler

Tasarlanan her filtre 6 ses işareti üzerinde de denenmiştir. Bazı filtreler bazı ses işaretlerinde iyi sonuç verirken bazı ses işaretlerinde kötü sonuçlar vermiştir. Tasarlanan filtreler şu şekildedir:



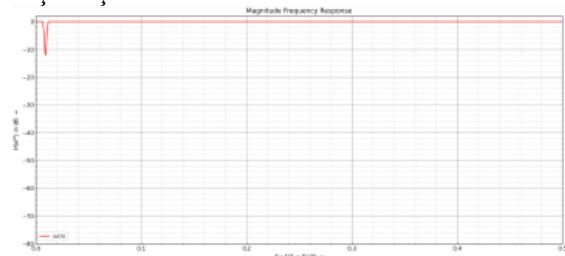
Şekil 1 Band söndüren hamming pencereli filtre

Şekil 1 de hamming penceresiyle üretilmiş 0.003π ile 0.004π kesim frekansı aralığına sahip bant söndüren filtre görülmektedir. Filtrenin derecesi 259 olarak alınmıştır.



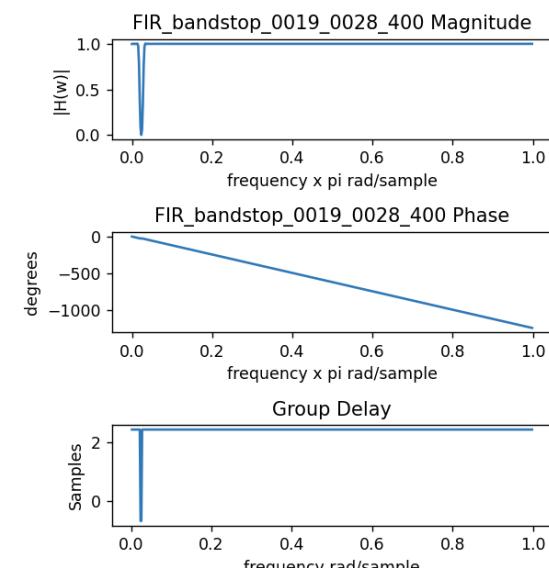
Şekil 1 Band söndüren hamming pencereli filtrenin genlik cevabı, faz cevabı ve grup gecikmesi

Şekil 2 de ise şekil 1 de Pyfda yazılımında genlik cevabı görülen filtrenin python ortamında genlik cevabı faz cevabı ve grup gecikmesi görülmektedir. Bu filtrenin faz cevabı lineer olduğu için grup gecikmesi de sabit çıkmıştır. Filtreleme derecesinden dolayı ani pik yaptığı nokta dışında faz cevabı lineer olduğu için grup gecikmesi de sabittir ve filtre çıkışı bozulmasız olmaktadır. Grup gecikmesinin sabit olması FIR filtrelerde dar bandlı keskin geçişlere sahip filtreler rahatlıkla oluşturulabilmştir. Bundan dolayı proje de kullanılan filtrelerin çoğu FIR olarak seçilmiştir.



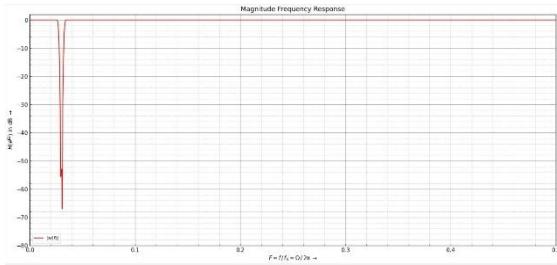
Şekil 2 Bant söndüren hamming pencereli filtre

Şekil 3 de hamming penceresiyle üretilmiş 0.0019π ile 0.0028π kesim frekansı aralığına sahip bant söndüren filtre görülmektedir. Filtrenin derecesi yüksek alınarak($N=801$) ihtiyaca yakın bir şekilde elde edilmiştir.



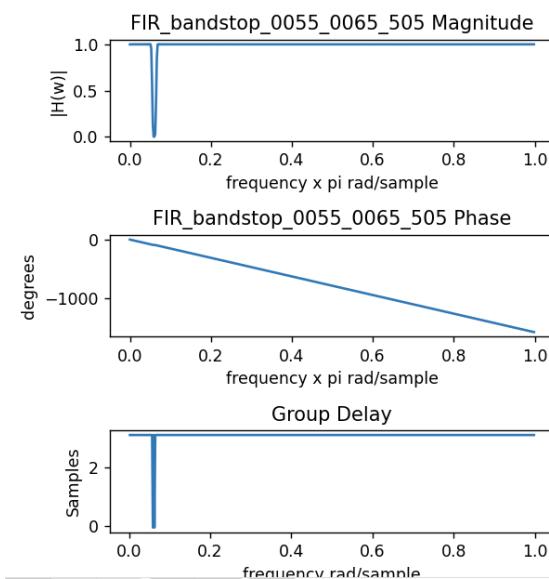
Şekil 3 Bant söndüren hamming pencereli filtrenin genlik cevabı, faz cevabı ve grup gecikmesi

Şekil 4 te hamming pencereli filtrenin genlik cevabı, faz cevabı ve grup gecikmesi görülmektedir. Bu filtre ses işaretlerinin frekans spektrumlarındaki 0.0019π ile 0.0028π kısmını elimine eder.



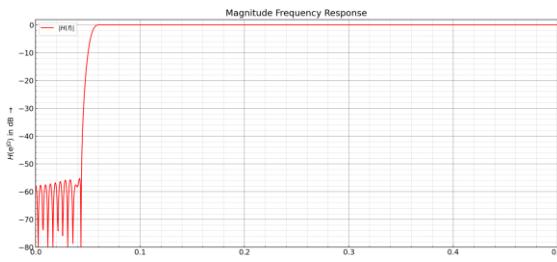
Şekil 4 Band sönüren hamming pencereli filtre

Şekil 5 te hamming pencereli band sönüren filtrenin genlik cevabı görülmektedir. Kesim frekansı 0.0055π ile 0.0065π aralığındadır. Filtrenin derecesi ise $N=1011$ olarak alınmıştır.



Şekil 5 Band sönüren hamming pencereli filtrenin genlik cevabı, faz cevabı ve grup gecikmesi

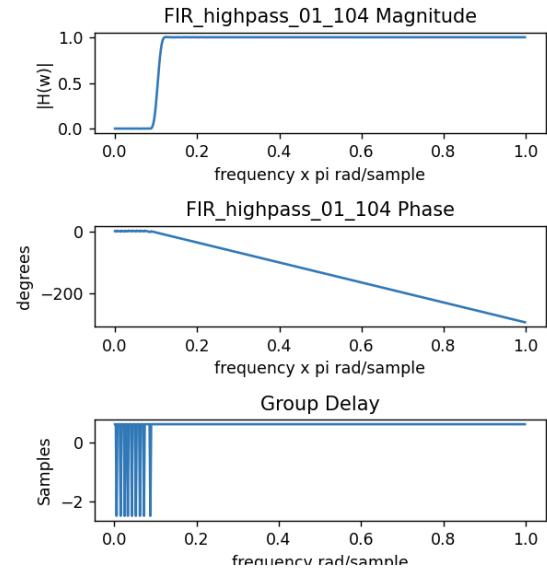
Şekil 6 da filtrenin Python ortamında genlik cevabı, faz cevabı ve grup gecikmesi görülmektedir. frekansı 0.0055π ile 0.0065π aralığında olan frekansları zayıflatmıştır.



Şekil 6 Yüksek Geçiren Hann Pencereli Filtre

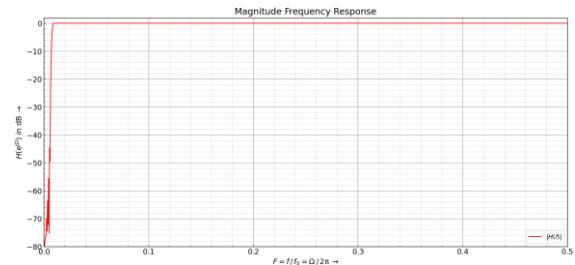
Şekil 7'de hamming pencereli yüksek geçirilen filtrenin genlik cevabı görülmektedir. Kesim

frekansı 0.01π dir. Filtrenin derecesi ise $N=209$ olarak alınmıştır.



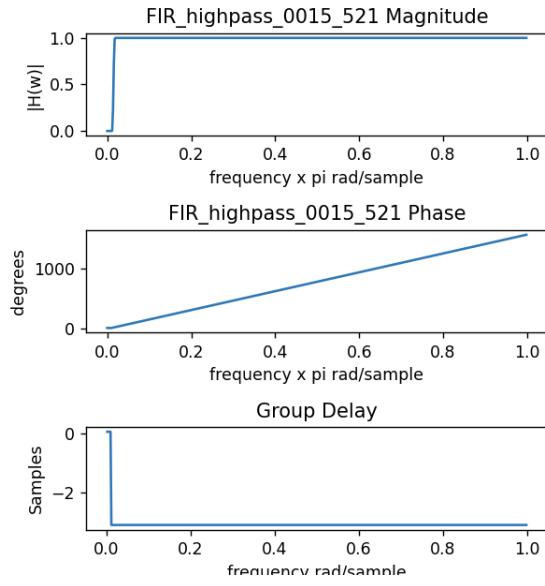
Şekil 7 Yüksek Geçiren Hamming Pencereli Filtrenin Genlik Cevabı, Faz Cevabı ve Grup Gecikmesi

Şekil 8 de filtrenin Python ortamında genlik cevabı, faz cevabı ve grup gecikmesi görülmektedir. Kesim frekansı olan 0.01π den sonraki frekansları geçirmiştir.



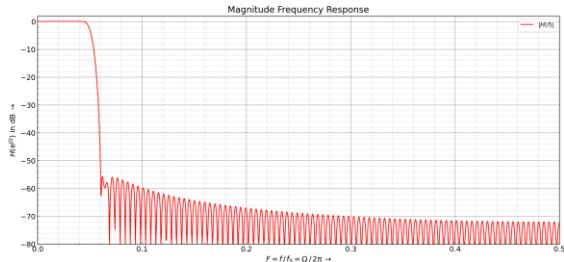
Şekil 8 Yüksek geçirilen hamming pencereli filtre

Şekil 9 te hamming pencereli yüksek geçirilen filtrenin genlik cevabı görülmektedir. Kesim frekansı 0.0015π dir. Filtrenin derecesi ise $N=1045$ olarak alınmıştır.



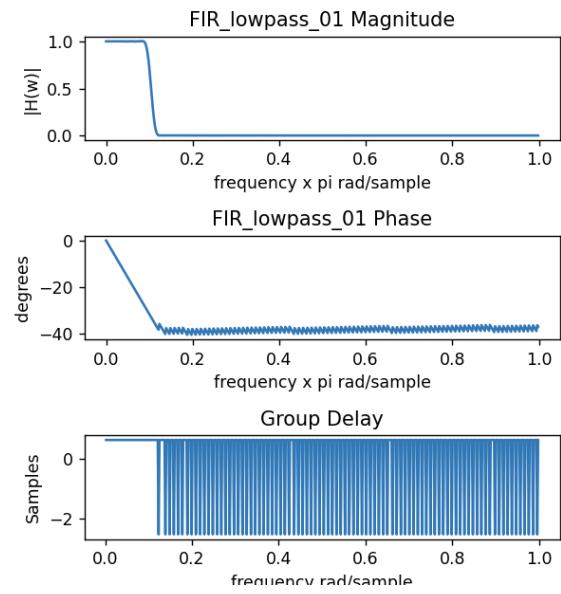
Şekil 10 Yüksek geçiren hamming pencereli Filtrenin Genlik Cevabı, Faz Cevabı ve Grup Gecikmesi

Şekil 10 de filtrenin Python ortamında genlik cevabı, faz cevabı ve grup gecikmesi görülmektedir. Kesim frekansı olan 0.0015π den sonraki frekansları geçirmiştir.



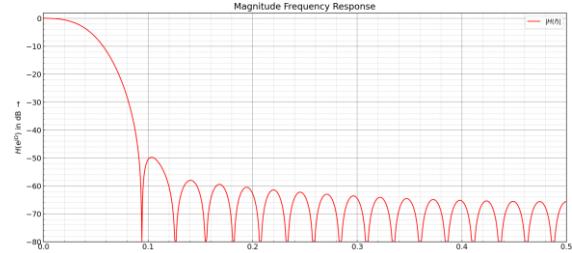
Şekil 11 Alçak geçiren hamming pencereli filtre

Şekil 11 te hamming pencereli alçak geçiren filtrenin genlik cevabı görülmektedir. Kesim frekansı 0.01π dir. Filtrenin derecesi ise $N=300$ olarak alınmıştır.



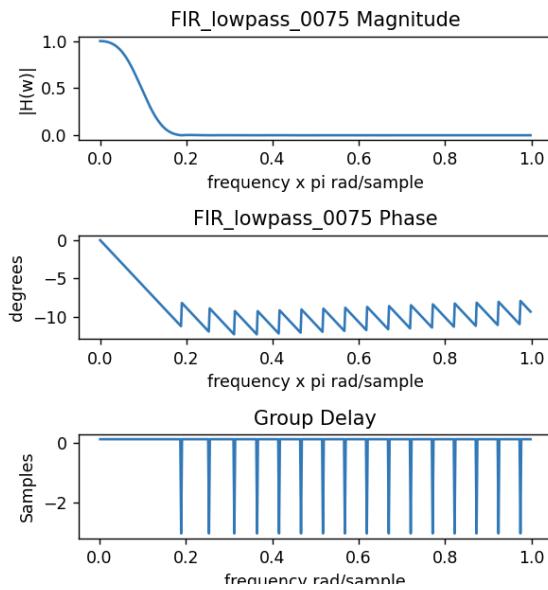
Şekil 12 Alçak geçiren hamming pencereli Filtrenin Genlik Cevabı , Faz Cevabı ve Grup Gecikmesi

Şekil 12 de hamming pencereli filtrenin genlik cevabı, faz cevabı ve grup gecikmesi görülmektedir. Bu filtre ses işaretlerinin frekans spektrumlarındaki 0.01π kısmı sonrası elimine eder.



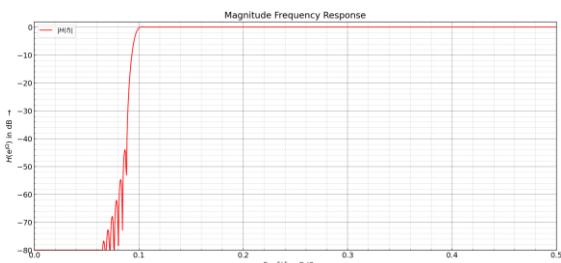
Şekil 13 Alçak geçiren hamming pencereli filtre

Şekil 13 de hamming pencereli alçak geçiren filtrenin genlik cevabı görülmektedir. Kesim frekansı 0.0075π dir. Filtrenin derecesi ise $N=39$ olarak alınmıştır.



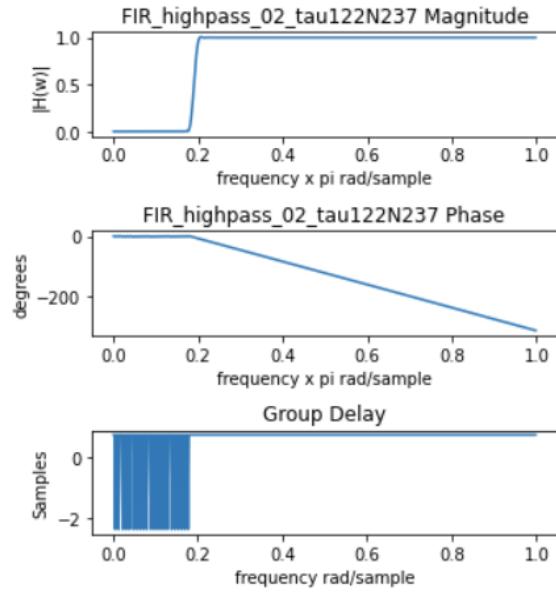
Şekil 14 Alçak geçiren hamming pencereli Filtrenin Genlik Cevabı , Faz Cevabı ve Grup Gecikmesi

Şekil 14 de hamming pencereli filtrenin genlik cevabı, faz cevabı ve grup gecikmesi görülmektedir. Bu filtre ses işaretlerinin frekans spektrumlarındaki 0.0075π kısmını sonrası elimine eder.



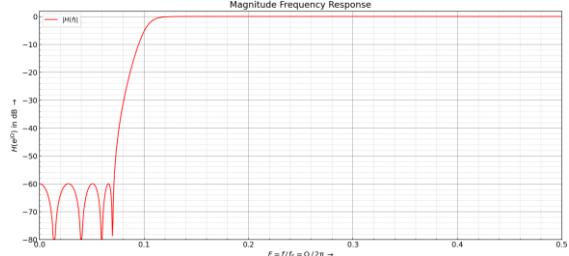
Şekil 15 Yüksek Geçiren Hann Pencereli Filtre

Şekil 15'te hann pencereli yüksek geçirilen filtrenin genlik cevabı görülmektedir. Kesim frekansı 0.18π dir. Filtrenin derecesi ise $N=237$ olarak alınmıştır.



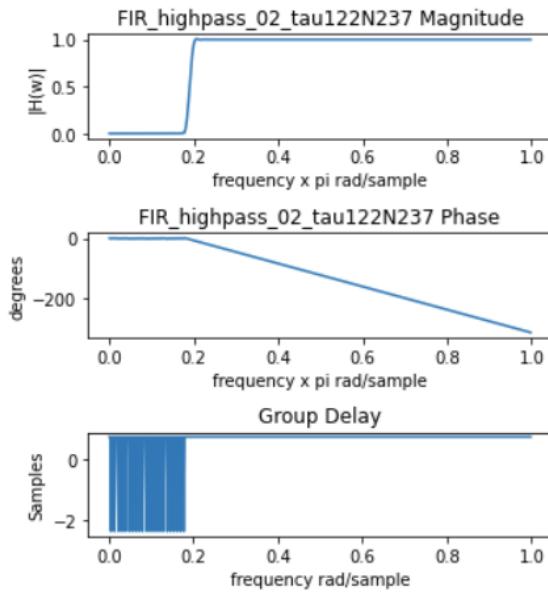
Şekil 16 Yüksek Geçiren Hann Pencereli Filtrenin Genlik Cevabı, Faz Cevabı ve Grup Gecikmesi

Şekil 16 da filtrenin Python ortamında genlik cevabı, faz cevabı ve grup gecikmesi görülmektedir. Kesim frekansı olan 0.02π den sonraki frekansları geçirmiştir.



Şekil 17 Yüksek Geçiren Chebshev Pencereli Filtre

Şekil 17'te chebshev pencereli yüksek geçirilen filtrenin genlik cevabı görülmektedir. Kesim frekansı 0.02π dir. Filtrenin derecesi ise $N=202$ olarak alınmıştır.



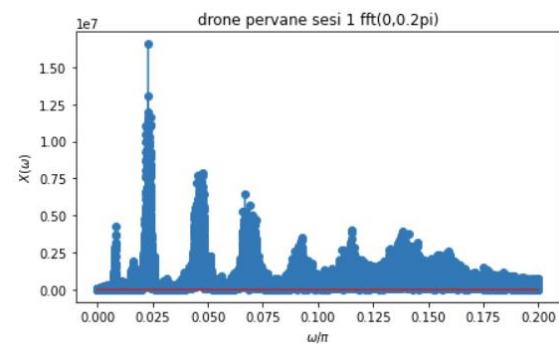
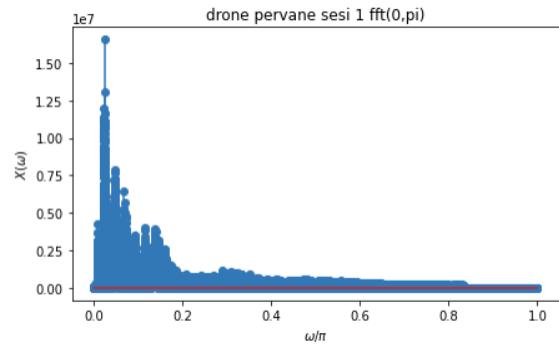
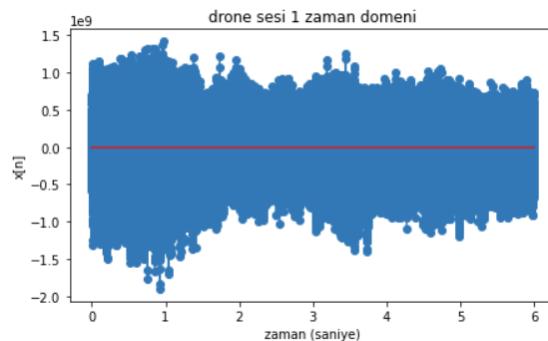
Şekil 18 Yüksek Geçiren Chebshev Pencereli Filtrenin Genlik Cevabı, Faz Cevabı ve Grup Gecikmesi

Şekil 18'de filtrenin Python ortamında genlik cevabı, faz cevabı ve grup gecikmesi görülmektedir. Kesim frekansı olan 0.02π den sonraki frekansları geçirmiştir.

2.2.1 Eğitim Sesi İşaretleri

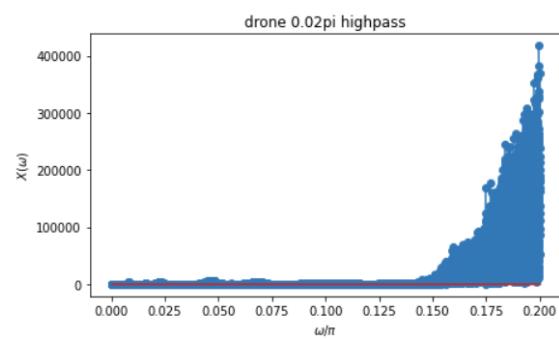
Bu kısımda ses işaretlerinin frekans uzayında grafiklerini incelenecek ve elde edilen bu grafiklere bakarak hangi filtrenin gürültülerini bastırmak için daha uygun olacağını yorumlanacak.

2.2.1.1 Ses-1 (djiavata.wav)



Şekil 19 Ses-1'in Sirayla Genlik-Zaman, Genlik-Frekans($0,\pi$), Genlik-Frekans($0,0.2\pi$), Güç Yoğunluk Fonksiyonu Grafiği Verilmiştir

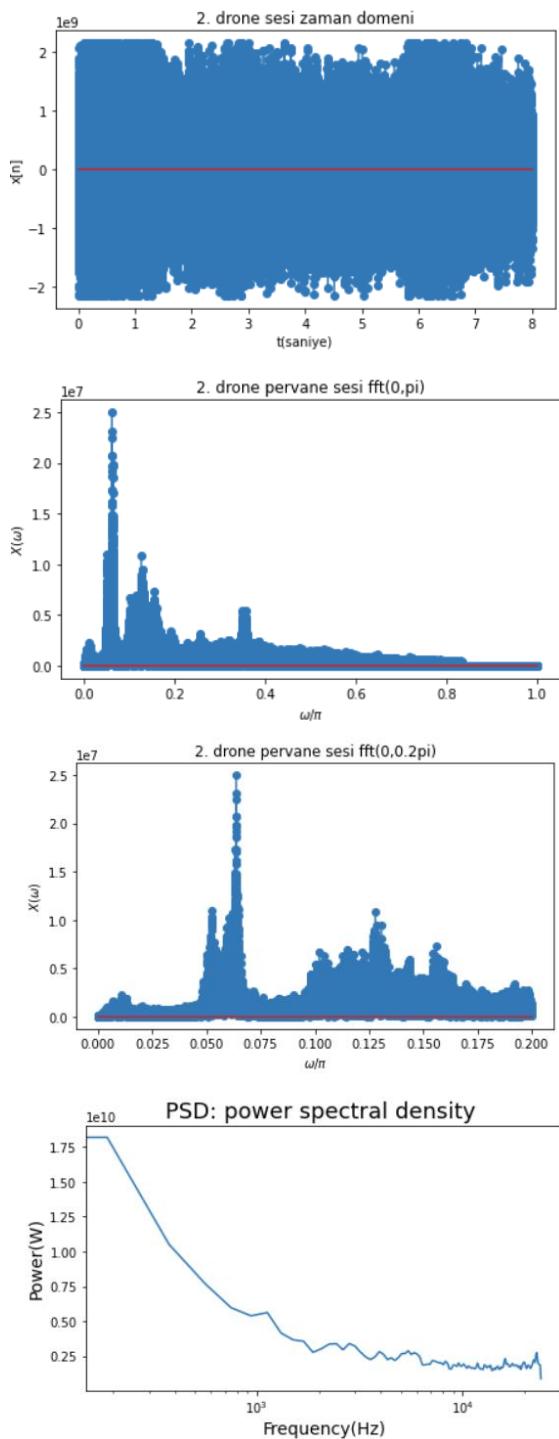
Şekil-19'da Ses işaretine ait genlik spektrumu ve Güç Yoğunluk Fonksiyonu görülmektedir. Genlik Spektrumunda görüleceği üzere drone ses işaretinin büyük çoğunluğu $0-0.02\pi$ arası yoğundur analiz aralığını test etmek için 0.02π frekansından sonrası geçirecek bir yüksek geçen filtre kullanmak gereklidir. Filtreden sonra bu aralıktan sonrasıının önemli olmadığını karar verilmiştir.



Şekil 20 Ses-1'in IIRhighpassfilterchebshev202 filtresinden geçirilmiş hali.

