**1-)**

**MATLAB KODUM:**

clear;

clc;

% A

f\_s = 400; %örnekleme frekansı

t = 0:1/f\_s:1-1/f\_s;

x = sinc(20\*t-10);

xf = fftshift(fft(x)); %frekans domeninde x

f = linspace(-200,200,f\_s);

figure('name',"Zaman ve Frekans Domeninde X");

subplot(2,1,1)

plot(t,x),xlabel("Zaman"),ylabel("Genlik"), title("Zamanda X");

subplot(2,1,2)

plot(f,abs(xf)),xlabel("Frekans"),ylabel("Genlik"), title("Frekansta X");

% B

fd=40; %impuls katarı frekansı

fsd=fd\*10;