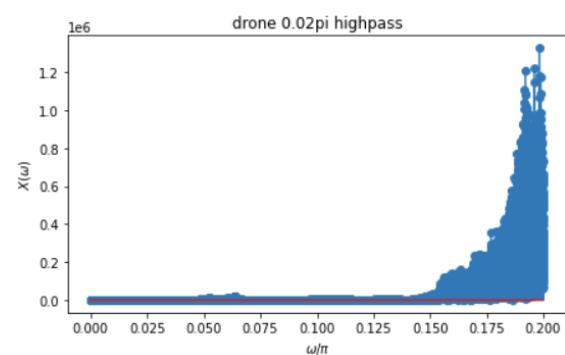
Ses-1 farklı filtreler ile denenmiştir ve en verimli çıktı Şekil-19'da $w=0-0.2\pi$ aralığına bakarak kesim frekansları $w=0.019-0.028\pi$ ve $w=0.04-0.05\pi$ bandsöndüren, $w=0.015$ yüksek geçiren滤re ve son olarak $w=0.075\pi$ düşük geçiren filtreden geçirilmesi planlanmıştır

2.2.1.2 Ses-2 (djifpv.wav)



Şekil 21 Ses-2'in Sırayla Genlik-Zaman, Genlik-Frekans($0,\pi$), Genlik-Frekans($0,0.2\pi$), Güç Yoğunluk Fonksiyonu Grafiği Verilmiştir

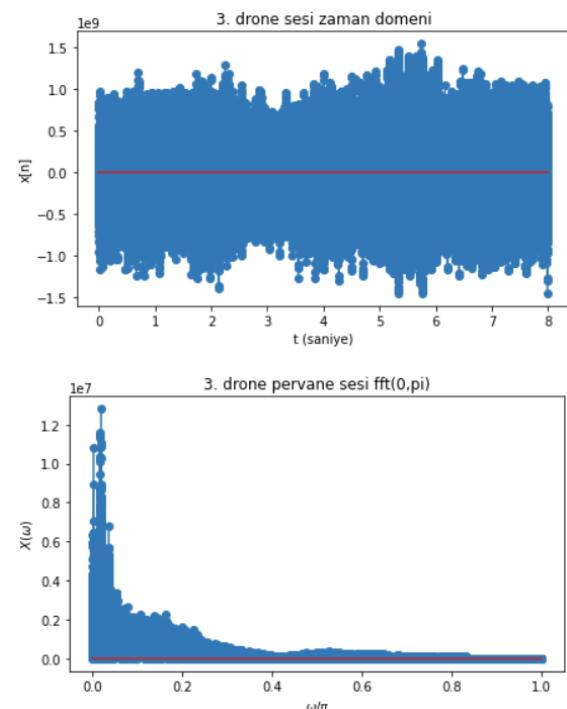
Şekil-21'de Ses işaretine ait genlik spektrumu ve Güç Yoğunluk Fonksiyonu görülmektedir. Genlik Spektrumunda görüleceği üzere drone ses işaretinin büyük çoğunluğu $0-0.02\pi$ arasında yoğundur analiz aralığını test etmek için 0.02π frekansından sonrası geçirecek bir yüksek geçiren滤re kullanmak gereklidir. Filtreden sonra bu aralıktan sonrasıının önemli olmadığını karar verilmiştir.

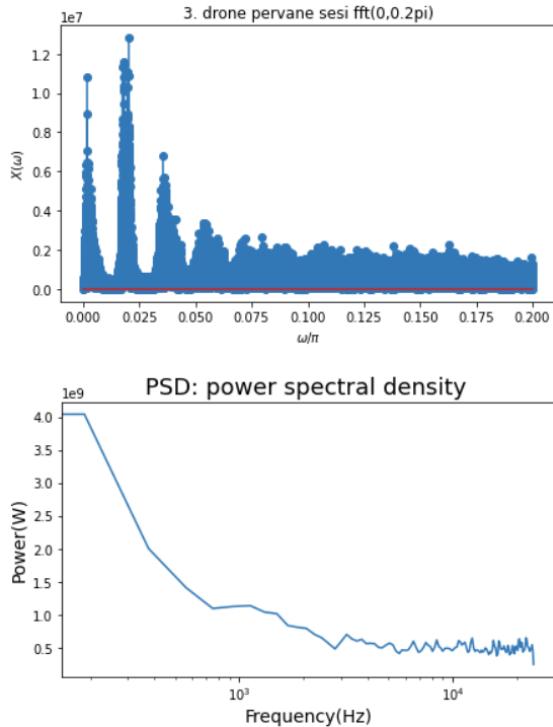


Şekil 22 Ses-1'in I IIRhighpassfilterchebshev202 filtresinden geçirilmiş hali.

Ses-2 farklı filtreler ile denenmiştir ve en verimli çıktı Şekil-21'de $w=0-0.2\pi$ aralığına bakarak kesim frekansları $w=0.055-0.065\pi$ bandsöndüren ve $w=0.075\pi$ düşük geçiren filtreden geçirilmesi planlanmıştır

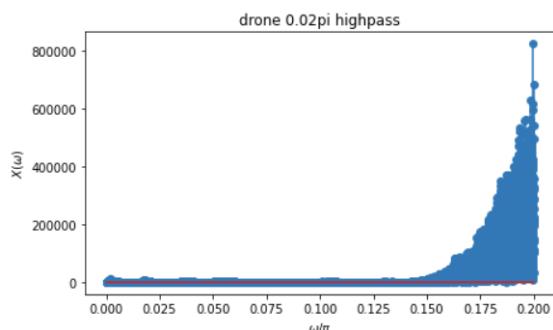
2.2.1.3 Ses-3 (fpvcinema.wav)





Şekil 23 Ses-3’ün Sirayla Genlik-Zaman, Genlik-Frekans($0,\pi$), Genlik-Frekans($0,2\pi$), Güç Yoğunluk Fonksiyonu Grafiği Verilmiştir

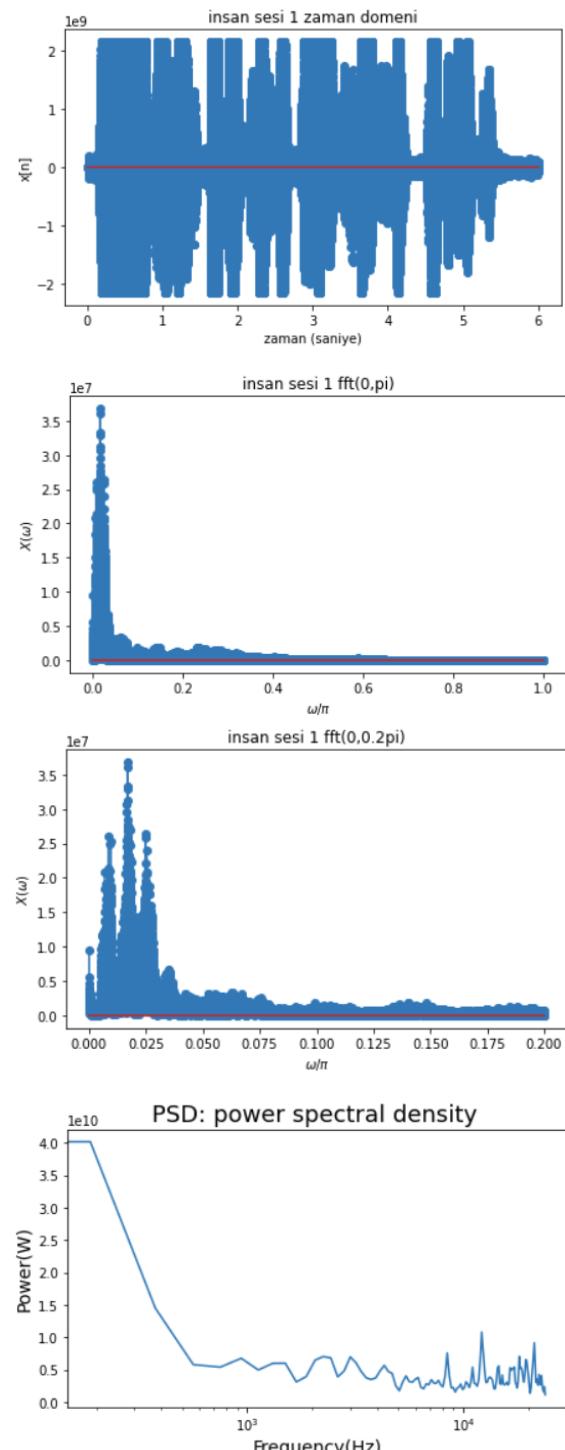
Şekil-23’te Ses işaretine ait genlik spektrumu ve Güç Yoğunluk Fonksiyonu görülmektedir. Genlik Spektrumunda görüleceği üzere drone ses işaretinin büyük çoğunluğu $0-0.02\pi$ arasında yoğundur analiz aralığını test etmek için 0.02π frekansından sonrası geçirecek bir yüksek geçiriren filtre kullanmak gereklidir. Filtreden sonra bu aralıktan sonrasıının önemli olmadığını karar verilmiştir.



Şekil 24 Ses-3’ün IIRhighpassfilterchebshev202 filtresinden geçirilmiş hali.

Ses-3 farklı filtreler ile denenmiştir ve en verimli çıktı Şekil-23’de $w=0-0.2\pi$ aralığına bakarak kesim frekansları $w=0.017-0.019\pi$ ve $w=0.033-0.037\pi$ bandsöndüren, $w=0.003$ yüksek geçiriren filtre ve son olarak $w=0.02\pi$ alçak geçirilen filtrden geçirilmesi planlanmıştır

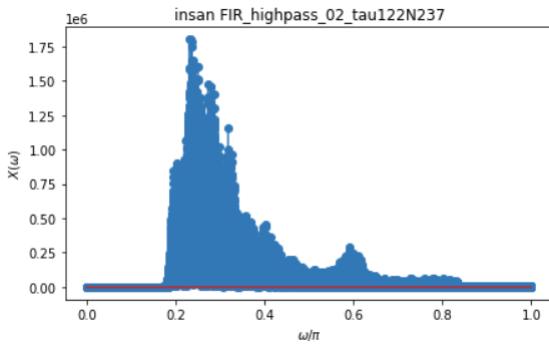
2.2.1.4 Ses-4 (people1.wav)



Şekil 25 Ses-4’ün Sirayla Genlik-Zaman, Genlik-Frekans($0,\pi$), Genlik-Frekans($0,2\pi$), Güç Yoğunluk Fonksiyonu Grafiği Verilmiştir

Şekil-25’te Ses işaretine ait genlik spektrumu ve Güç Yoğunluk Fonksiyonu görülmektedir. Genlik Spektrumunda görüleceği üzere drone ses işaretinin büyük çoğunluğu $0-0.2\pi$ arasında yoğundur analiz

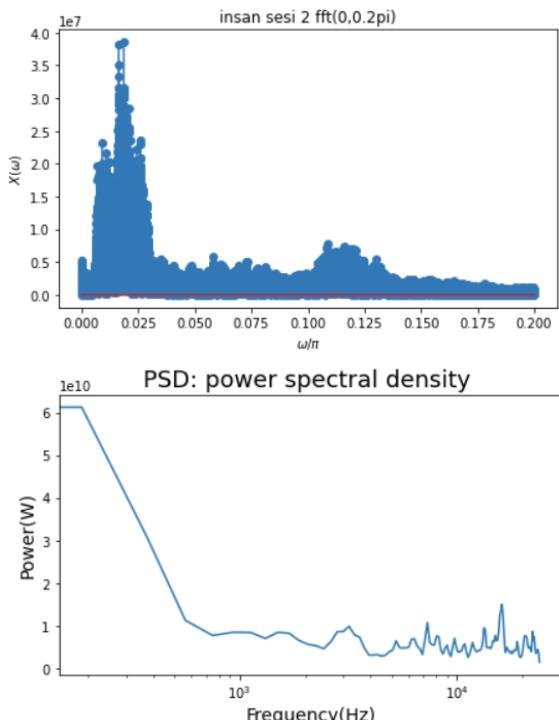
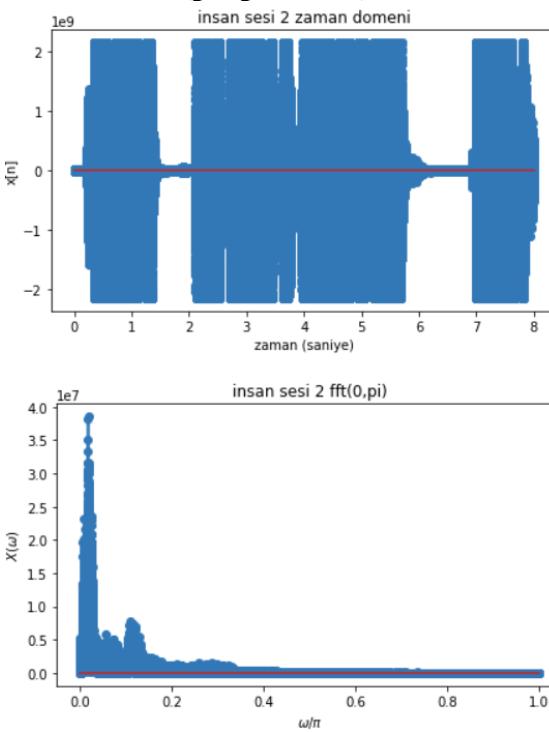
aralığını test etmek için 0.02π frekansından sonrası geçirecek bir yüksek geçen filtre kullanmak gerekir. Filtreden sonra bu aralıktan sonrasıının önemli olmadığını karar verilmiştir.



Şekil 26 Ses-4'ün FIR_highpass_02_tau122N237 filtresinden geçirilmiş hali.

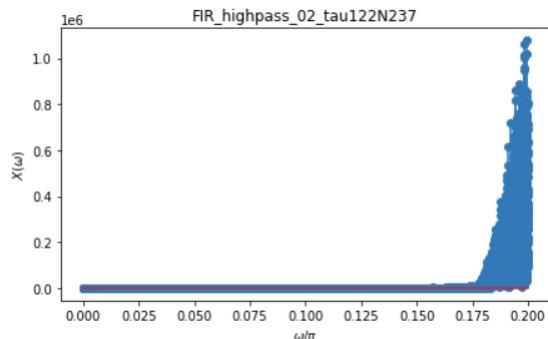
Ses-4 farklı filtreler ile denenmiştir ve en verimli çıktı Şekil-26'da gösterildiği üzere FIR_highpass_02_tau122N237 filtresinden sağlanmıştır.

2.2.2.5 Ses-5 (people2.wav)



Şekil 27 Ses-5'ün Sırayla Genlik-Zaman, Genlik-Frekans(0,pi), Genlik-Frekans(0,2*pi), Güç Yoğunluk Fonksiyonu Grafiği Verilmiştir

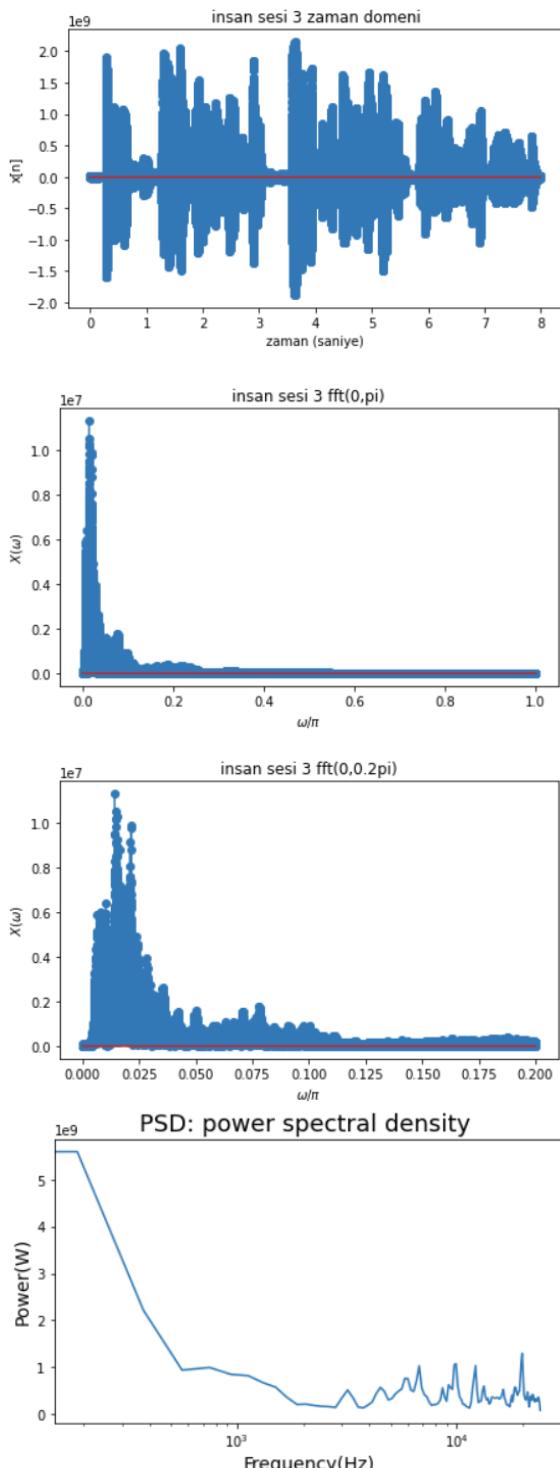
Şekil-27'de Ses işaretine ait genlik spektrumu ve Güç Yoğunluk Fonksiyonu görülmektedir. Genlik Spektrumunda görüleceği üzere drone ses işaretinin büyük çoğunluğu $0-0.2 \pi$ arasında yoğundur analiz aralığını test etmek için 0.02π frekansından sonrası geçirecek bir yüksek geçen filtre kullanmak gerekir. Filtreden sonra bu aralıktan sonrasıının重要意义 olduğunu karar verilmiştir.



Şekil 28 Ses-5'in FIR_highpass_02_tau122N237 filtresinden geçirilmiş hali.

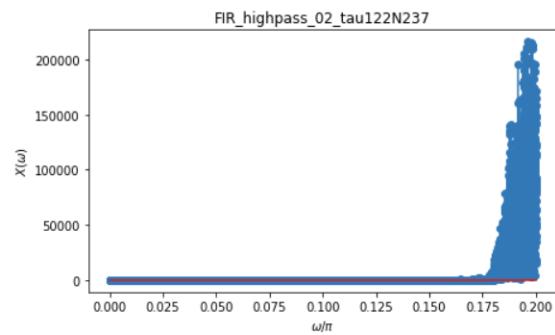
Ses-5 farklı filtreler ile denenmiştir ve en verimli çıktı Şekil-28'de gösterildiği üzere FIR_highpass_02_tau122N237 filtresinden sağlanmıştır.

2.2.2.6 Ses-6 (people3.wav)



Şekil 29 Ses-6'ün Sırayla Genlik-Zaman, Genlik-Frekans($0,\pi$), Genlik-Frekans($0,2\pi$), Güç Yoğunluk Fonksiyonu Grafiği Verilmiştir

Şekil-29'da Ses işaretine ait genlik spektrumu ve Güç Yoğunluk Fonksiyonu görülmektedir. Genlik Spektrumunda görüleceği üzere drone ses işaretinin büyük çoğunluğu $0-0.2\pi$ arasında yoğundur analiz aralığını test etmek için 0.02π frekansından sonrası geçirecek bir yüksek geçirgen filtre kullanmak gereklidir. Filtreden sonra bu aralıktan sonrasına önemli olmadığına karar verilmiştir.



Şekil 30 Ses-6'nın FIR_highpass_02_tau122N237 filtrinden geçirilmiş hali.

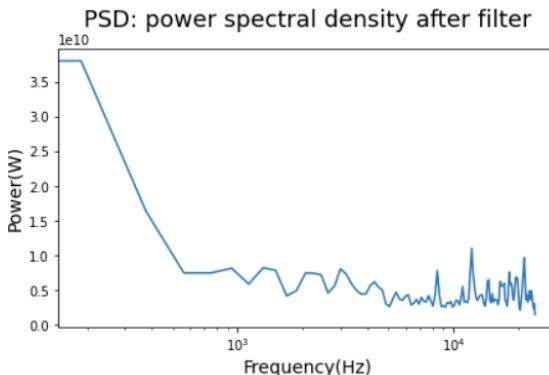
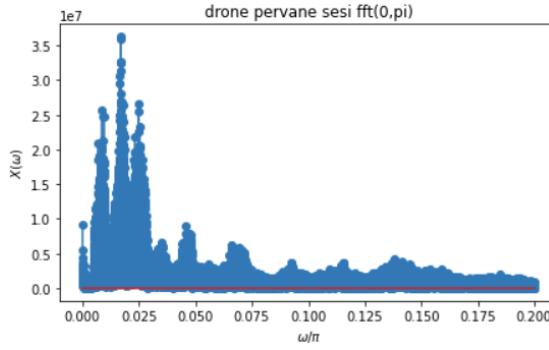
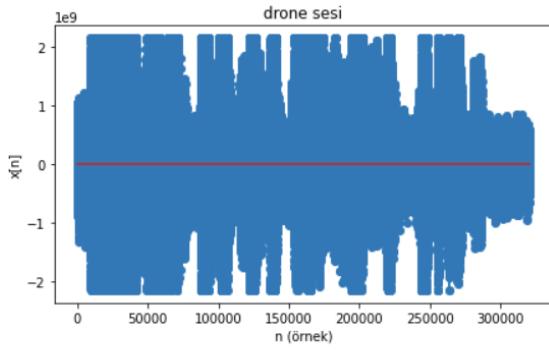
Ses-6 farklı filtreler ile denenmiştir ve en verimli çıktı Şekil-30'da gösterildiği üzere FIR_highpass_02_tau122N237 filtresinden sağlanmıştır.

2.2.2 Test Sesi İşaretleri

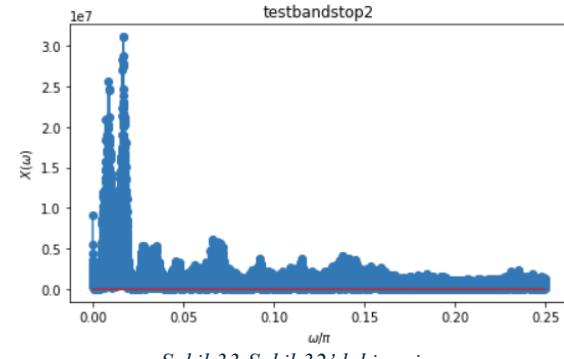
Bu kısımda test için oluşturulmuş ses işaretleri 2.1 Filtre Tasarımları ve Filtrelenmiş İşaretler başlığı altında tasarlanmış olan ve 2.2.1 Eğitim Sesi İşaretleri başlığı altında test edilmiş olan filtrelerin üst üste kullanılması ile gürültüden mümkün olduğunda arındırılmaya çalışılmıştır.

Üç farklı tipte drone sesi incelenmiştir. Bu droneların sesinin fft grafiklerinden genel bir filtre deseni elde edilememiştir. Her bir drone tipi için amaca uygun birçok filtre tasarlanmıştır. Tasarlanan filtreler içinden en uygun olanlar seçilerek en iyi ihtimaller değerlendirilmiştir. İşaretlerin fft grafikleri incelendiğinde analiz edilmesi gereken aralık $w=(0,0.2\pi)$ olduğu belirlenmiştir. $w=(0,0.2\pi)$ aralığı yüksek geçirgen filtre ile test edilerek filtre sonucu analiz aralığının doğru seçildiğinden emin olunmuştur. Tasarlanan filtrelerin FIR tipte çoğu dar bantlı bandsöndüren filtrelerdir. Ayrıca FIR tipte alçak geçirgen ve yüksek geçirgen filtrelerde bulunmaktadır. FIR filtreler sabit grup geçikmesine sahip olması filtre sonucunda işaret bozulmamasına yarar sağlamaktadır. Bozulmayan işarete yeni filtreler uygulanabilmesine de olanak sağlar.

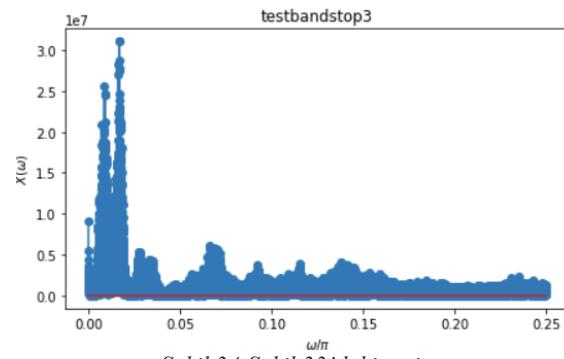
2.2.2.1 Test Sesi-1 (djiavata_people.wav)



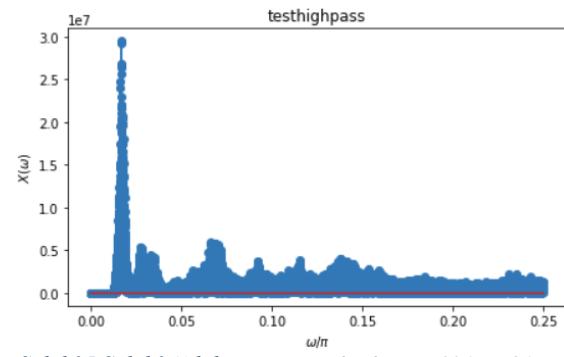
Şekil 31 Test Sesi-I 'in Strayla Genlik-Zaman, Genlik-Frekans, Güç Yoğunluk Fonksiyonu Grafiği Verilmiştir



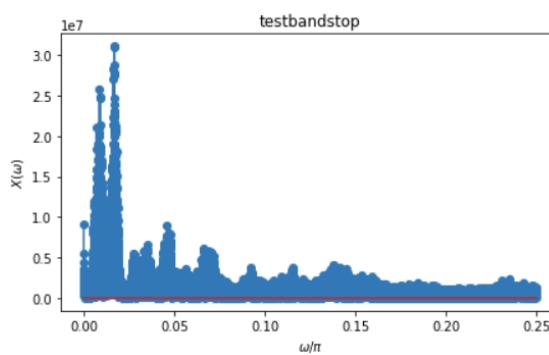
Şekil 33 Şekil 32'deki sesin
FIR_bandstop_004_005_129.matfiltresinden geçirilmiş
hali.



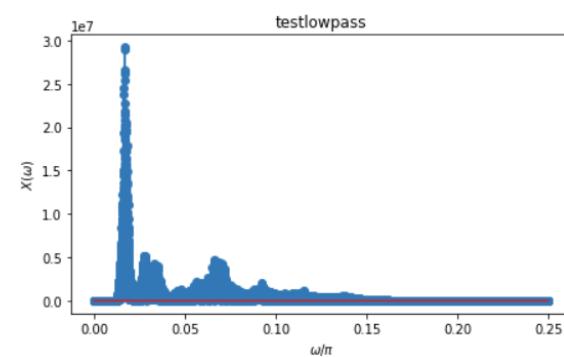
Şekil 34 Şekil 33'deki sesin
FIR_bandstop_004_005_129.matfiltresinden geçirilmiş
hali.



Şekil 35 Şekil 34'deki sesin FIR_highpass_0015_521.mat
filtresinden geçirilmiş hali.

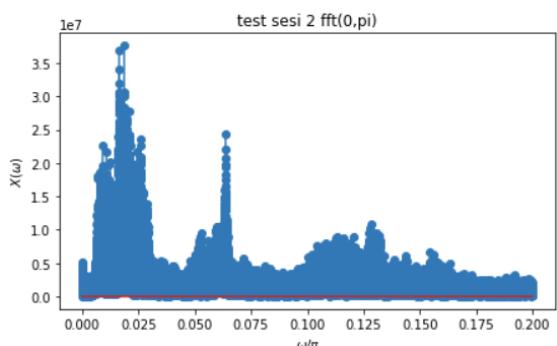
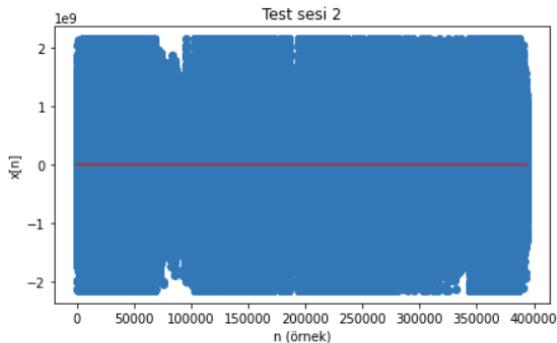


Şekil 32 Test Ses-I 'in
FIR_bandstop_0019_0028_400.matfiltresinden
geçirilmiş hali.

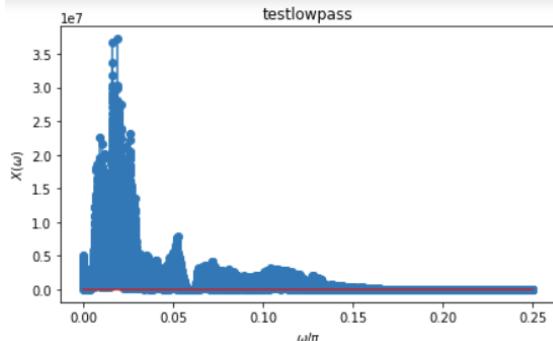
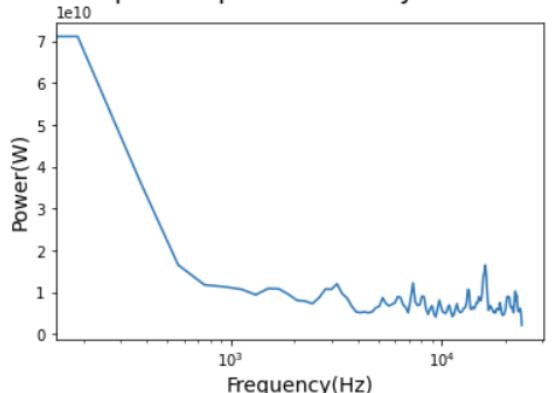


Şekil 36 Şekil 35'deki sesin FIR_lowpass_0075.mat
filtresinden geçirilmiş hali.

2.2.1.2 Test Sesi-2 (djifpv_people.wav)

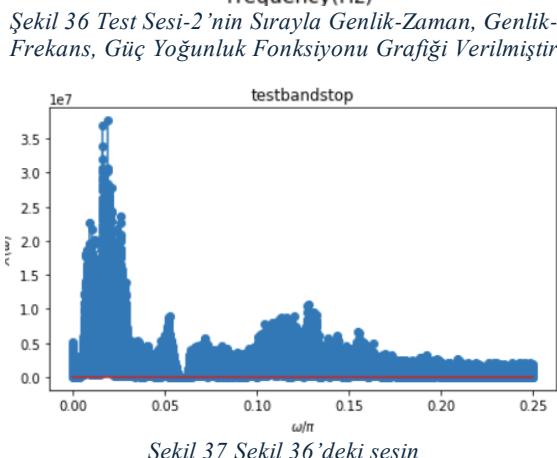


Şekil 36 Test Sesi-2'nin Sirayla Genlik-Zaman, Genlik-Frekans, Güç Yoğunluk Fonksiyonu Grafiği Verilmiştir

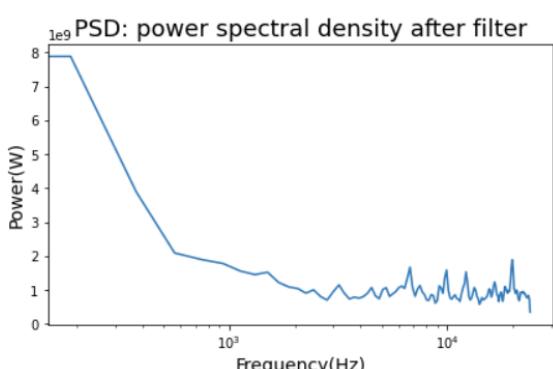


Şekil 38 Şekil 37'deki sesin FIR_lowpass_0075.mat filtresinden geçirilmiş hali.

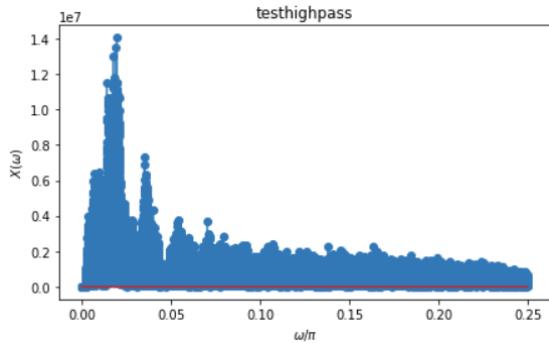
2.2.1.3 Test Ses-3 (fpvcinema_people.wav)



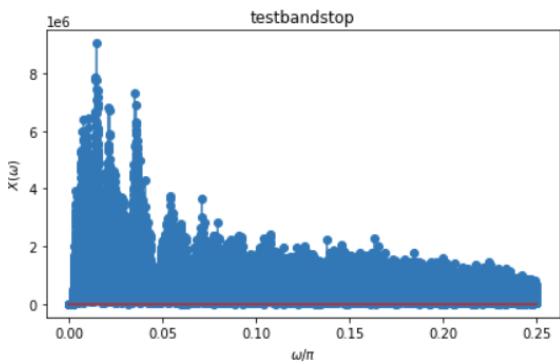
Şekil 37 Şekil 36'deki sesin _bandstop_0055_0065_505.mat filtresinden geçirilmiş hali.



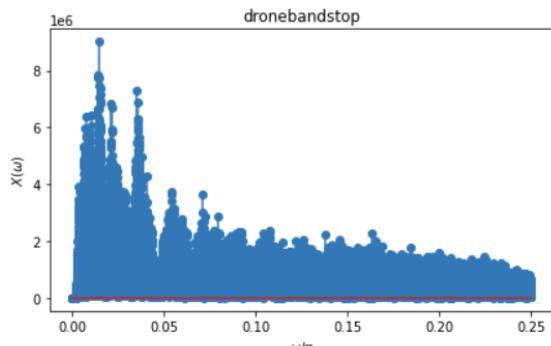
Şekil 39 Test Ses-3'in Sirayla Genlik-Zaman, Genlik-Frekans, Güç Yoğunluk Fonksiyonu Grafiği Verilmiştir



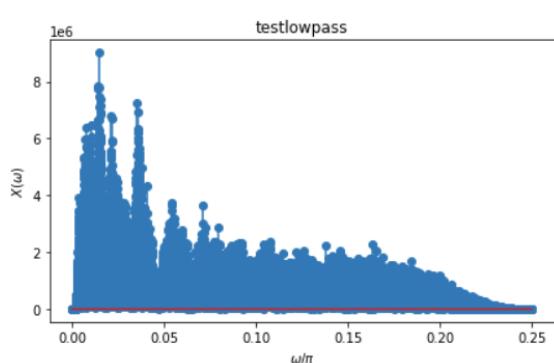
*Şekil 40 Şekil 39'daki sesin
FIRhighpassfiltertau1413band0003.mat filtresinden
geçirilmiş hali.*



*Şekil 41 Şekil 40'daki sesin
FIRbandstopfiltertau397band00085-00095.mat
filtresinden geçirilmiş hali.*



*Şekil 42 Şekil 41'deki sesin
FIRbandstopfiltertau127band033-037.mat filtresinden
geçirilmiş hali.*



*Şekil 43 Şekil 42'deki sesin
FIRlowpassfiltertau46band02.mat filtresinden geçirilmiş
hali.*

3. Sonuç ve Yorum

Elde edilen deneyimler ve önerilen çözümün avantajları/kısıtları:

Etkili Gürültü Bastırma: Önerilen çözüm gürültünün yoğun olduğu kısımları çeşitleri filtreler bastırma amacıyla ulaşmıştır

Uyum Sağlama Yeteneği: Çeşitli gürültü tiplerine ve seviyelerine uyum sağlama yeteneği sunar.

Kısıtlamalar: Farklı drone tipleri gürülü desenini değiştirdiği için her drone tipi için farklı filtreler tasarılanması gerekmektedir. Bu durumda, yöntemlerin gerçek zamanlı uygulamalar için optimize edilmesi gerekebilir.

Çözümü geliştirmek için öneriler:

Yöntem ve Algoritma Karşılaştırması: Farklı filtreleme yöntemlerini ve algoritmalarını deneySEL olarak karşılaştırmak ve performanslarını değerlendirmek.

Veri Çeşitlendirmesi: Daha fazla veri toplayarak ve modele eğitim vererek gürültü türlerine ve seviyelerine daha iyi uyum sağlamak.

Gerçek Zamanlı İşleme: Gerçek zamanlı işlemeyi geliştirmek için optimize edilmiş algoritmalar ve düşük gecikme gerektiren yaklaşımlar kullanmak.

Bu sonuçlar ve öneriler, proje süresince elde edilen bulgulara dayanmaktadır ve gelecekteki çalışmalar için yol gösterici olacaktır. Bu proje ile drone videolarındaki gürültü sorununu azaltarak, daha profesyonel ve yüksek kaliteli ses kayıtları elde edilebilecektir. Özellikle film yapımı, gazetecilik, doğa belgeselleri ve spor etkinlikleri gibi alanlarda bu teknolojinin önemli bir katkı sağlayacağı öngörlülmektedir.

GEBZE TECHNICAL UNIVERSITY

ELECTRONIC ENGINEERING DEPARTMENT

**ELEC 365
Fundamentals of Digital Communication**

MATLAB PROJECT REPORT

MUHAMMET KOÇ

1901022017

1. Table of contents

Contents

| | |
|---|----|
| 1. Table of contents | 1 |
| 2. Abstract..... | 2 |
| 3. Matlab Solution | 4 |
| 3.a. Simulation outputs for the theoretical approach..... | 7 |
| 3.b. Simulation outputs for equal probability | 8 |
| 3.c. Simulation with probability P1=1/4 P0=3/4 | 9 |
| 4. Analytic Solution | 9 |
| 5. Conclusion..... | 10 |
| 6. References | 10 |

2. Abstract

In a baseband communication system, $s_1(t) = \begin{cases} A \cdot \sin\left(\frac{2\pi t}{T}\right), & 0 < t < \frac{T}{2} \\ 0, & \text{else} \end{cases}$ and $s_2(t) = -s_1(t - \frac{T}{2})$ are transmitted for the bits "1" and "0", respectively. The bit error rate expression of this system is obtained and plotted on an additive white Gaussian channel (AWGN) for a) $P(1)=P(0)=1/2$, b) $P(1)=1/4$, $P(0)=3/4$. The system was simulated to obtain the BER curve versus SNR. The theoretical and simulated BER curves are analyzed.

Baseband Communication;

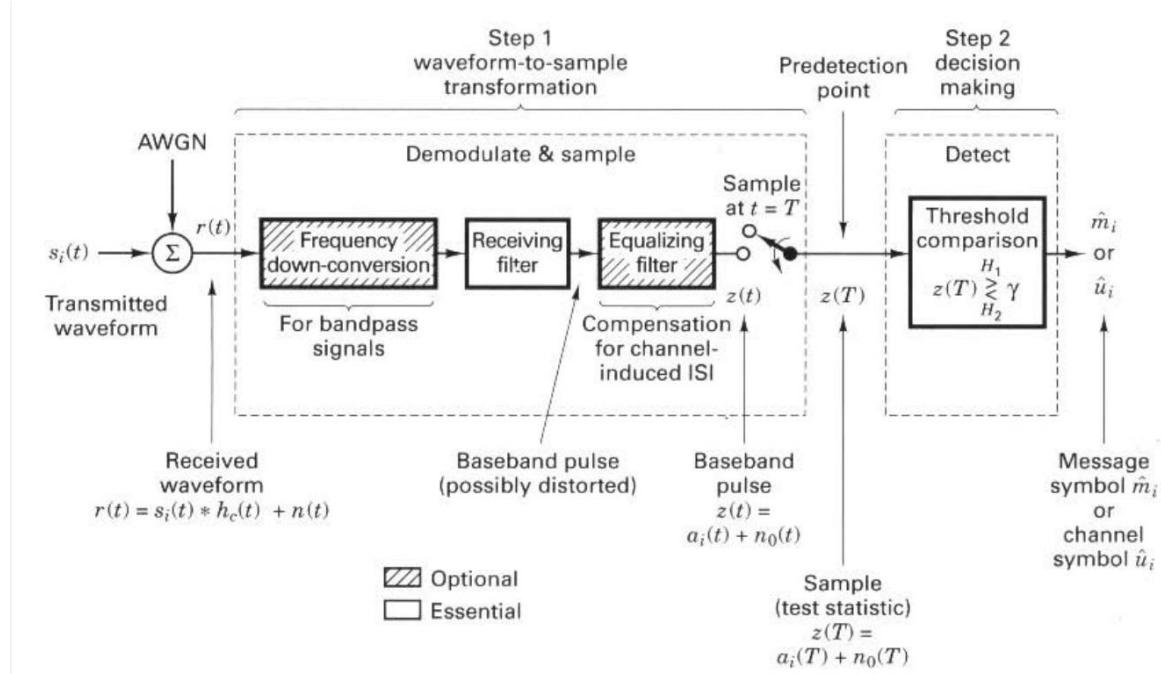


Figure 3.1 Two basic steps in the demodulation/detection of digital signals.

Figure 1

Figure 1 shows how to demodulate the noisy signal and make a decision. For the simulation, a vector simulation is used instead of a waveform. The vector simulation is obtained by calculating the error rate with 10 million bits after creating the required vectors in Matlab. The signal with white Gaussian noise added goes through the decision phase according to the parameters in the analytical solution and graphs are created accordingly.

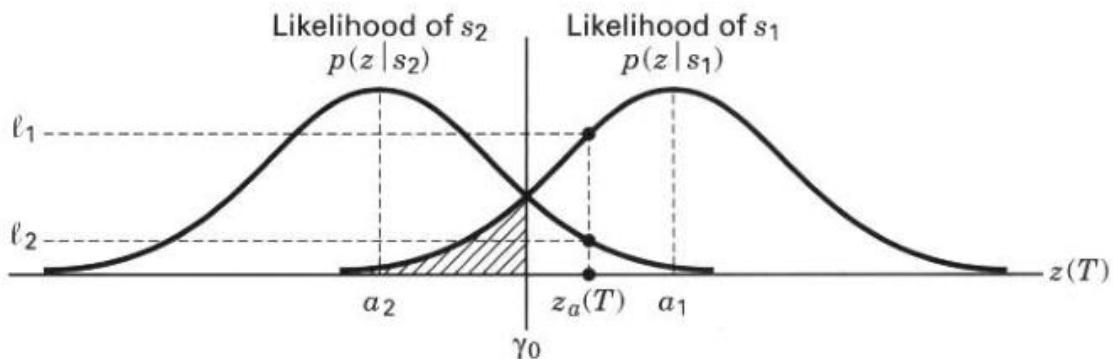


Figure 2:likelihood plots of sampled signals

In vector simulation, the gamma value determines the decision threshold. In binary baseband communication, if the simulation result is greater than the gamma value, it is treated as '1' bit, if it is less than the gamma value, it replaces the '0' bit.

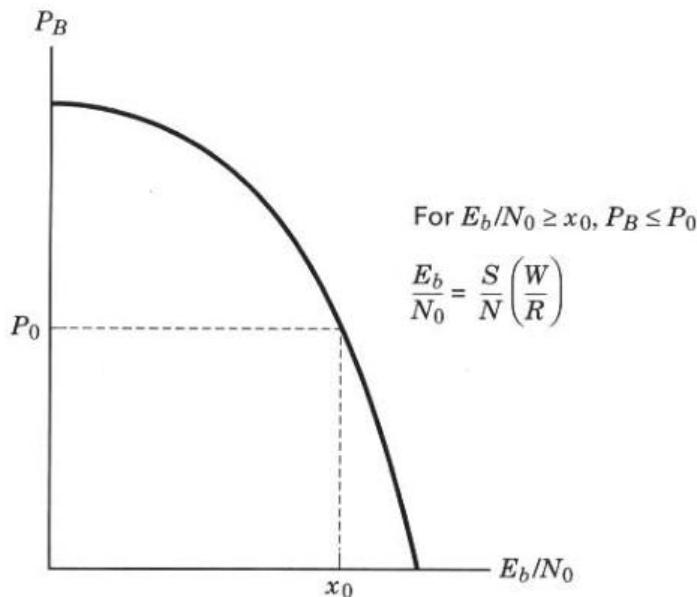


Figure 3.6 General shape of the P_B versus E_b/N_0 curve.

Figure 3: BER CURVE

In the simulation, all operations were performed assuming Eb value of 1. Figure 3 shows the bit error rate graph.

3. Matlab Solution

The code used for all stages of the simulation is given below. simulation outputs are placed after the code.

```
clc
close all;
snr = 0:1:15;
Eb=1;
%10log(Eb/N0)=snr

%teorik yaklaşım
N0 = 10.^(-snr/10); % Element-wise operation with .^
x = sqrt(2*Eb./N0); % Element-wise operation with ./ to ensure division is
element-wise
Pb = qfunc(x);
%teorik grafik
figure(1)
semilogy(snr, Pb);
ylabel("Pb")
xlabel("SNR")
title("teorik grafik")

%Dizi uzunluğu
n = 10^7;

% 0 ve 1'lerden oluşan dizi oluşturma
bit_array = randi([0, 1], 1, n);
%1lerden oluşan a vektörü
a=ones(1,n);
for i=1:n
    if bit_array(i)==0
        a(i)=-1; %0 değerli bitler için analitik olarak hesaplanan -1değeri
atanır
    end
end

% ortalaması 0 sigma=1 gauss gürültüsü
nx=randn(1,n);

% ortalaması 0 sigma^2=N0 gauss gürültüsü. varyans N0'a eşit hesaplandı
%n0 gürültü vektörü
n0=sqrt(N0).*nx;

%hata olasılığını hesaplamak için oluşturulan olasılık vektörü
error_number_a=zeros(1,16);

%a şıkçı eşit olasılık için analitik çözüm
Pb_a=qfunc(1./sqrt(N0));
%a şıkçı eşit olasılık için simülasyon
gamma_a=0;
```

```
%a şıkçı eşit olasılık için analitik çözüm
Pb_a=qfunc(1./sqrt(N0));
%a şıkçı eşit olasılık için simülasyon
gamma_a=0;

first20=sprime(1:20);
for i=1:1:16
error=0;
z=a+n0(i,:); %z vektörü
sprime=zeros(1,n);
for k=1:n

    if z(k)>gamma_a %z vektörü gamadan büyükse
        sprime(k)=1; % s' vektörü 1 değerini alır
        %gamadan küçükse sıfır değerini alır
        %s' vektörü sıfırlardan oluştuğu için bu yeterlidir
    end
    if sprime(k) ~= bit_array(k) % s' vs s vektörü kontrol edilir
        error =error+ 1; % fark varsa hata sayısına bir eklenir
    end

end
disp(error)

error_number_a(i)=error/n; % ilgili No değeri için hata olasılığı
end

figure(2)
semilogy(snr, Pb_a,"b--");
ylabel("Pb(Bit error rate)")
xlabel("SNR")
hold on;
semilogy(snr, error_number,"r");
ylabel("Pb(Bit error rate)")
xlabel("SNR(dB)")
title("eşit olasılıklı simülasyon","1901022017 Muhammet KOÇ")
legend ('Q fonk. ile analitik çözüm','simülasyon sonucu');
```

```
%b şıkları P1=1/4 P0=3/4 için analitik çözüm
gamma_b=0.55*N0; % analitik olarak hesaplanan gama değeri
P1=1/4; P0=3/4;
Pb_b=qfunc((1-gamma_b)./sqrt(N0))./4+qfunc((1+gamma_b)./sqrt(N0)).*3./4;
%b şıkları P1=1/4 P0=3/4 için simülasyon
%hata olasılığını hesaplamak için oluşturulan olasılık vektörü
error_number_b=zeros(1,16);

%asagidakı for döngüsü a kismı ile aynı işlemleri içerir.
%sadece gama değeri vektör olduğu için o kismın değişimiştir
%yukarıdaki kod açıklamaları aynen geçerlidir
for i=1:1:16
error=0;
z=a+n0(i,:);
sprime=zeros(1,n);
for k=1:n

    if z(k)>gamma_b(i) %z vektörü gamadan büyükse
        sprime(k)=1; % s' vektörü 1 değerini alır
    end
    if sprime(k) ~= bit_array(k) % s' vs s vektörü kontrol edilir
        error =error+ 1; % fark varsa hata sayısına bir eklenir
    end

end
disp(error)
error_number_b(i)=error/n; % ilgili N0 değeri için hata olasılığı
end

figure(3)
semilogy(snr, Pb_b,"b--");
ylabel("Pb(Bit error rate)")
xlabel("SNR")
hold on;
semilogy(snr, error_number_b,"r");
ylabel("Pb(Bit error rate)")
xlabel("SNR(dB)")
title("P1=1/4 P0=3/4 olasılıklı simülasyon","1901022017 Muhammet KOÇ")
legend ('Q fonk. ile analitik çözüm','simülasyon sonucu');
```

3.a. Simulation outputs for the theoretical approach

The theoretical graph up to 15 decibels where $E_b=1$ is shown in figure 4.

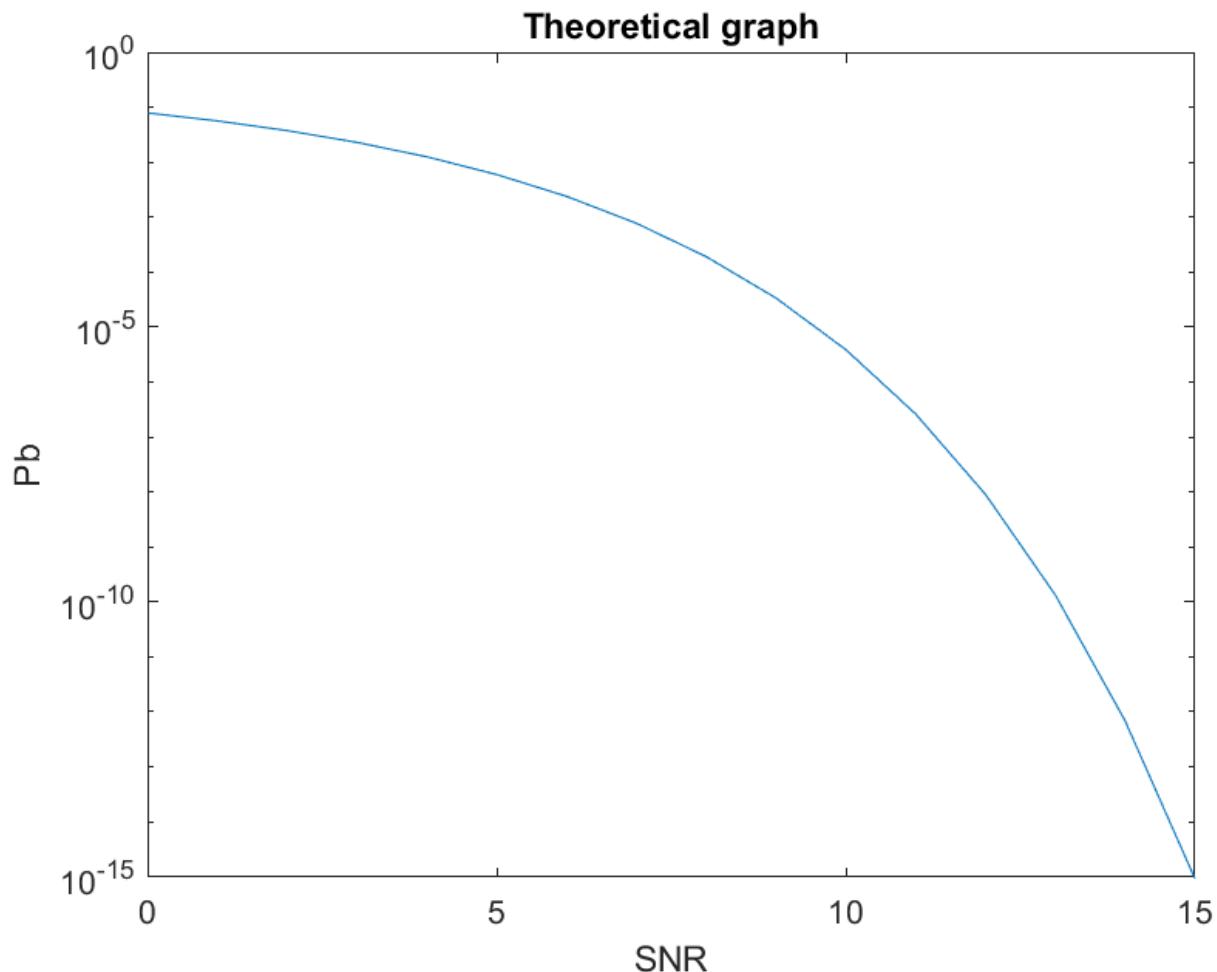


Figure 4

3.b. Simulation outputs for equal probability

The graph with equal probability of bits is shown in figure 5.

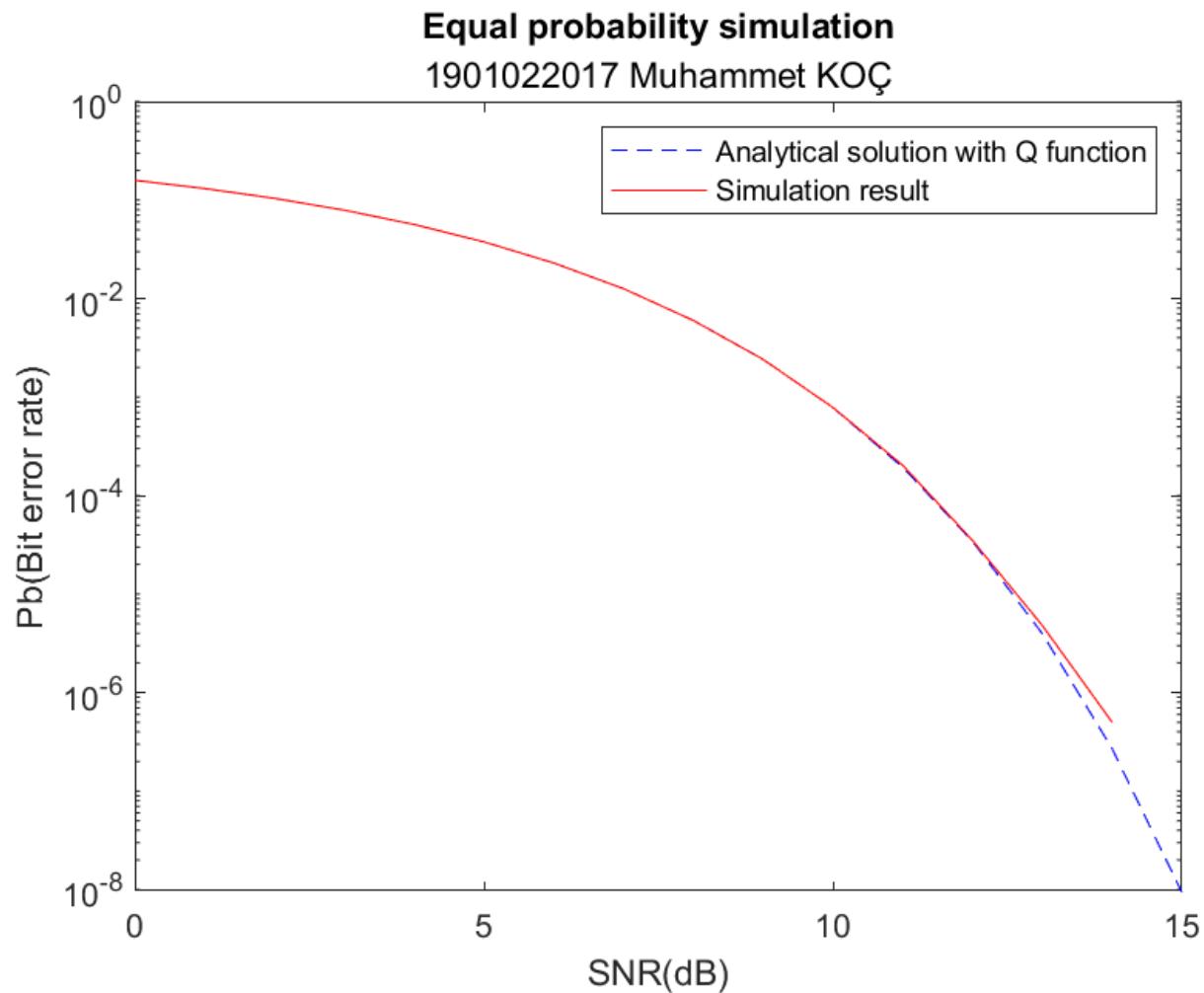


Figure 5: Equal probability

3.c. Simulation with probability $P_1=1/4$ $P_0=3/4$

The graph obtained for the simulation with probability $P_1=1/4$ $P_0=3/4$ is shown in figure 6

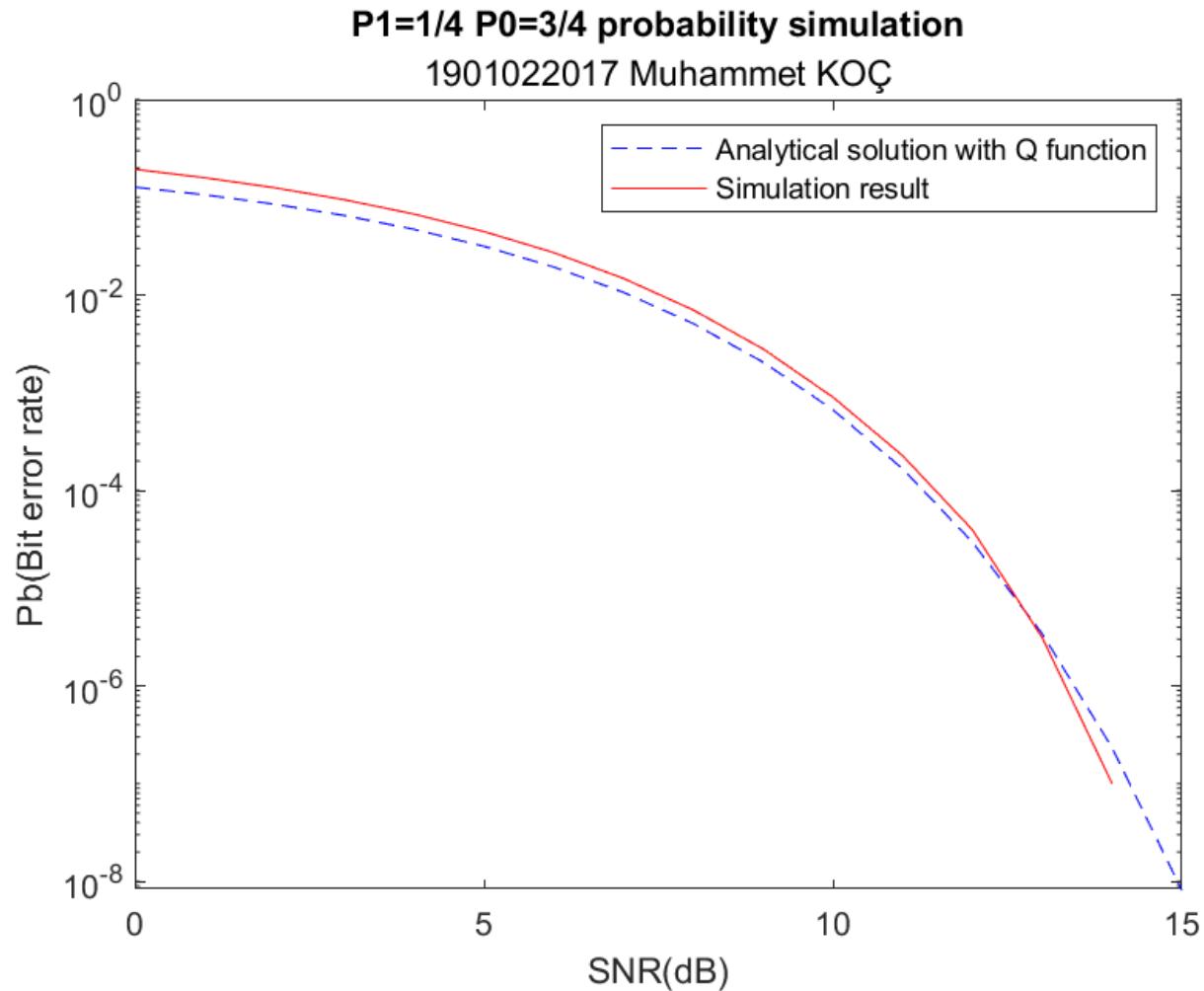


Figure 6: not equal probability

4. Analytic Solution

The analytical solution is available on the next pages.

5. Conclusion

As a result of this project, the vector simulation of the binary communication system of baseband communication was realized in all stages. First, the theoretical bit error rate graph for $E_b=1$ was obtained. The N_0 values obtained for the theoretical graph were used in other parts of the simulation. The first stage of the simulation is the equal probability case. When bits 1 and 0 are equally likely to be transmitted, it is easy to calculate analytically. The bit error probability function becomes quite simple. If we look at figure 5, the analytical solution and the simulation result are almost identical. In the second case, where there is no equal probability for bit transmission, the bit error probability function becomes quite complex. In the simulation, the gamma value used in the decision step becomes more deterministic with other parameters. Figure 6 shows that the analytical solution approximates the simulation result, but the agreement is not as good as in the first case. When the system becomes more complex, analytical solutions can be used to obtain approximate results, but simulations and experiments are more effective.

6. References

1. Bernard Sklar, Digital Communications Fundemantal and Applications, Second Edition, Prentice Hall

**GEBZE TECHNICAL UNIVERSITY
ELECTRONICS ENGINEERING DEPARTMENT**

ELM 366 COMMUNICATIONS LABORATORY

REPORT

**EXPERIMENT-III
Sampling and Reconstruction**

Prepared by: *Muhammet Koç*

Student Number: *1901022017*

Team and Group Number: Group-3 and Team-2

Team Mates: HALİL ÇİFTÇİ, İSMAİL AKKOYUN

Date of the experiment: *03.04.2023*

Contents

Table of Contents

| | |
|--|----|
| Contents..... | 1 |
| 1. GOAL..... | 2 |
| 2. BACKGROUND | 2 |
| 2.1. Sampling Method..... | 2 |
| 2.2. Reconstruction of a Sampled Signal..... | 3 |
| 3. PROCEDURE..... | 4 |
| 3.1. Sampling of a Band-limited Signal | 4 |
| 3.2. Reconstruction of the Sampled Signal | 7 |
| 4. CONCLUSION..... | 14 |

1. GOAL

This experiment will cover how to sample and reconstruct a band-limited signal

2. BACKGROUND

2.1. Sampling Method

In digital signal processing, one of the fundamental processes is sampling of a band-limited analog signal that is continuous in time domain. This process can be described by the following expression.

$$x_\delta(t) = \sum_{n=-\infty}^{\infty} x(nT_s)\delta(t - nT_s)$$

where, $x(t)$ is a band-limited continuous signal and T_s is the sampling period and $\delta(t)$ is an impulse function with a definition as given below. Block diagram of the sampling process is shown in Figure 2.1.1.

$$\delta(t) = \begin{cases} 1, & t = 0 \\ 0, & t \neq 0 \end{cases}$$

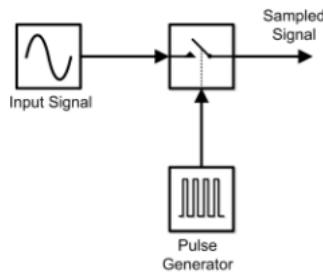


Figure 2.1.1. A scheme of the sampling process

Discrete Fourier Transform (DFT) of a sampled signal $x_\delta(t)$ is given as follows:

$$X_\delta(f) = T_s \sum_{n=-\infty}^{\infty} x(nT_s)e^{-j2\pi f n T_s} = T_s \sum_{m=-\infty}^{\infty} X(f - mf_s)$$

where, $f_s = \frac{1}{T_s}$ is the sampling frequency. To avoid aliasing, the sampling frequency f_s must be greater than $2 \times f_{max}$ where f_{max} is the maximum frequency component of $x(t)$ and $f_s = 2 \times f_{max}$ is also known as the Nyquist Frequency. Let us consider a continuous input signal as shown below. Sampled signal can be generated as the following figure. In the sampling process, frequency spectrums of the input signal and the sampled signal are shown as below.

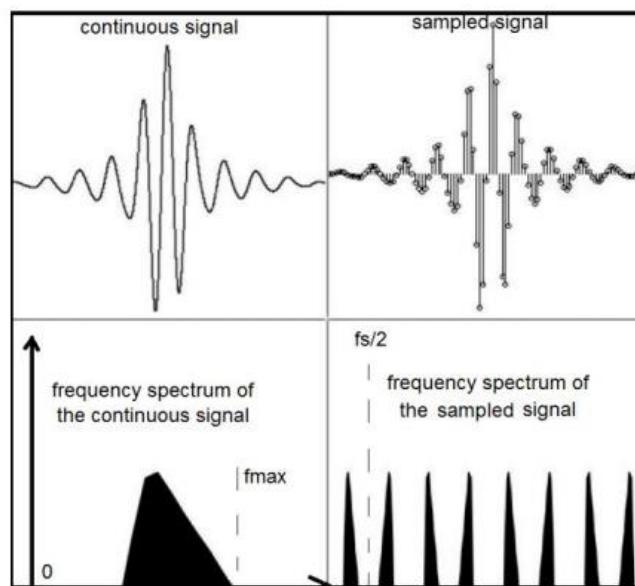


Figure 2.1.2. Frequency spectrums of input and sampled signals

2.2. Reconstruction of a Sampled Signal

A sampled signal with the sampling frequency greater than the Nyquist frequency can be used to reconstruct input signal by using a low pass filtering process. A scheme of the reconstruction process is illustrated in Figure 2.1.3.

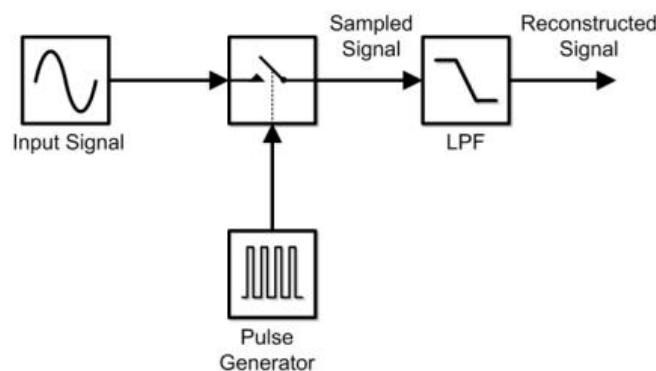


Figure 2.1.3. A scheme of the sampling process

3. PROCEDURE

3.1. Sampling of a Band-limited Signal

As can be seen in Figure 1, the following operations were performed respectively:

- In the Master Signals block, set the frequency of the sine tone to 2 kHz.
- In the Function Generator block, a square signal with a frequency of 10 kHz, an offset of 2.5 V and a peak-to-peak amplitude of 5 V was generated.
- From the Pulse Generator block, samples of the tone signal were generated using 10 kHz pulses with 40 μ s width and 40 μ s delay.
- Channel Voltage Range: 1V/div and Horizontal Timebase: 100 μ s/div.

After these operations, the program was run and the input and output graphs were observed from the oscilloscope section.

In the most general case (ideal) sampling is done with periodic impulse (dirac) functions. Here, however, sampling is done using periodic square pulse signals. This sampling is called natural sampling. In natural sampling, the square pulse signals take the shape of the sampled signal. This is clearly seen in the oscilloscope image.

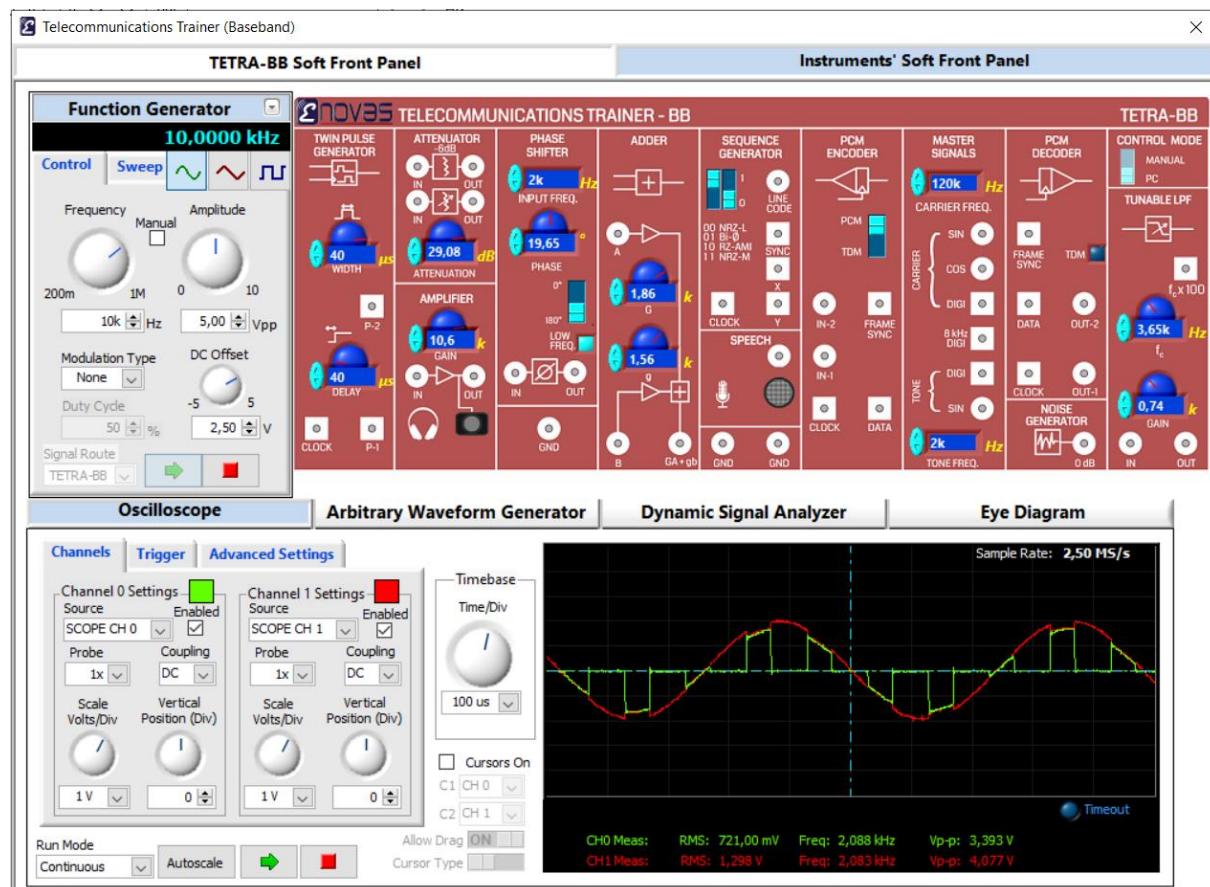


Figure 1 Natural Sampling

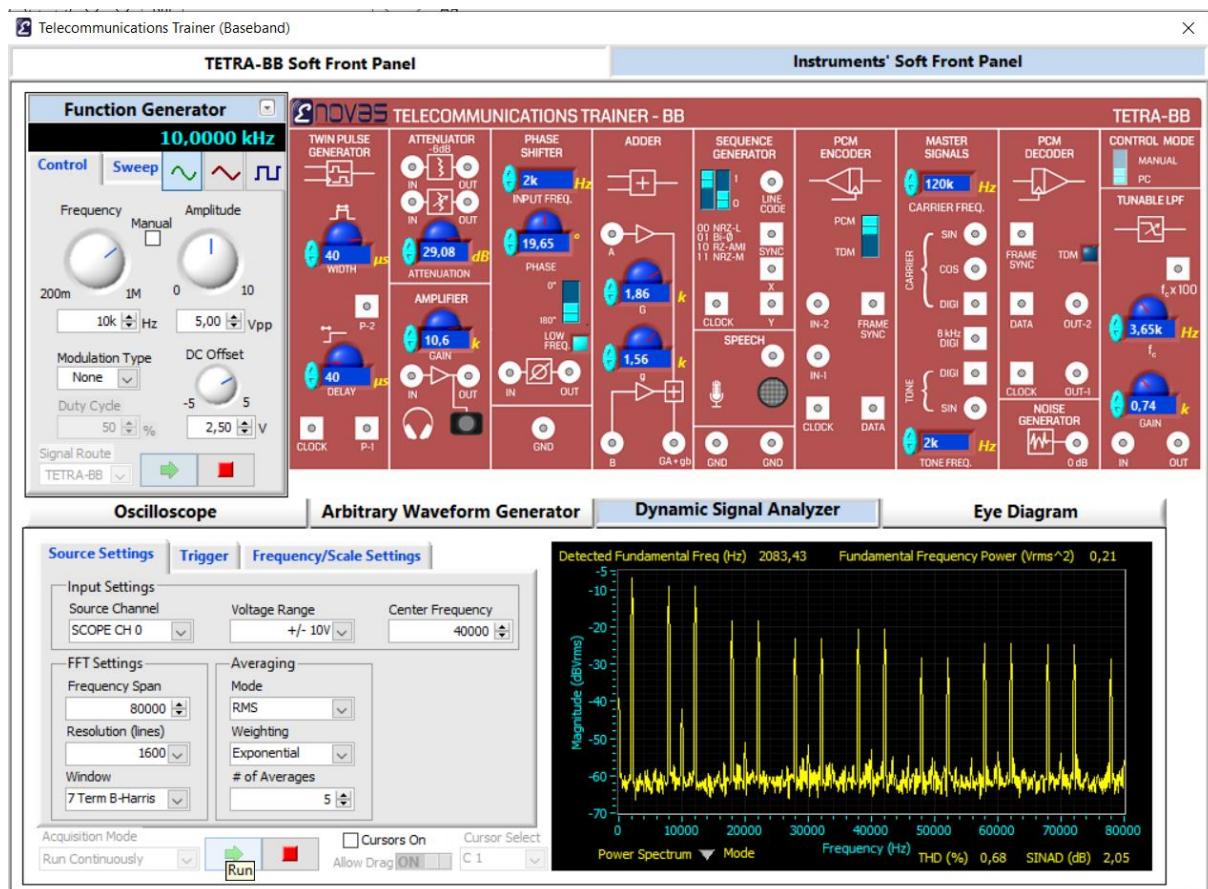


Figure 2 DSA Panel of Naturel Sampled Signal

In Figure 2, the same signal is processed on the DSA panel. The DSA panel shows the Fourier Transform of the signal sent to the program.

As can be seen in Figure 2, the following operations were carried out respectively:

- Center Frequency: 40000
- Frequency Span: 80000
- Mode: RMS
- Resolution (lines): 1600
- Weighting: Exponential
- Window: 7 Term-B Harris and Averages 5

If we look at Figure 2, it is seen that the frequency in the center is 40 kHz, which we set. At the +/-2kHz edges of 40 kHz, one dirac is observed. Since the tone frequency is 2 kHz, these dirac values are located 2 kHz to the right and left of the center frequency. Since the sampling frequency is 10kHz, the spectrum of the sampled signal repeats shifted at every 10kHz frequency.

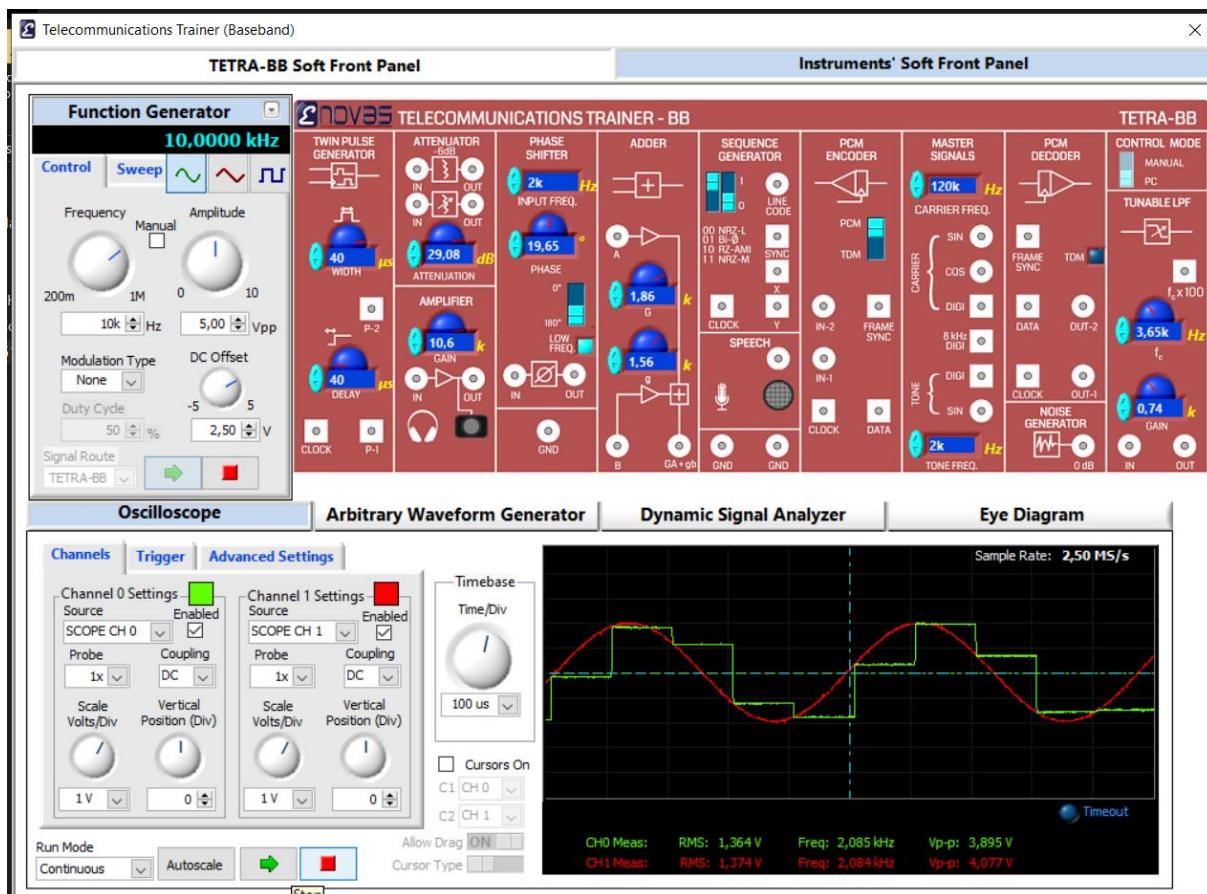


Figure 3 Sample and Hold

The same connection in the previous circuit is attached to the sample and hold circuit this time.

A **sample and hold circuit** is an analog device that takes the voltage of a continually changing analog signal and holds it at a consistent level for a set amount of time.

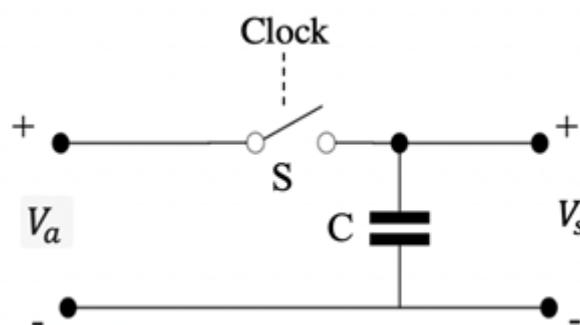


Figure 4 The simplest sample and hold circuit

As seen in Figure 3, it is the result of the sample and hold circuit. The tone signal is sampled and held for a certain period of time, then sampled again and the same process is repeated.

Natural sampling and sample and hold are two different sampling methods. Natural sampling is performed using a pulse train. In Sample and Hold circuit modeling, an impulse train is used

and then the sampling is completed by passing the impulse through a filter with impulse response $h(t)$.

3.2. Reconstruction of the Sampled Signal

As can be seen in Figure 5, the following operations were performed respectively:

- In the Master Signals block, set the frequency of the sine tone to 2 kHz.
- In the Function Generator block, a square signal with a frequency of 10 kHz, an offset of 2.5 V and a peak-to-peak amplitude of 5 V was generated.
- From the Pulse Generator block, samples of the tone signal were generated using 10 kHz pulses with 40 μ s width and 40 μ s delay.
- Channel Voltage Range: 1V/div and Horizontal Time Base: 100 μ s/div

To recover the sampled signal, it is necessary to pass it through a low-pass filter. The cut-off frequency of the low-pass filter must be higher than the frequency of the sampled signal or it cannot be recovered. Figure 5 shows that the signal cannot be recovered when it passes through an LPF with a cut-off frequency lower than 2kHz.

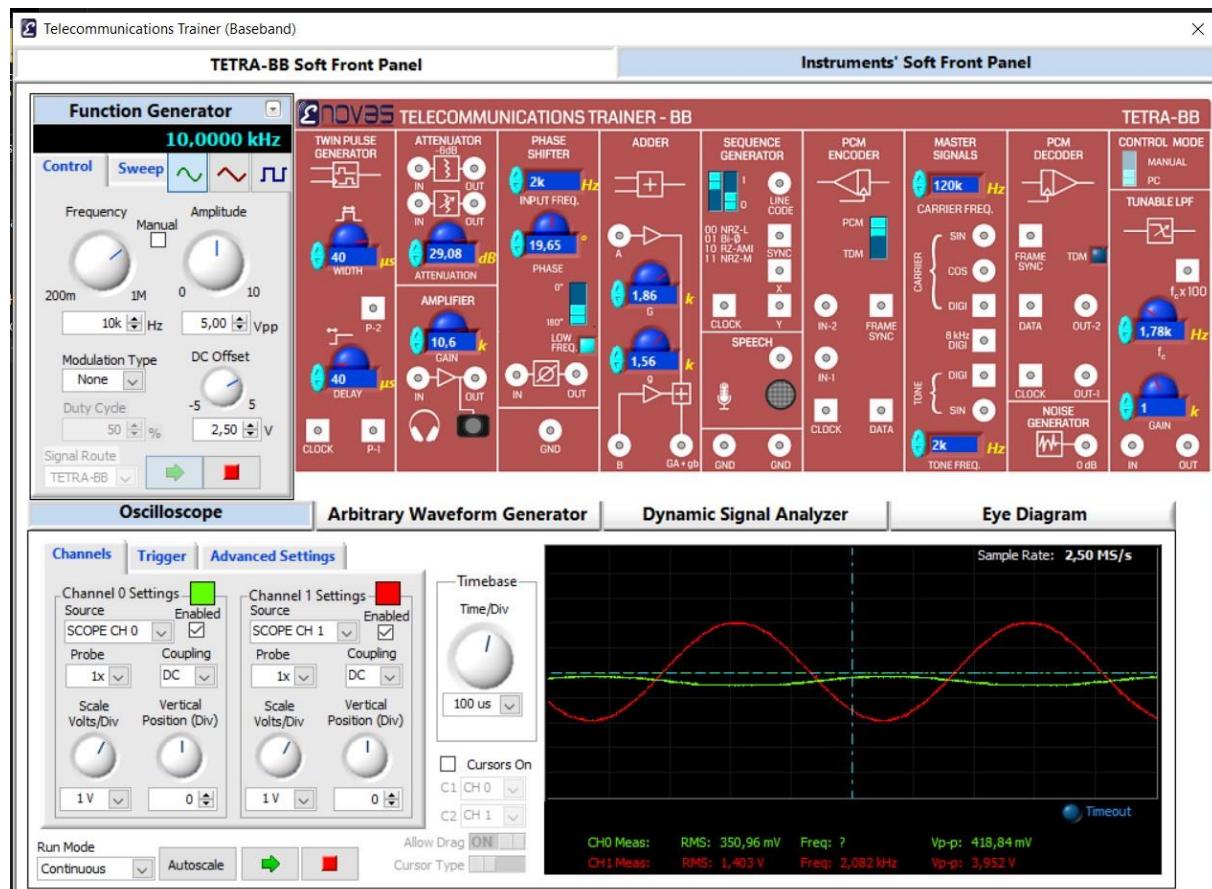


Figure 5: not reconstruct sampled signal

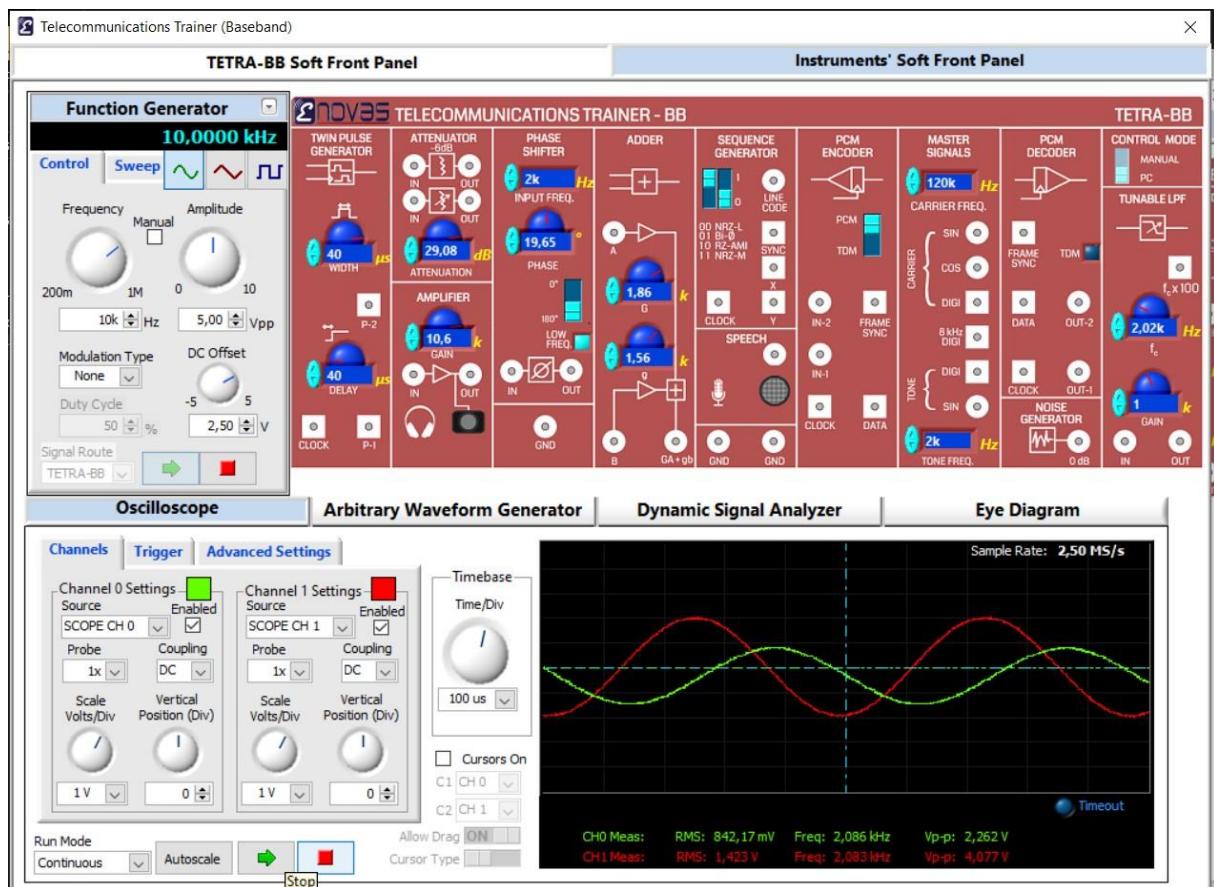


Figure 6 Reconstruction of the Sampled Signal

As shown in Figure 6, the sampled signal with pulse train was restored by passing it through the Lpf circuit with cutoff frequency 2.02kHz. There is a phase shift in the recovered signal.

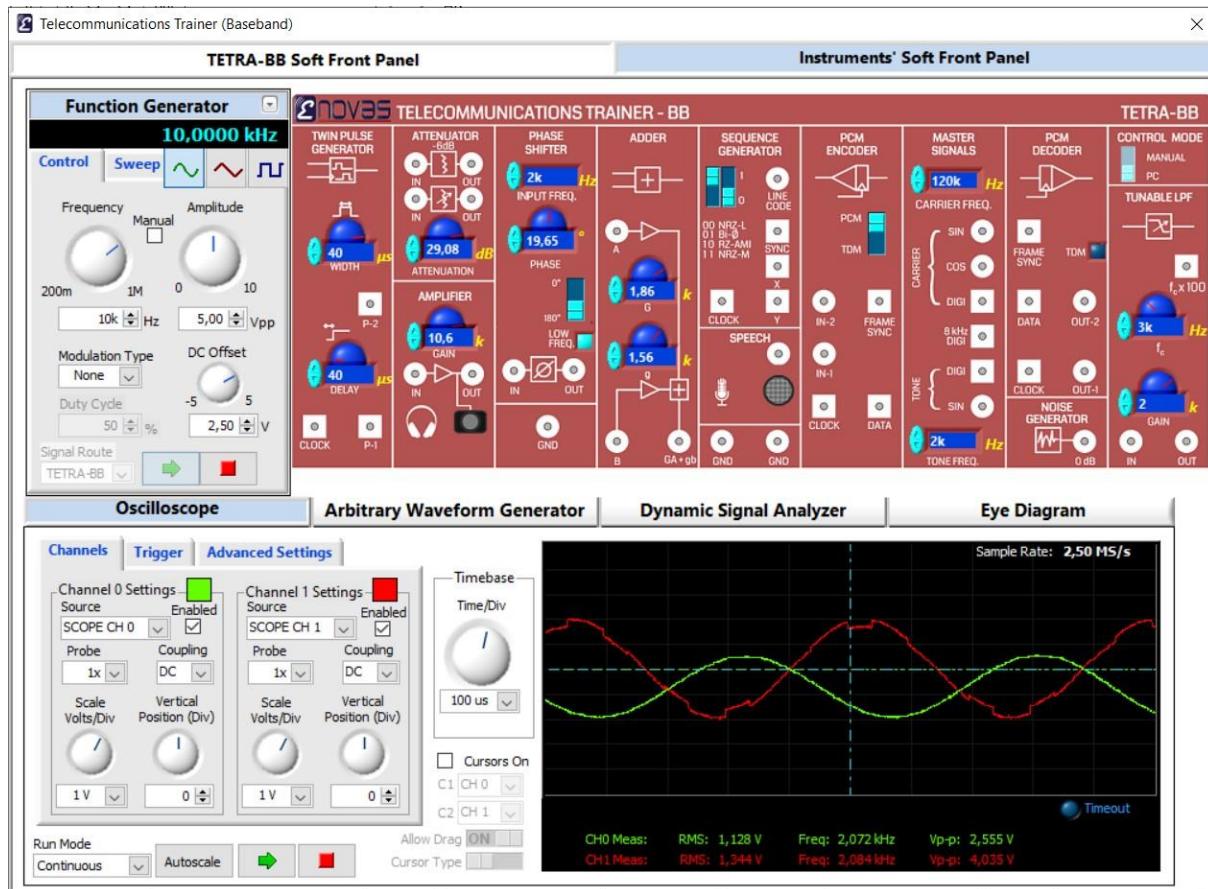


Figure 7 Reconstruction of Sample and Hold

As shown in Figure 7, the sampled signal with sample and hold was restored by passing it through the Lpf circuit with cutoff frequency 3kHz. There is a phase shift in the recovered signal.

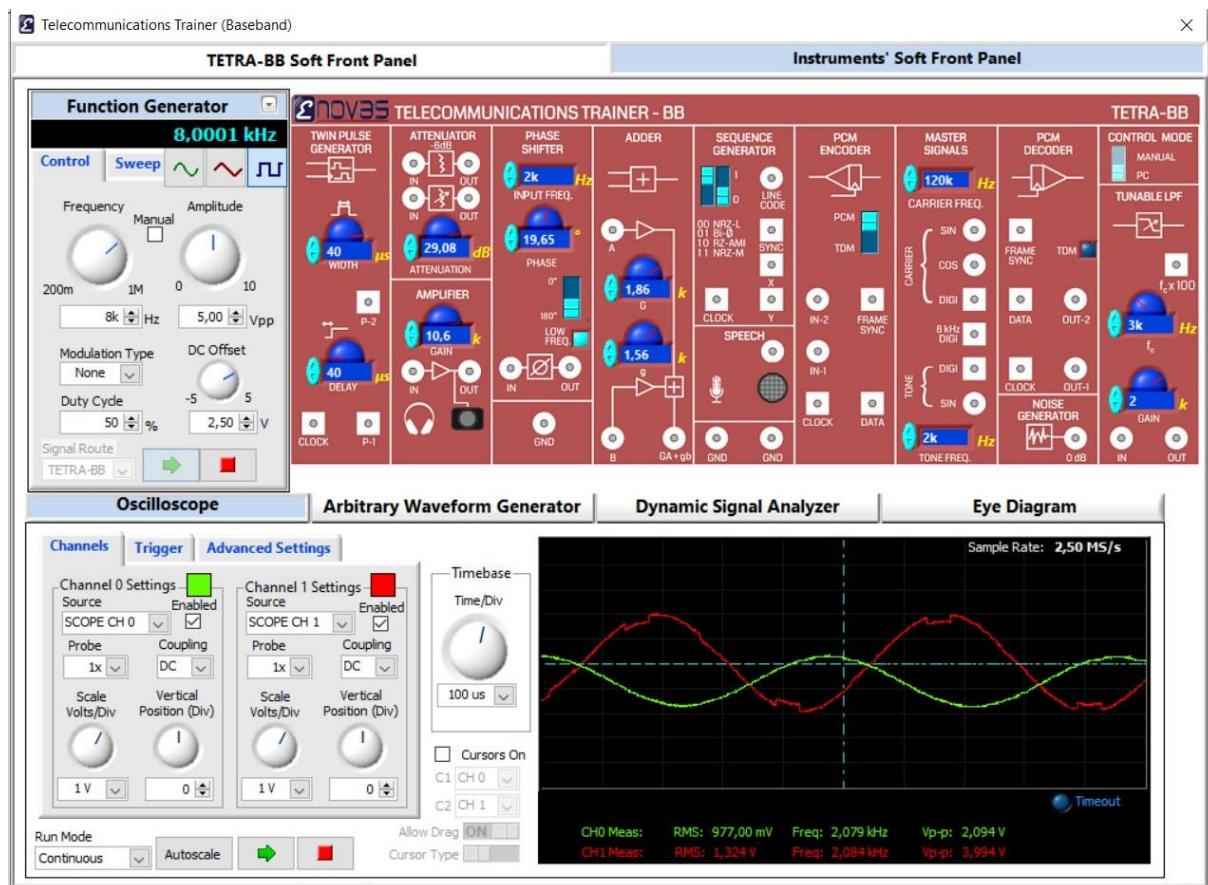


Figure 8 -8kHz- sampling

As can be seen in Figure 8, the following operations were carried out respectively:

- Generate a square signal with 8kHz frequency and 2.5V offset and peak to peak amplitude 5V in the Function Generator block.
- Reduce the frequency of the frequency generator's output by 1kHz

At this point, looking at the oscilloscope image in Figure 8, it is seen that the signal is recovered without distortion. The aim of this section is to observe the distortion by lowering the frequency of the function generator, in other words the sampling frequency.

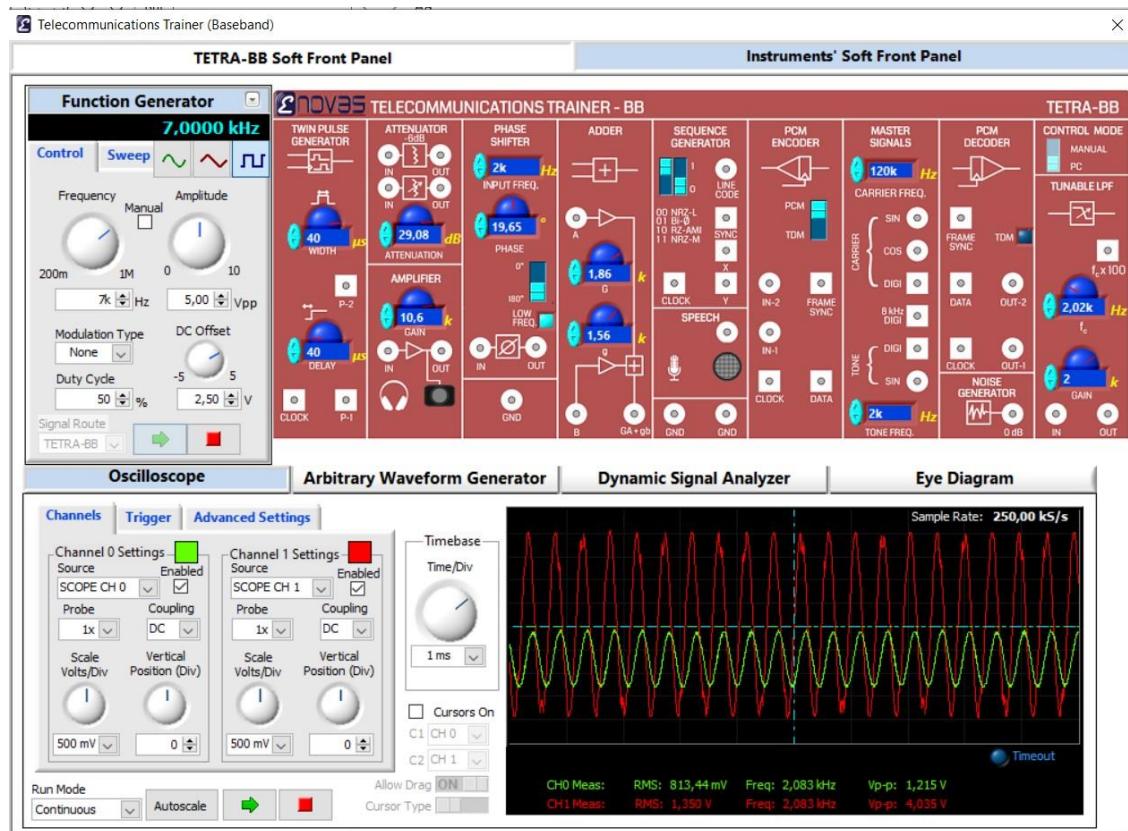


Figure 9 7kHz- sampling

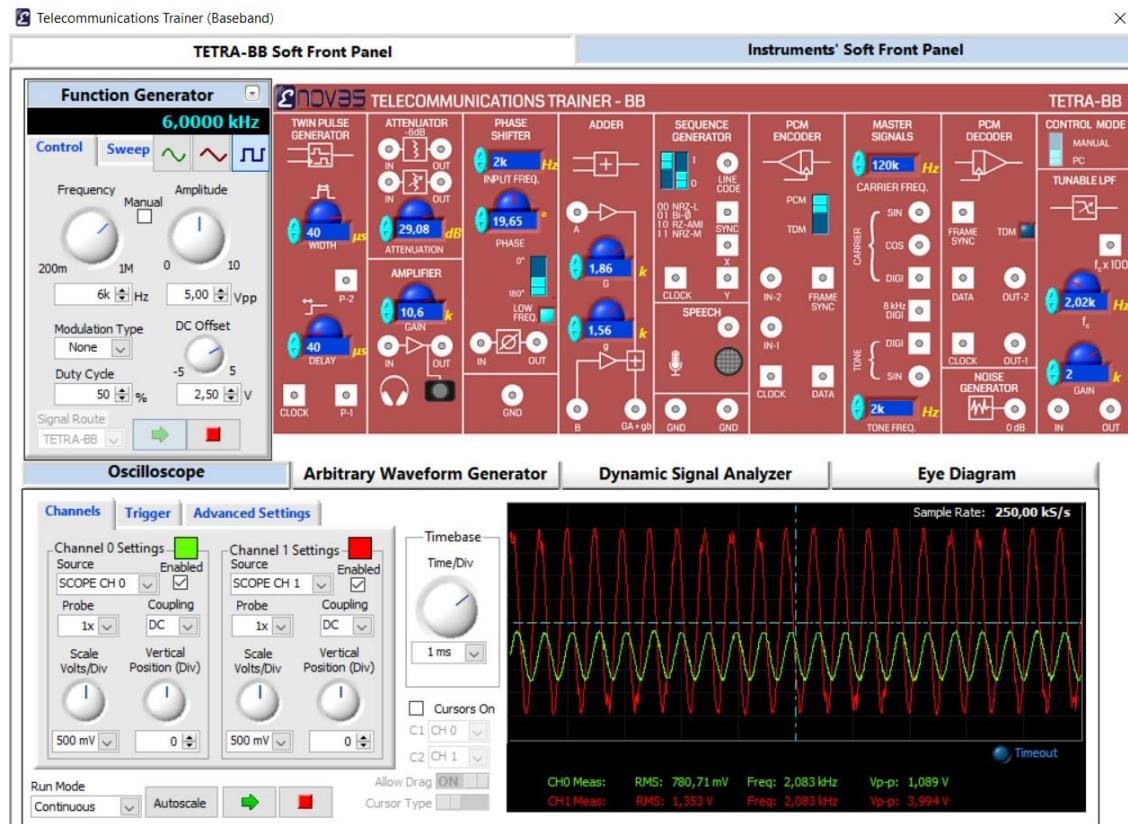


Figure 10 6kHz- sampling

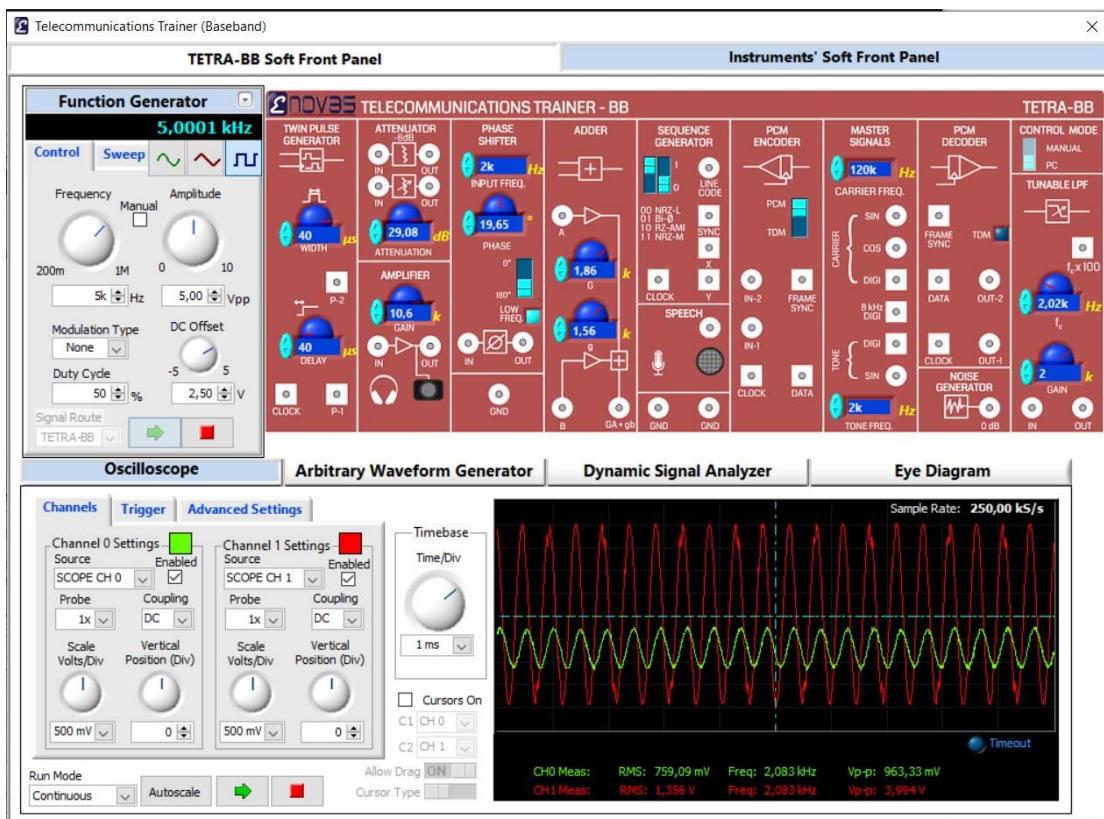


Figure 11 5kHz- sampling

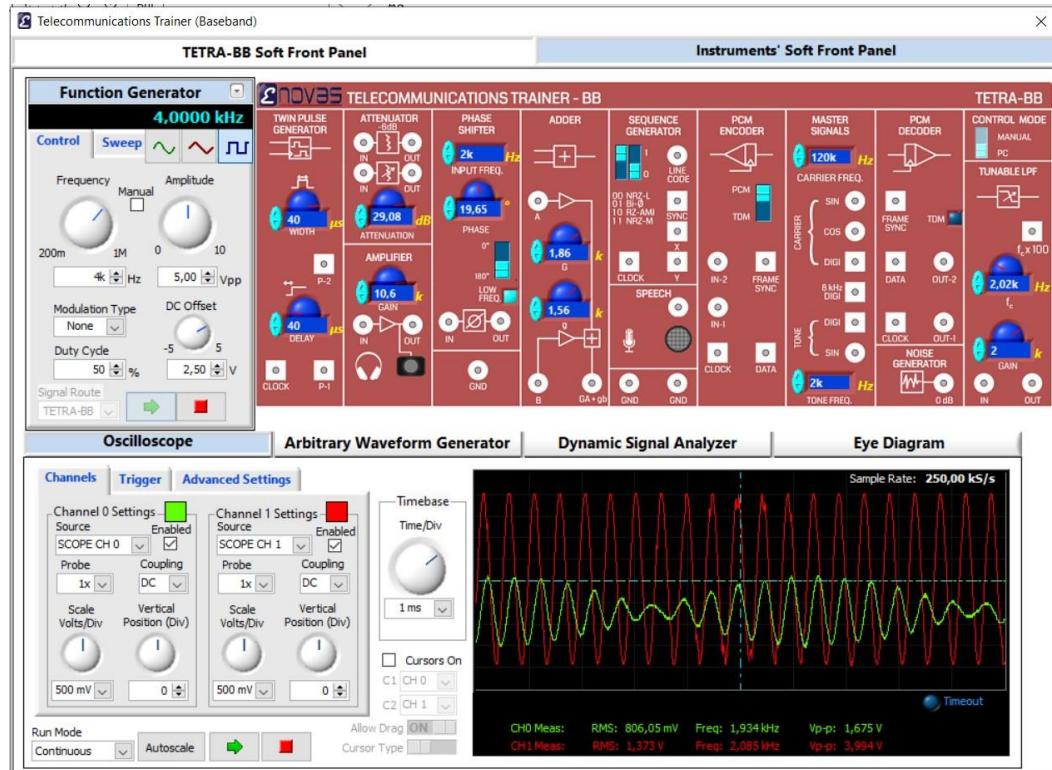


Figure 12 4kHz- sampling

Figure 12 shows the distortion graph when the frequency of the function generator is set to 4kHz. As can be seen in the figure, when we go below 5kHz, the distortion has occurred seriously.

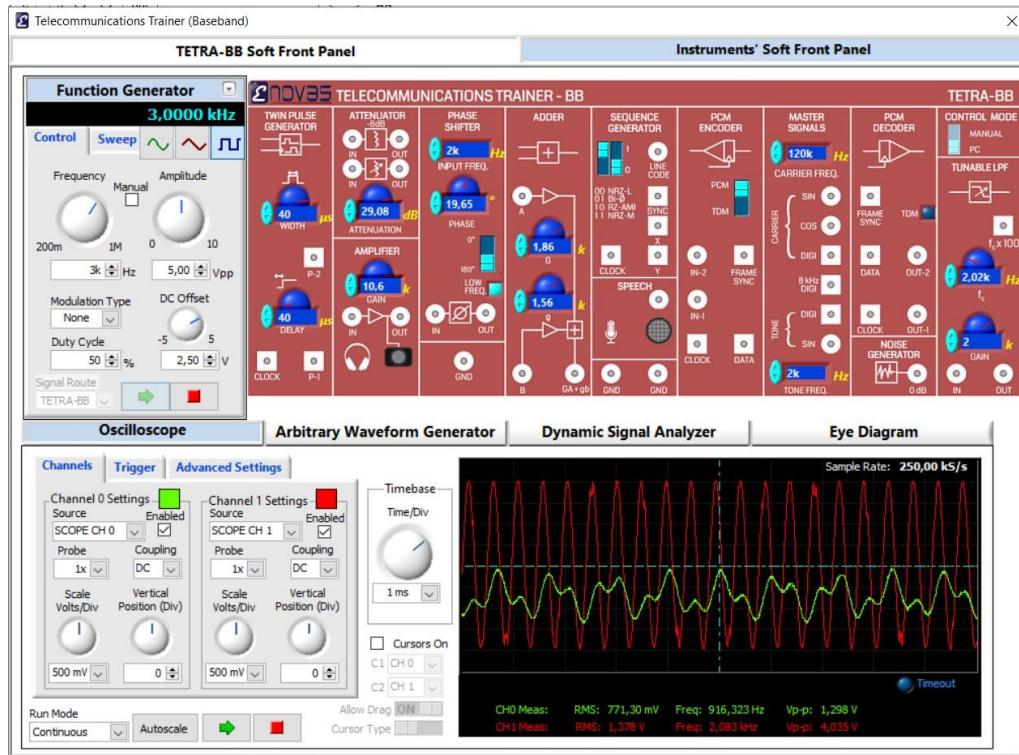


Figure 13 3kHz- sampling

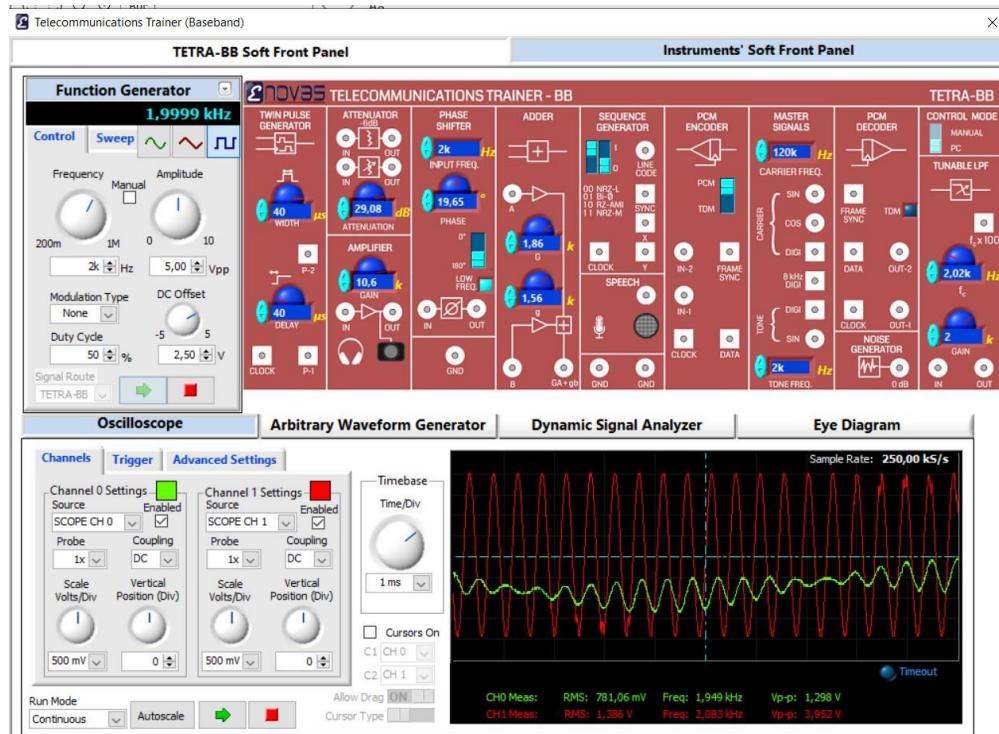


Figure 14 2kHz- sampling

As can be seen in the graphs above, distortions are observed as the sampling frequency decreases. These distortions are called aliasing. Aliasing was observed starting from 4 Khz. The sampling frequency must be greater than or equal to twice the maximum frequency of the signal

4. CONCLUSION

In the experiment, two different sampling methods, natural sampling and sample-and-hold, were discussed and related circuits were realized. In the first method, natural sampling was applied and in the second method, sample-and-hold was applied. Since natural sampling is performed with a pulse train, the sampled signal consists of pulses that take the shape of the sampled signal. Sample and Hold is performed with an impulse train, which takes the value from the signal and stores it. In the rest of the experiment, in order to recover these signals, the signal is passed through a low-pass filter and the importance of the cut-off frequency of the low-pass filter is analyzed. It was also observed that the sampling frequency should be higher than the Nyquist rate in order to recover the signal. In addition, how distortions occur from the images was observed in the experiment and stated in the report.

**GEBZE TECHNICAL UNIVERSITY
ELECTRONICS ENGINEERING DEPARTMENT**

**ELM 366 COMMUNICATIONS LABORATORY
REPORT**

**EXPERIMENT-VI
BPSK AND QPSK**

Prepared by: *Muhammet Koç*

Student Number: *1901022017*

Team and Group Number: Group-3 and Team-2

Team Mates: HALİL ÇİFTÇİ, İSMAİL AKKOYUN

Date of the experiment: *23.05.2024*

Contents

Table of Contents

| | |
|---------------------------------|----|
| Contents..... | 1 |
| 1. GOAL..... | 2 |
| 2. BPSK..... | 2 |
| 2.1. BPSK MODULATION | 2 |
| 2.2. DEMODULATION OF BPSK | 5 |
| 3. QPSK..... | 8 |
| 3.1. QPSK MODULATION | 8 |
| 3.2. QPSK DEMODULATION | 12 |
| 4. CONCLUSION..... | 14 |

1. GOAL

In this experiment, we have learned how to perform binary phase shift keying (BPSK) moduation & demodulation and quadrature phase shift keying (QPSK) modulation & demodulation.

2. BPSK

2.1. BPSK MODULATION

In this section, we generate a BPSK signal and observe the modulated signal and the input signal. Then measure the bit rate and calculate the bandwidth of the modulated signal.

Binary Phase Shift Keying (BPSK) is a digital modulation technique. In this way, the modulated signals can have discrete phase for a given digital input and are constant over a symbol period. A BPSK modulator is shown in Figure 1.

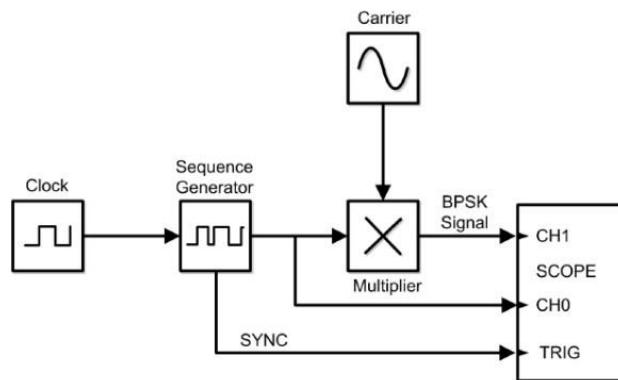


Figure 1 BPSK modulator block diagram

BPSK uses the 1s and 0s of the data stream to switch the carrier between two phases. A BPSK signal changes phase with the change of the digital signal used to generate it.

When the change in logic level causes the phase of the BPSK signal to change, it does so by 180° . For example, when the signal is traveling towards a positive peak, the change in logic level causes it to reverse direction and travel back towards the negative peak (and vice versa). 180° phase means negative amplitude. It is similar to binary amplitude shift keying (BASK).

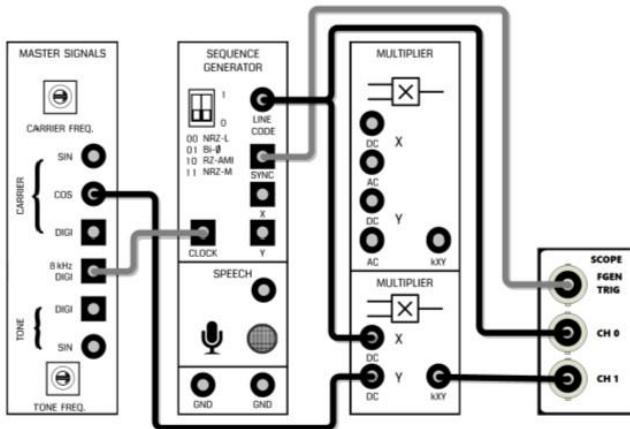


Figure 2 Schematic TETRA-BB

The circuit shown in figure 2 was set up and the modulated signal and the modulated signal were observed with an oscilloscope.

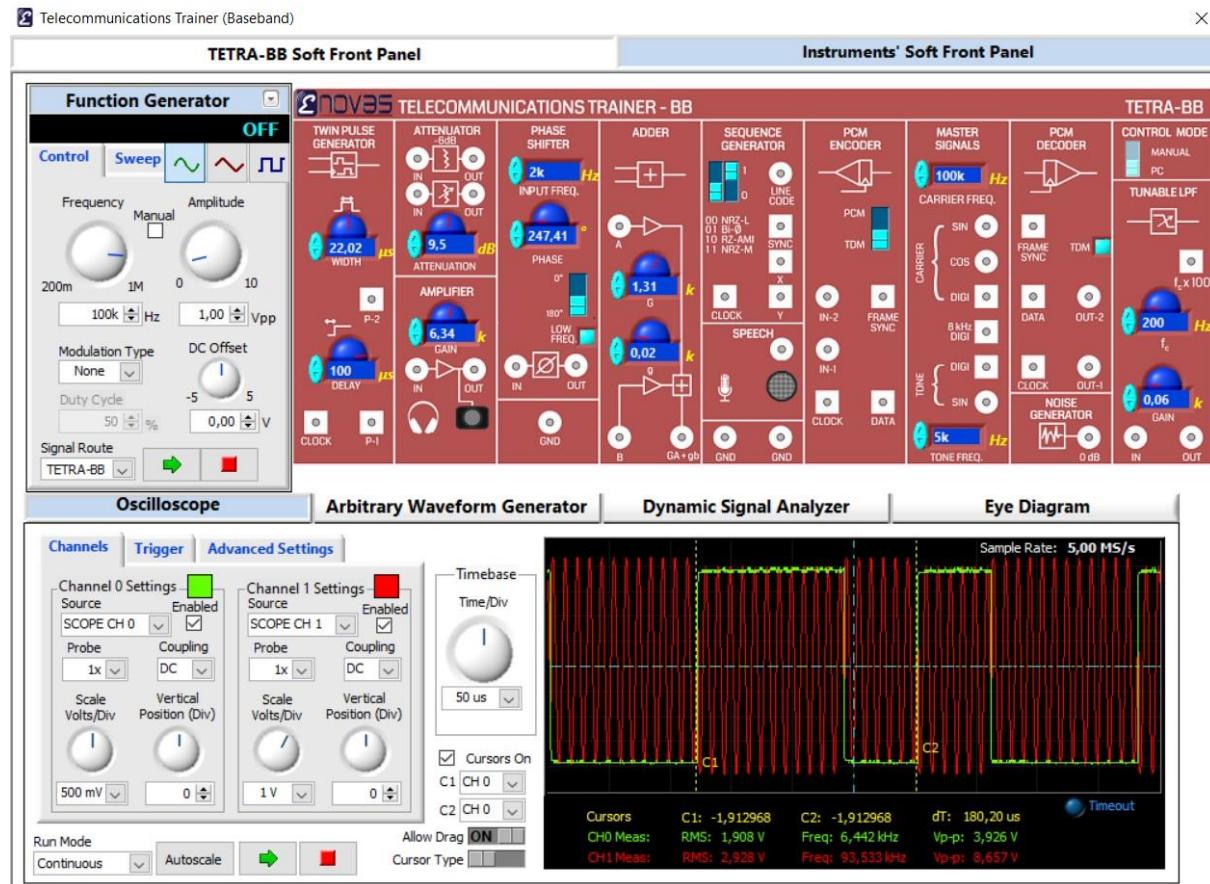


Figure 3 BPSK modulation on the oscilloscope

Figure 3 shows an oscilloscope image of BPSK modulation. For BPSK, each symbol represents 1 bit. When the logic level changes, the phase changes. Figure 3 shows that it takes 180 microseconds for 3 bits(110). Figure 4 shows that 60 microseconds pass for one bit, hence the bit rate (R_b): $\frac{1}{T_s} = \frac{1}{60 \times 10^{-6}} = 16.6 \text{ kbit/sec}$

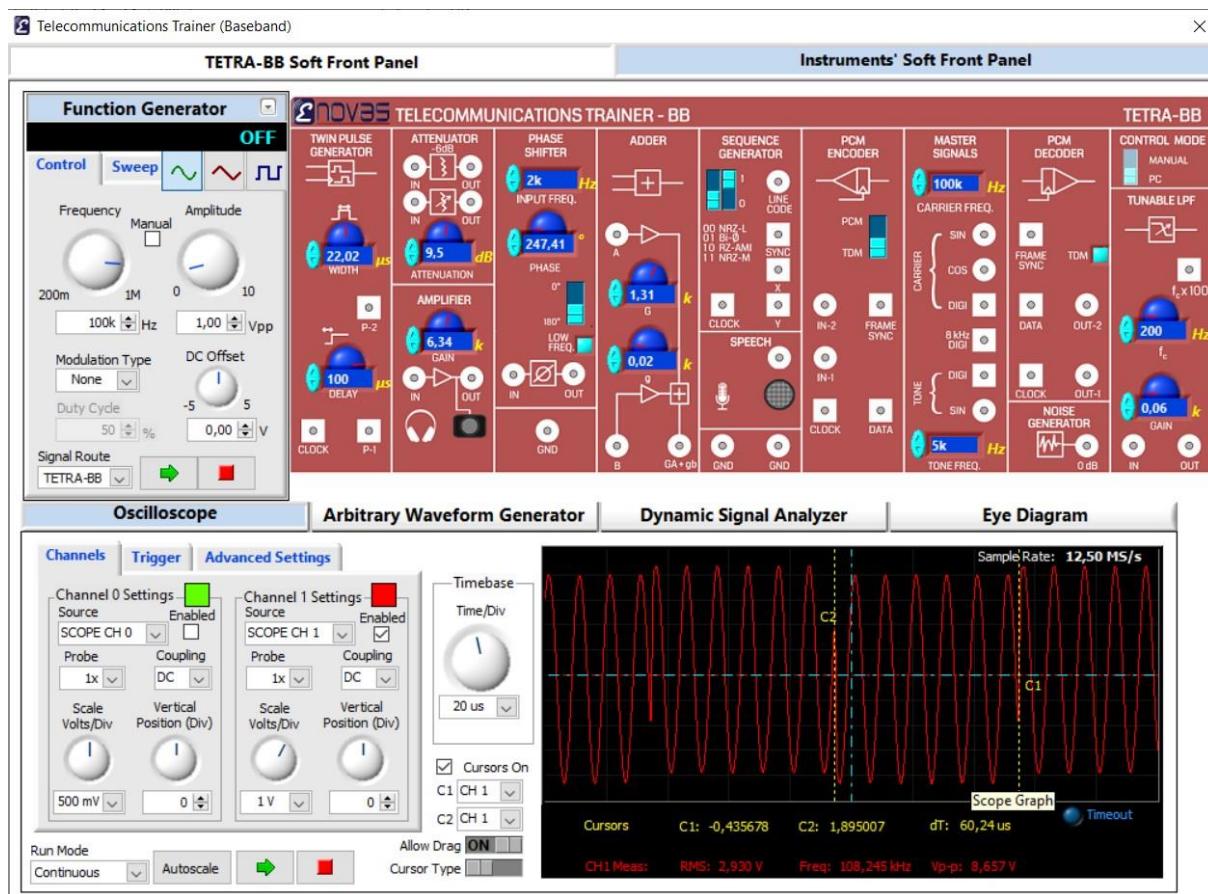


Figure 4 period of one symbol

We use the period information to calculate the bandwidth.

$$B = \frac{1}{2xT_s} = \frac{1}{2x60x10^{-6}} = 8.3\text{kHz}$$

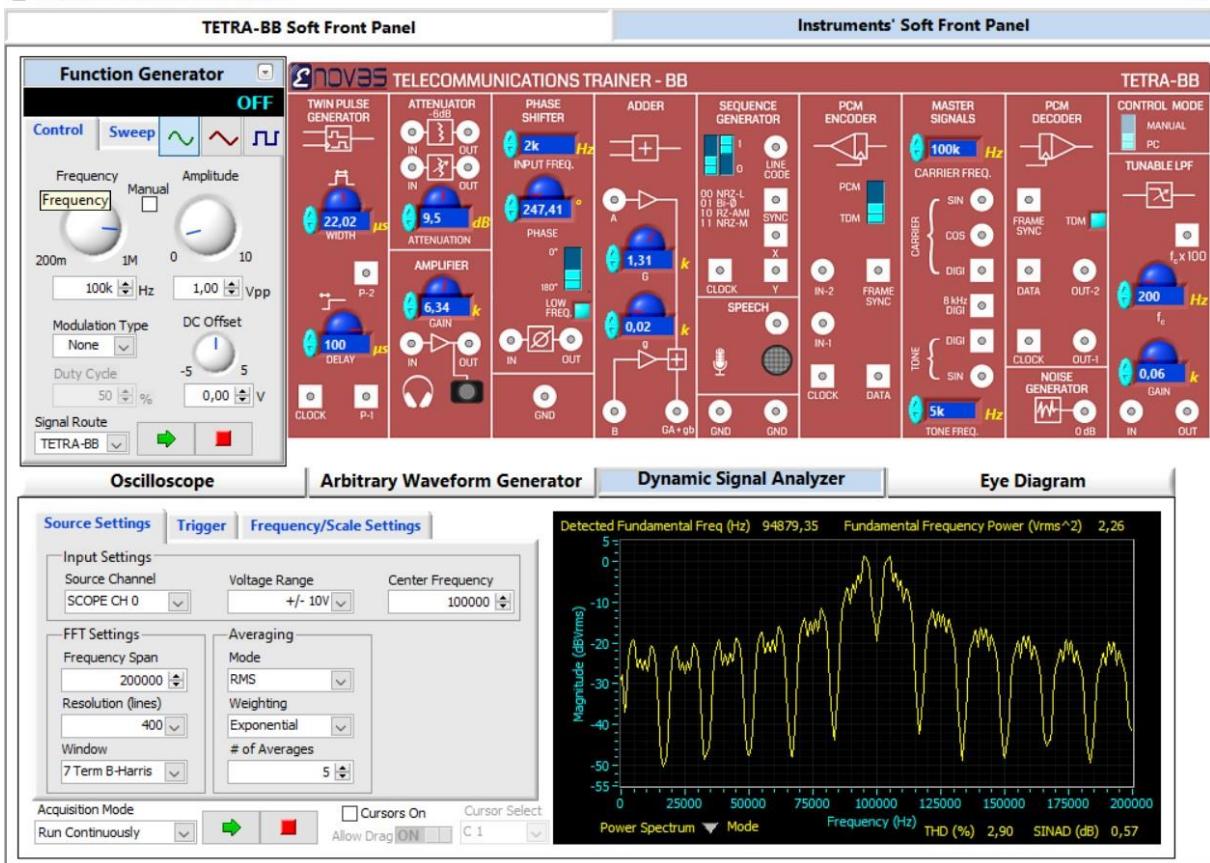


Figure 5 DSA of BPSK modulated signal

Since the carrier frequency of the modulated signal is 100kHz, the Fourier transform produces the sinc signal with a center of 100kHz. The bandwidth can be seen in Figure 5 to be about 30kHz.

2.2. DEMODULATION OF BPSK

The aim of this section is to demodulate a BPSK signal. To observe the demodulated signal and the digital signal at the input of the modulator. For this, a tunable lpf is used and the digital signal is obtained with the output of the comparator. Demodulator shown in figure 6.

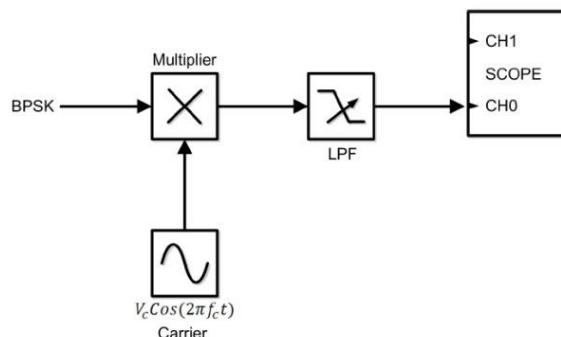


Figure 6 BPSK demodulator block diagram

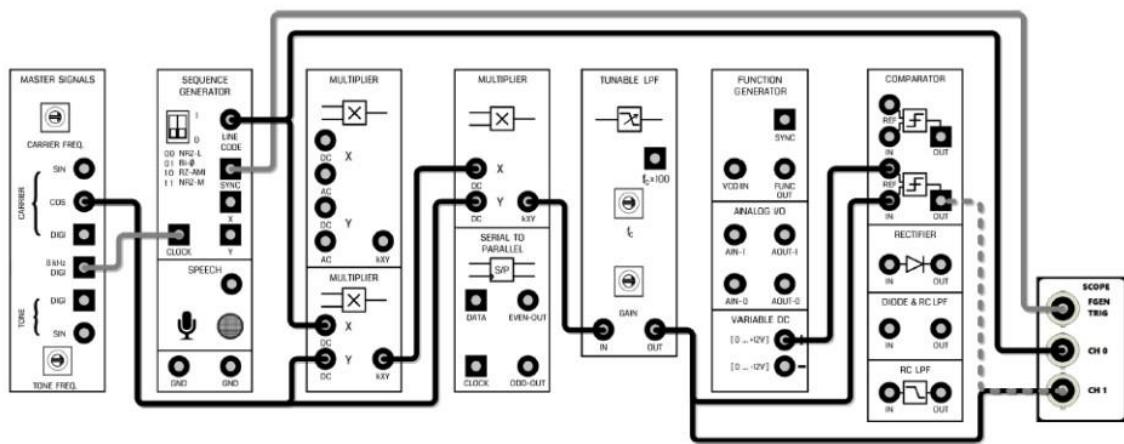


Figure 7 Schematic TETRA-BB

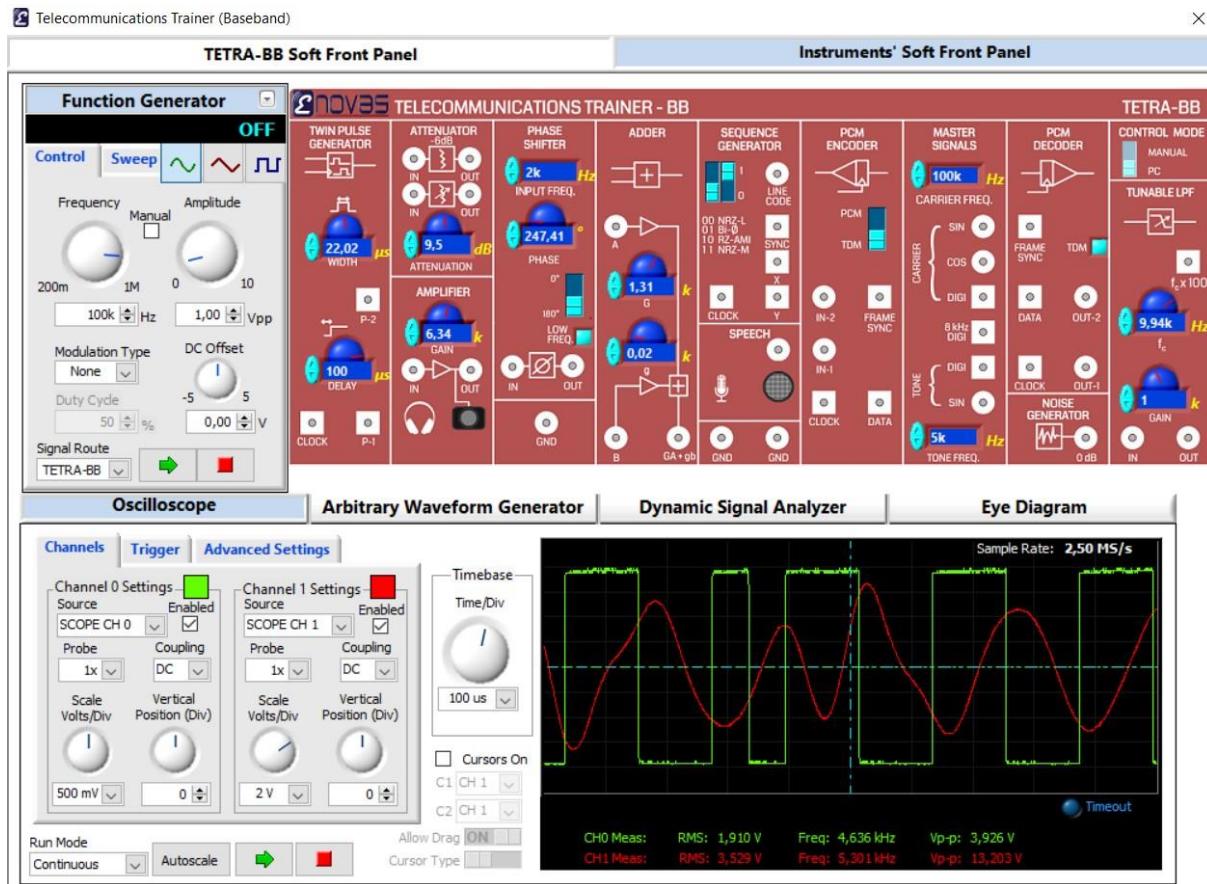


Figure 8 Cutoff frequency of LPF

Figure 8 shows the tunable low-pass filter with the cut-off frequency set to approximately 10kHz.

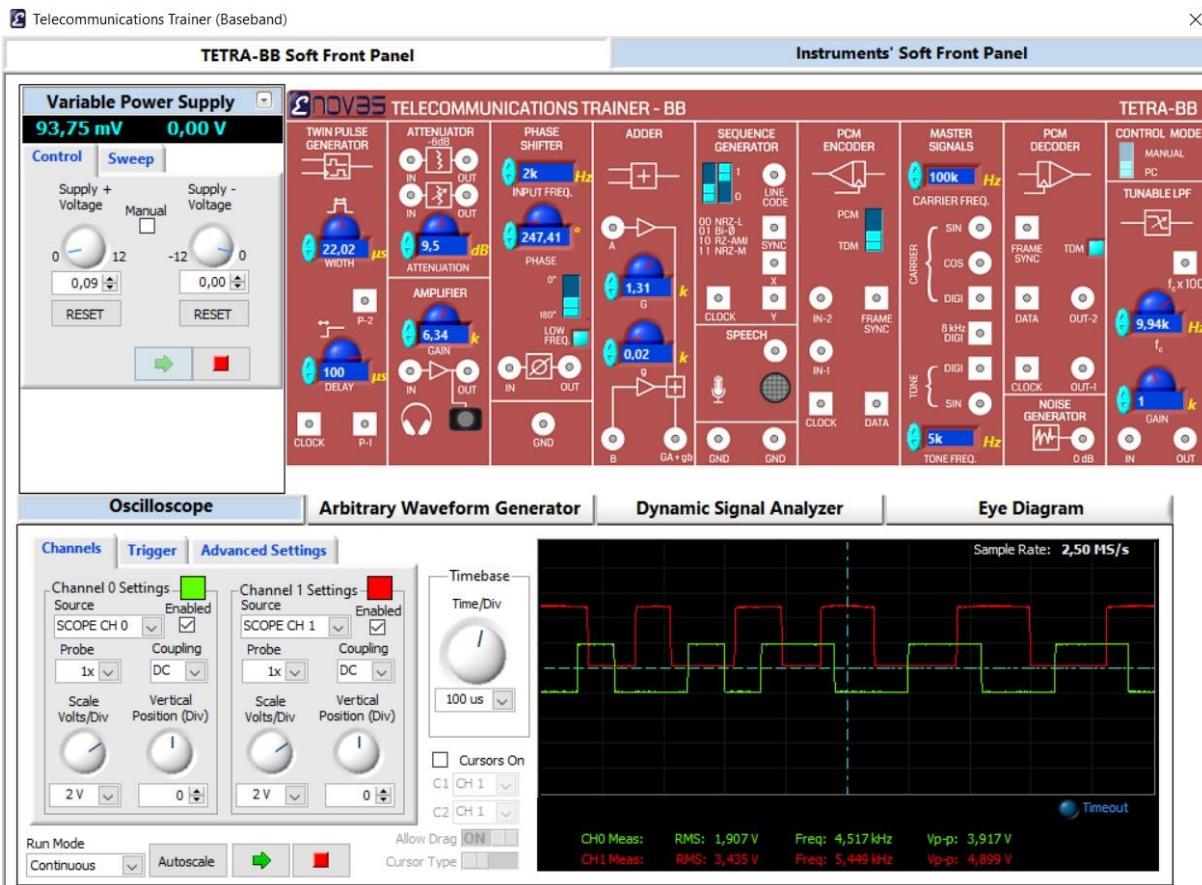


Figure 9 Comparator output

94 mV DC voltage was applied for the comparator. Figure 9 shows the comparator output on an oscilloscope. The digital signal given at the input is obtained with a demodulator.

3. QPSK

3.1. QPSK MODULATION

The aim of this section is to generate a QPSK signal and observe the modulated signal. Finally the bit rate is measured and the bandwidth of the modulated signal is calculated.

BPSK transmissions represent one bit per symbol. In QPSK transmissions each symbol represents two bits.

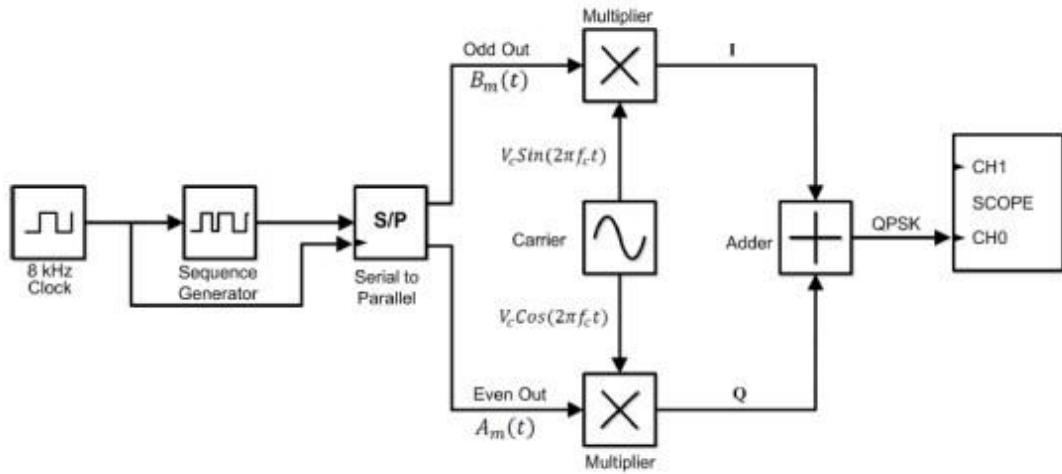


Figure 10 QPSK modulator block diagram

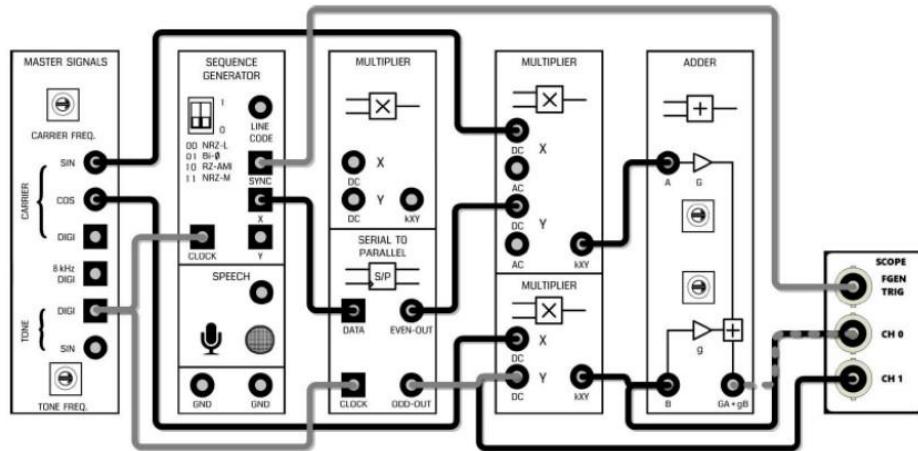


Figure 11 Schematic TETRA-BB

After these operations, the program was run and the input and output graphs were observed from the oscilloscope section.

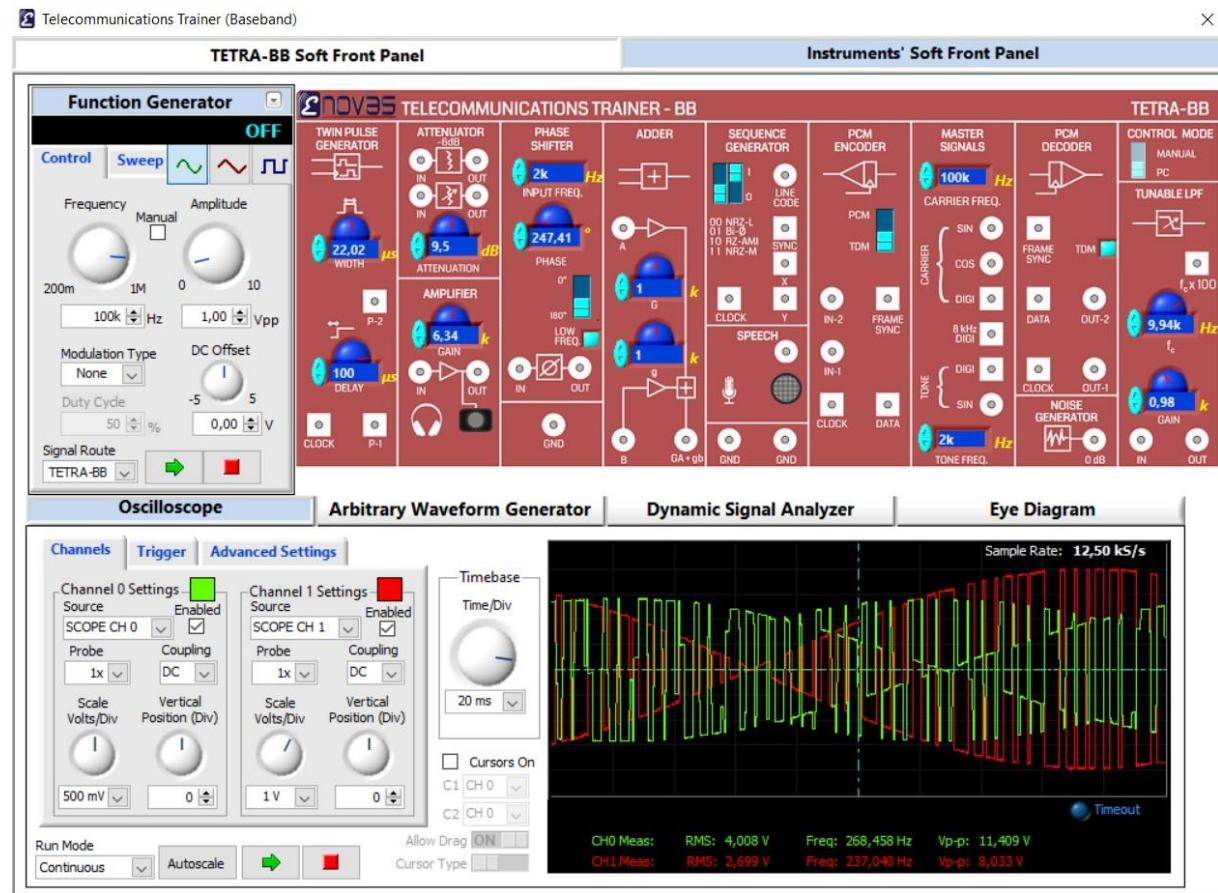


Figure 12 CH0 probe shows modulated QPSK signal

In figure 12, the CH0 probe shows the modulated QPSK signal. The CH1 probe shows the I part of the modulator shown in figure 10, i.e. the odd bits multiplied by the sine carrier signal. The QPSK modulator is similar to the summation of two BPSK signals using carriers with 90 degrees of phase between them.

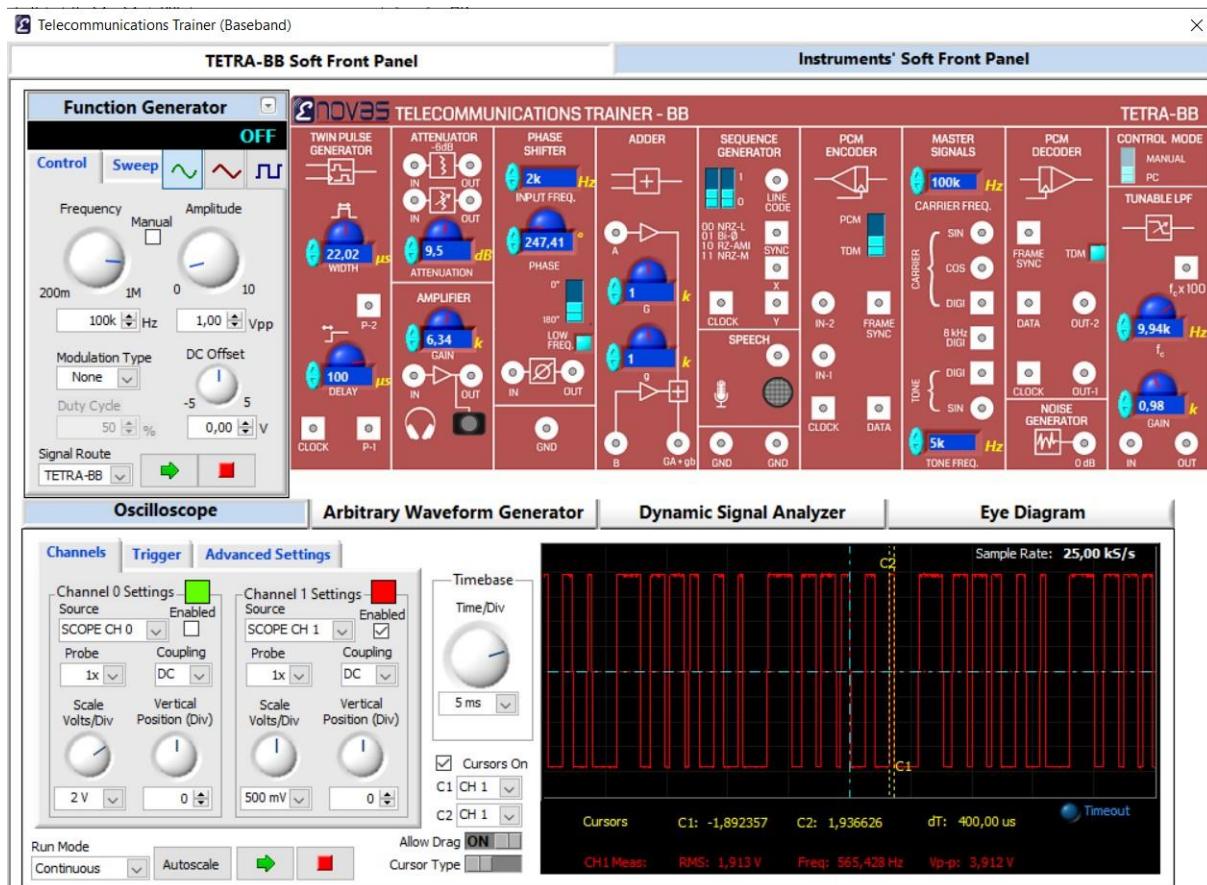


Figure 13 period of one bit

Figure 13 shows that the period for a bit is 400 microseconds when the clock frequency is 5kHz. Since two bits are transmitted with a QPSK symbol, the period is calculated as $T_s=800$ microseconds. Then, the frequency bandwidth $B = 1/(2T_s)$ of the modulated signal was calculated

$$B = \frac{1}{2xT_s} = \frac{1}{2x800x10^{-6}} = 625 \text{ Hz}$$



Figure 14 DSA of QPSK signal

In Figure 14, the same signal is processed on the DSA panel. The DSA panel shows the Fourier Transform of the signal sent to the program.

As can be seen in Figure 14, the following operations were carried out respectively:

- Center Frequency: 100000
- Frequency Span: 20000
- Mode: RMS
- Resolution (lines): 400
- Weighting: Exponential
- Window: 7 Term-B Harris and Averages 5

The bandwidth can be seen in Figure 14 to be about 5kHz.

3.2. QPSK DEMODULATION

The aim of this part is that demodulating an QPSK signal and observing the demodulated signal , I signal and Q signal using oscilloscope.

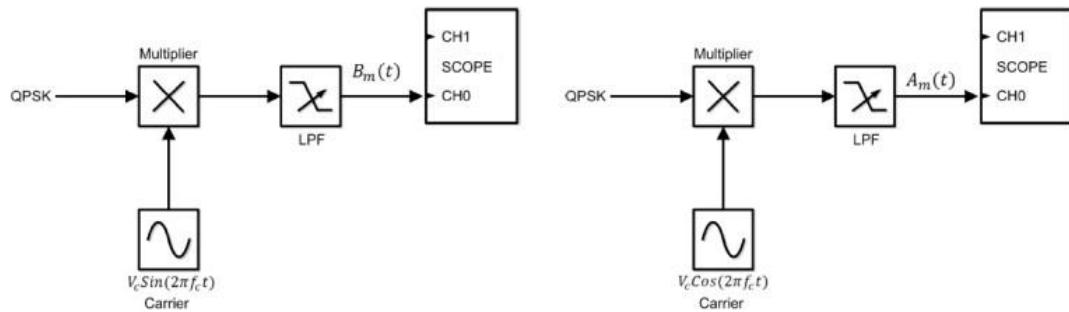


Figure 15 QPSK demodulator block diagram

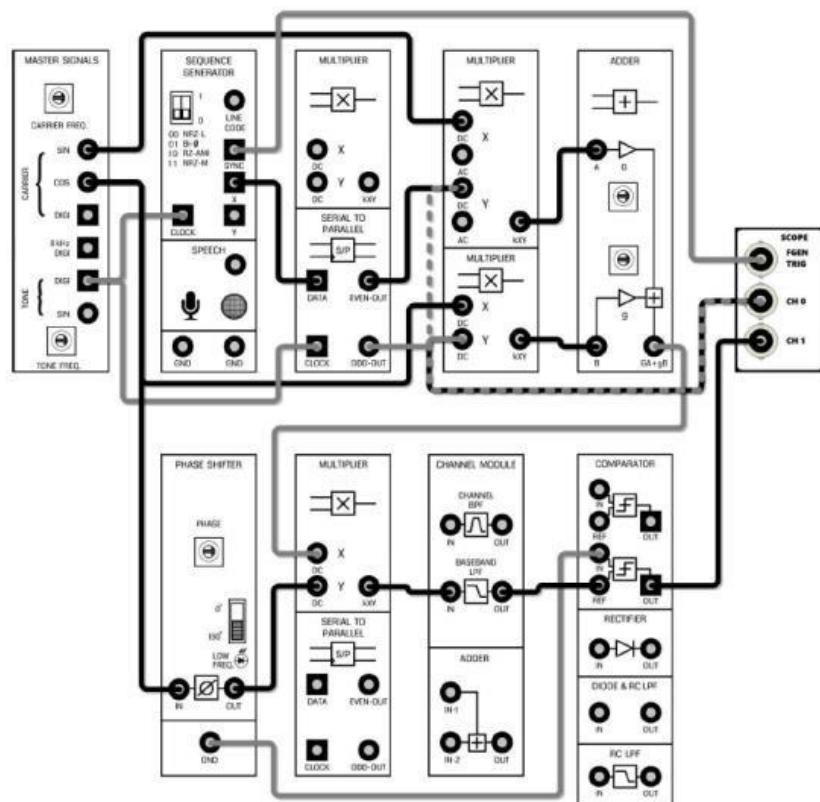


Figure 16 Schematic TETRA-BB

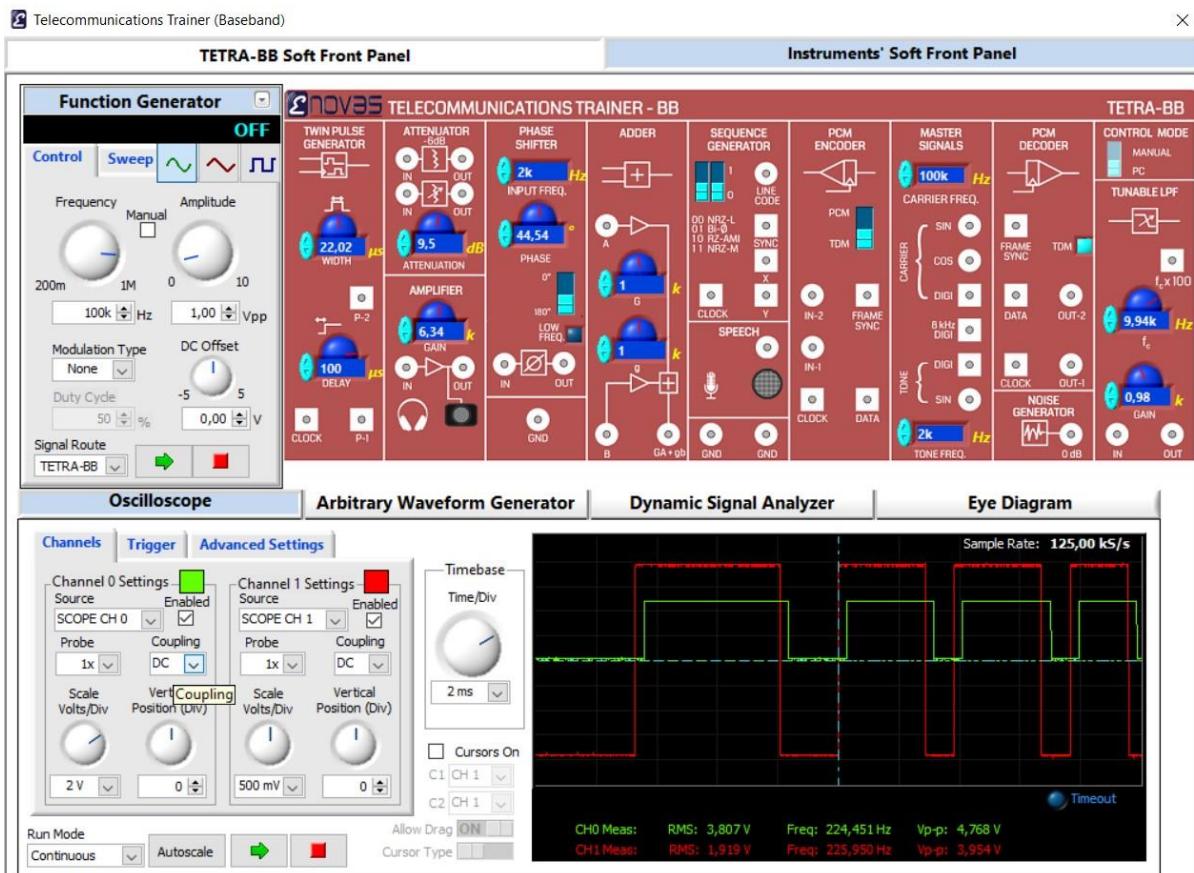


Figure 17 Qpsk demodulated and recovered ODD bits

The odd-out of the serial to parallel block has multiplied by cosinus carrier signal, The I signal has obtained. (phase shifter is 44.54 degrees)

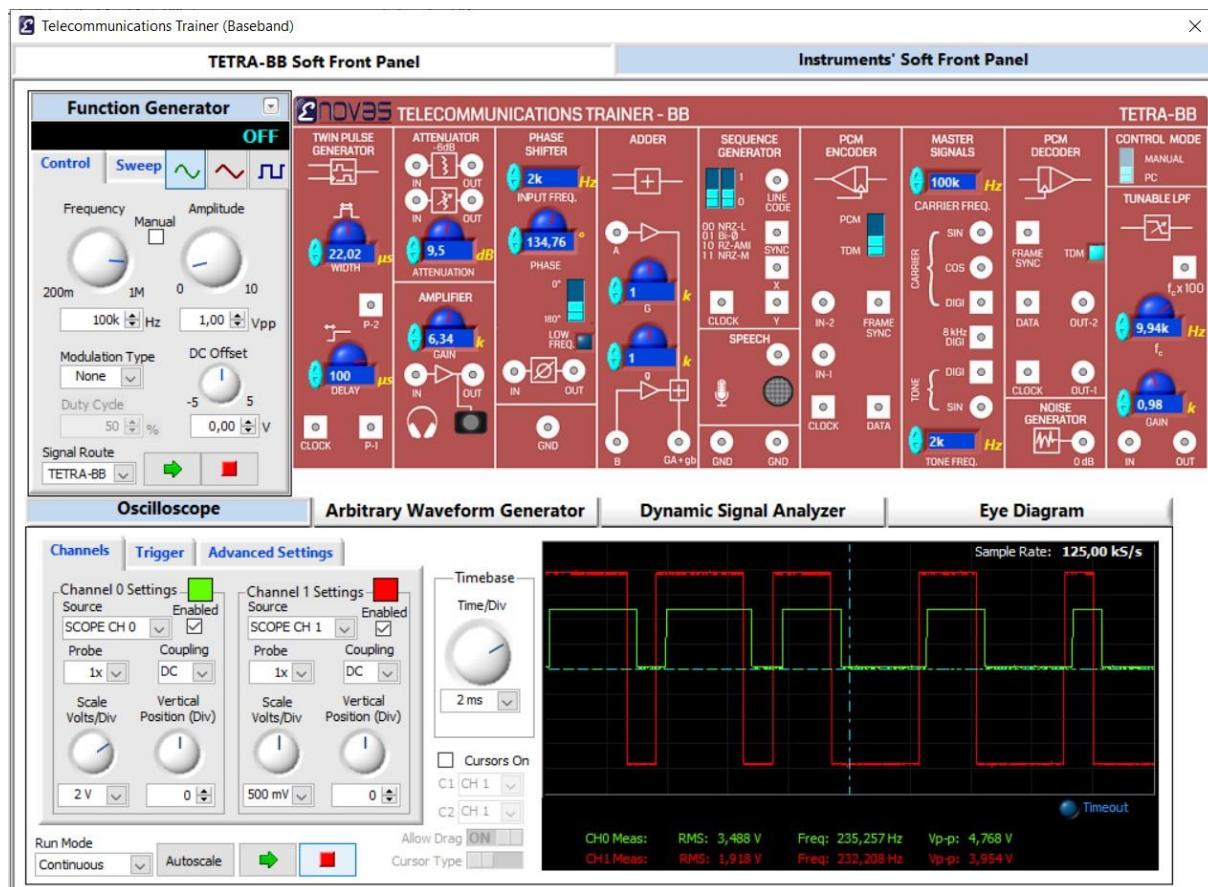


Figure 18 Qpsk demodulated and recovered EVEN bits

The even-out of the serial to parallel block has multiplied by sinus carrier signal, The Q signal has obtained. (phase shifter is 134.76 degrees)

4. CONCLUSION

In the experiment, two different modulation methods, binary phase shifting keying and quadrature phase shifting keying, were considered and the related circuits were realized. In the first method, each symbol represents one bit and in the second method each symbol represents two bits.

BPSK (Binary Phase Shift Keying) and QPSK (Quadrature Phase Shift Keying) are two different modulation techniques used in digital communication systems. Both techniques are commonly used in areas such as radio communications, telecommunications, and wireless communications. However, each has specific uses.

BPSK is a straightforward modulation technique utilized for transmitting digital data. It operates by using two distinct phase states (0 and 180 degrees) to convey information. BPSK is typically favored in scenarios where low data rates and minimal channel bandwidth are sufficient. Some common applications include:

- Wireless communication systems: BPSK is employed in wireless communication standards like Wi-Fi and Bluetooth.
- Low bandwidth communication systems: It is preferred for applications demanding low bandwidth, such as satellite communications and military communications.
- Data storage: BPSK can also be implemented in data storage media, including magnetic tapes and optical discs.

QPSK is a modulation technique used for transmitting digital data, similar to BPSK. However, QPSK uses four different phase states (0, 90, 180, and 270 degrees) to carry information, enabling higher data rates compared to BPSK. Examples of QPSK applications include:

- High-speed data communication: QPSK is employed in systems requiring high data rates, such as digital TV broadcasts, satellite communication, and mobile communication (including 2G, 3G, 4G, and 5G), as well as wireless broadband internet.
- Digital data transmission: QPSK is suitable for applications needing efficient data transmission, such as Ethernet or internet-based data transfer.
- Wireless networks: QPSK is used in wireless networks, including WiMAX (Wireless Interoperability for Microwave Access), Wi-Fi, and 4G/5G networks.

BPSK and QPSK are modulation techniques optimized for communication systems, each suited to different data rates and bandwidth requirements. The selection between BPSK and QPSK depends on the specific needs of the communication system, including factors such as data rate demands, ambient noise levels, channel conditions, and other environmental considerations.

GEBZE TECHNİCAL UNİVERSİTY

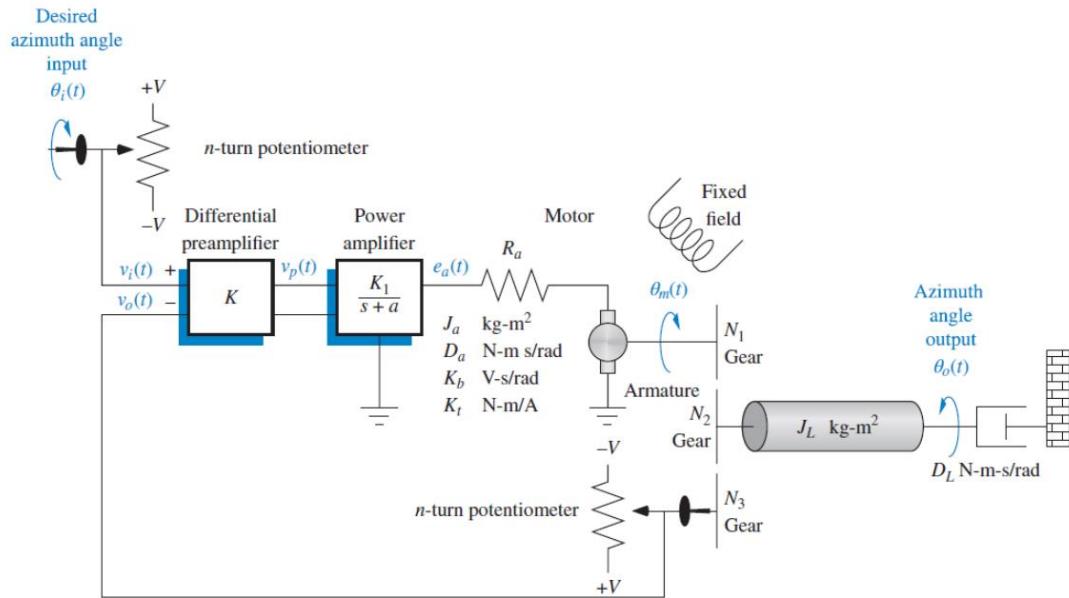
CONTROL SYSTEM THEORY(ME332/ELEC322)
COURSE PROJECT:
THE ANTENNA POSITION CONTROL SYSTEM

MUHAMMET KOÇ
1901022017

PROJECT SUPERVİSOR:
ASSİST. PROF. SAEED LOTFAN

The antenna position control system now serves as an example that summarizes the major objectives of the course. The schematic of this system is shown below.

Schematic



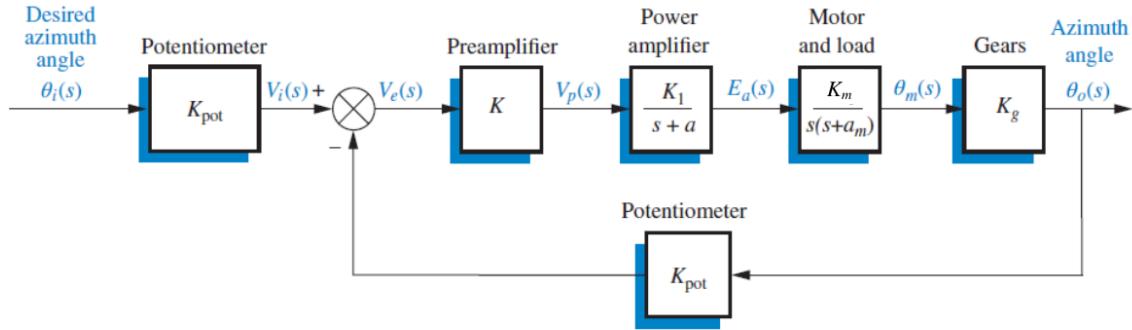
The parameters for this schematic are considered as:

Schematic Parameters

| Parameter | Values |
|-------------------------|--------|
| V | 10 |
| n | 10 |
| K | — |
| K_1 | 100 |
| a | 100 |
| R_a | 8 |
| $J_a \text{ kg-m}^2$ | 0.02 |
| $D_a \text{ N-m s/rad}$ | 0.01 |
| $K_b \text{ V-s/rad}$ | 0.5 |
| $K_t \text{ N-m/A}$ | 0.5 |
| N_1 | 25 |
| N_2 | 250 |
| N_3 | 250 |
| $J_L \text{ kg-m}^2$ | 1 |
| $D_L \text{ N-m s/rad}$ | 1 |

Figure 1:Schematic parameters

Block Diagram



B.1. Considering the block diagram above, by showing the details, has been calculated the values of parameters K_{pot} , K_m , a_m and K_g .

To calculate the K_{pot} parameter, we need to know the transfer function. The function of potentiometer is to convert angle form, $\theta_i(s)$, to voltage form, $V_i(s)$, so the potentiometer gain K_{pot} is (values from figure 1):

$$K_{pot} = \frac{V_i(s)}{\theta_i(s)} = \frac{10}{10\pi} = \frac{1}{\pi} = 0.318$$

K_g is gears ratio parameter. To calculate K_g , we use ratio of N_1 and N_2 values. Hence,

$$K_g = \frac{\theta_o(s)}{\theta_m(s)} = \frac{N_1}{N_2} = \frac{25}{250} = \frac{1}{10} = 0.1$$

K_m and a_m are parameters in block motor and load. Motor and load block of transfer function is $\frac{\theta_m(s)}{E_a(s)}$. Hence,

$$\frac{\theta_m(s)}{E_a(s)} = \frac{K_m}{s(s + a_m)}$$

$$E_a(s) = R_a I_a + V_b$$

$$input current = I_a(s) = \frac{T_m(s)}{K_t}$$

$$T_m(s) = (J_m s^2 + D_m s) \theta_m(s)$$

$$V_b(s) = K_b s \theta_m(s)$$

$$\frac{R_a(J_m s^2 + D_m s) \theta_m(s)}{K_t} + K_b s \theta_m(s) = E_a(s)$$

$$\frac{\theta_m(s)}{E_a(s)} = \frac{\left(\frac{K_t}{R_a J_m}\right)}{s(s + \frac{D_m}{J_m} + \frac{K_b K_T}{R_a J_m})}$$

$$Jm = J_a + J_L \left(\frac{N_1}{N_2} \right)^2 = 0.02 + 0.1^2 = 0.03$$

$$D_m = D_a + D_L \left(\frac{N_1}{N_2} \right)^2 = 0.01 + 0.1^2 = 0.02$$

$$R_a = 8, K_b = K_t = 0.5$$

$$\frac{\theta_m(s)}{E_a(s)} = \frac{\frac{0.5}{0.24}}{s(s + (\frac{0.02}{0.03} + \frac{0.25}{0.24}))} = \frac{2.083}{s(s + 1.708)}$$

So that are found as $K_m = 2.083$ and $a_m = 1.708$.

B.2. Used root locus technique to show the range of K for stability of the system, repeated this problem based on bode diagrams.

Open loop transfer function:

$$G(s) = \frac{K_1}{s + a} \times \frac{K_m}{s(s + a_m)} \times K_g = \frac{100}{s + 100} \times \frac{2.083}{s(s + 1.708)} \times 0.1 = \frac{20.83}{s^3 + 101.708s^2 + 170.8s}$$

$$1 + KG(s)H(s) = 0$$

$$1 + K \left(\frac{20.83}{s^3 + 101.708s^2 + 170.8s} \right) 0.318 = 0$$

$$s^3 + 101.708s^2 + 170.8s + K6.623 = 0$$

We use Routh–Hurwitz Criterion to find the range of K for stability of the system.

$$\begin{array}{c|cc} s^3 & 1 & 170.803 \\ s^2 & 101.708 & 6.623K \\ s^1 & \frac{17362.37 - 6.623K}{101.708} & 0 \\ s^0 & 6.623K & 0 \end{array}$$

For stability of the system must be no sign changes at routh table. Hence,

$$6.623K > 0 \quad \& \quad K > 0$$

$$17362.37 - 6.623K > 0 \quad \& \quad K < 2621.53$$

The range of K for stability: $0 < K < 2621.53$

Closed loop transfer function:

$$\frac{KG(s)}{1 + KG(s)H(s)}$$

$$\frac{K \left(\frac{20.83}{s^3 + 101.708s^2 + 170.8s} \right) 0.318}{\left(1 + K \left(\frac{20.83}{s^3 + 101.708s^2 + 170.8s} \right) 0.318 \right)} = \frac{K6.623}{s^3 + 101.708s^2 + 170.8s + K6.623}$$

When drawing root locus graph, we was used matlab. Matlab codes which used was shown below.

For this graph we choose K value as a 100 so;

```
antennapositioncontrol.m
1 numgh=[662.3]; %Define numerator
2 denumgh=[1, 101.708, 170.8, 662.3]; %Define denominator
3 Gs=tf(numgh,denumgh); %create G(s)H(s) display
4 rlocus(Gs); %Draw Root Locus
5 % z=0.1:0.1:0.8; %Define damping ratio values
6 %wn=0:1:10; %Define natural frequency response
7 %sgrid(z,wn); % Generate Damping ratio and natural frequency grid lines for root locu
```

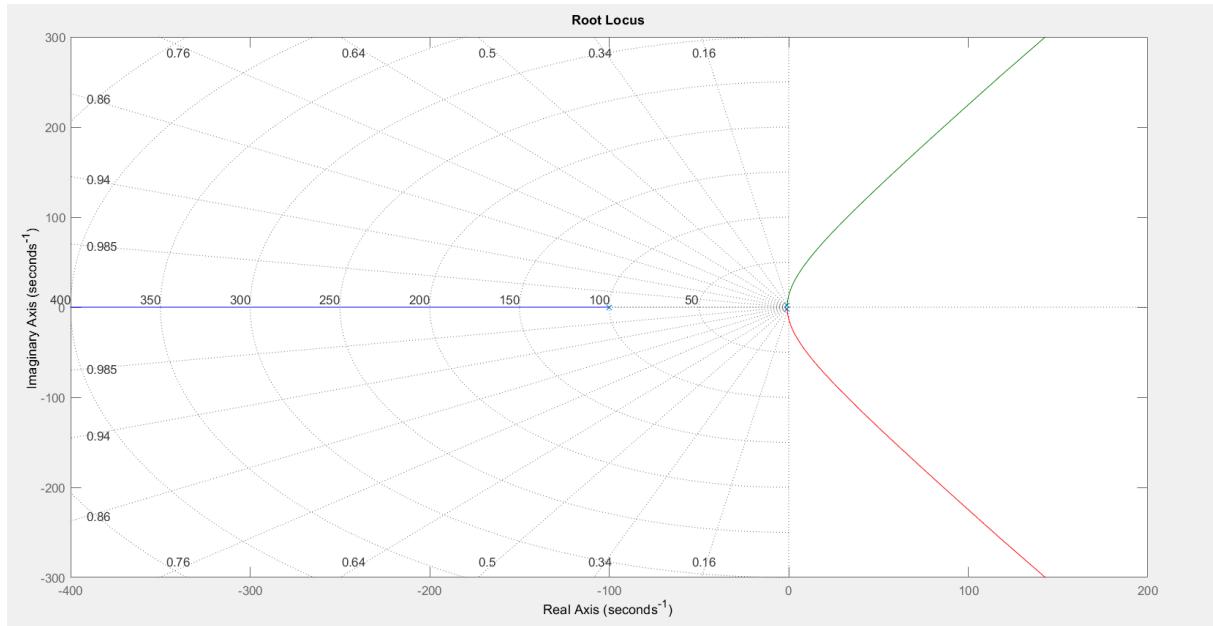


Figure 2: Root Locus Graph

Codes for bode diagram:

```

antennapositioncontrol.m      Bodegraph.m
1 numgh=[662.3]; %Define numerator
2 denumgh=[1, 101.708, 170.8, 662.3]; %Define denominator
3 Gs=tf(numgh,denumgh); %create G(s)H(s) display
4 bode(Gs); %Draw Root Locus
5 grid on %turn on grid for bode graph

```

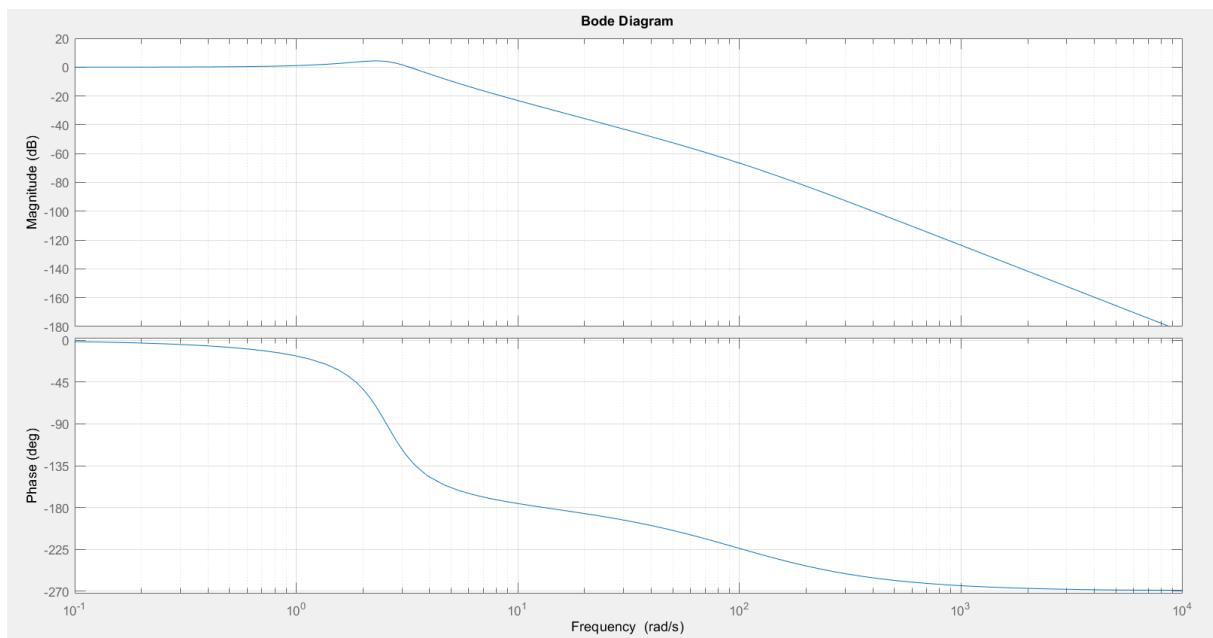


Figure 3: Bode diagram

B.3. Designed a PID controller so that the system can operate with a peak time that is two-thirds that of the uncompensated system at 35% overshoot and with zero steady-state error for a step input.

Matlab codes of designing:

The screenshot shows a MATLAB code editor window with the file 'piddesigning.m' open. The code is a script for designing a PID controller. It starts by defining the plant's transfer function using given values of K and transfer function coefficients. It then calculates the step response of the uncompensated system. The script then determines the desired peak time (two-thirds of the uncompensated peak time) and desired overshoot (35%). It calculates the damping ratio and natural frequency based on these requirements. The crossover frequency is also determined. The script uses the 'pidtune' function to find a PID controller (C) that meets the specified performance criteria. Finally, it plots the step responses of the uncompensated and compensated systems for comparison.

```
1 %Defined the plant's transfer function using the given values of
2 %K and the transfer function coefficients.
3 K = 100;
4 numgh = K * 6.623;
5 dengh = [1, 101.708, 170.8, K * 6.623];
6 Gs = tf(numgh, dengh);
7
8 step_info = stepinfo(G);% Obtained the step response of the uncompensated system
9
10 %We want the peak time of the compensated system to be two-thirds that of
11 % the uncompensated system and an overshoot of 35%.
12 peak_time_uncompensated = step_info.PeakTime;
13 desired_peak_time = (2/3) * peak_time_uncompensated;
14 desired_overshoot = 0.35;
15
16 % Calculate damping ratio and natural frequency
17 zeta = -log(desired_overshoot) / sqrt(pi^2 + log(desired_overshoot)^2);
18 omega_n = 4 / (zeta * desired_peak_time);
19
20 % Phase margin of desired overshoot
21 phi = atan(sqrt(1 - desired_overshoot^2)) / (2 * desired_overshoot);
22
23 %The crossover frequency from desired peak time
24 wcf = 2 * pi / desired_peak_time;
25
26 optns = pidtuneOptions('CrossoverFrequency', wcf, 'PhaseMargin', phi);
27
28 C = pidtune(Gs, 'PID', optns);
29
30 T = feedback(C * Gs, 1);
31
32 %response:
33 figure;
34 step(Gs);
35 hold on;
36 step(T);
37 legend('Uncompensated', 'Compensated');
38 grid on;
39 step_info_uncompensated=stepinfo(Gs);
40 step_info_compensated = stepinfo(T);
41 disp('Uncompensated:');
42 disp(step_info_uncompensated);
43 disp('Compensated:');
44 disp(step_info_compensated);
45
```

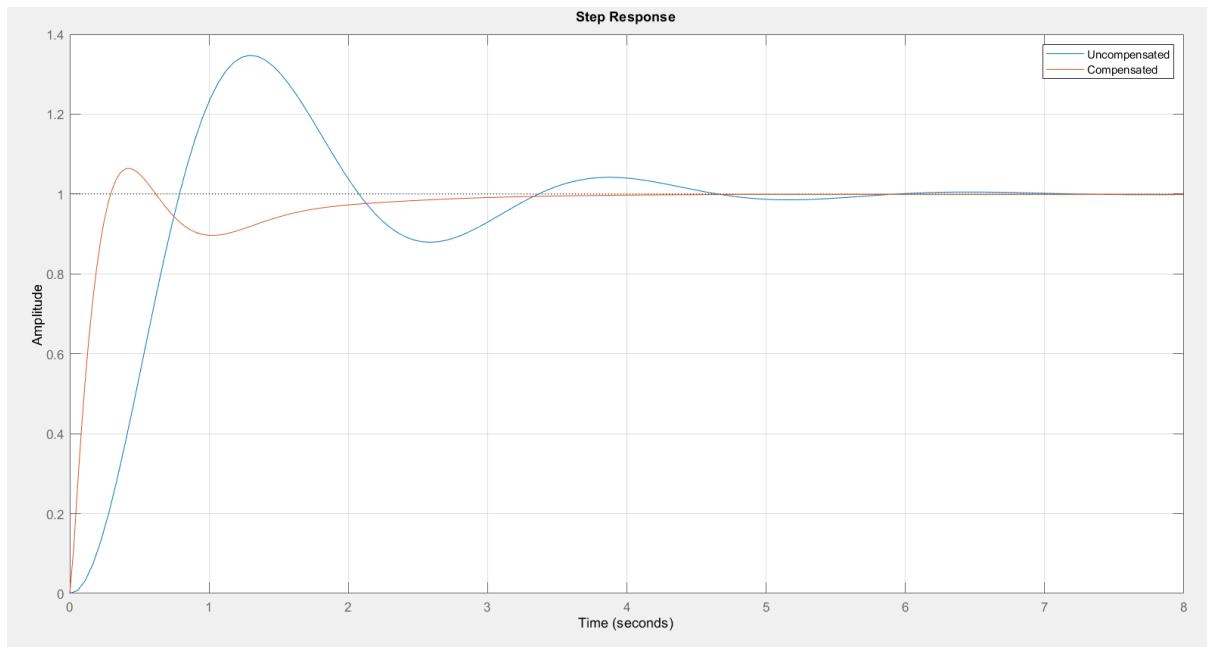


Figure 4: Difference of uncompensated and compensated systems

```
>> piddesigning
Uncompensated:
    RiseTime: 0.5255
    TransientTime: 4.3523
    SettlingTime: 4.3523
    SettlingMin: 0.8793
    SettlingMax: 1.3473
    Overshoot: 34.7335
    Undershoot: 0
    Peak: 1.3473
    PeakTime: 1.2911

Compensated:
    RiseTime: 0.2050
    TransientTime: 2.2871
    SettlingTime: 2.2871
    SettlingMin: 0.8964
    SettlingMax: 1.0642
    Overshoot: 6.4223
    Undershoot: 0
    Peak: 1.0642
    PeakTime: 0.4125
```

Figure 5: Parameters of systems

B.4. A new schematic and block diagram of the compensated system was drawn by realizing the physics of the PID controller.

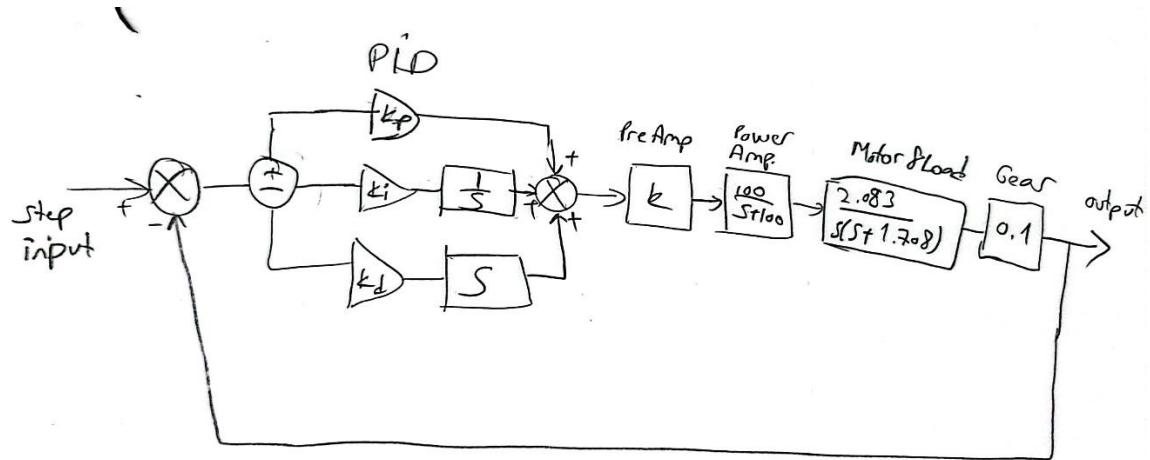


Figure 6: Block diagram of compensated system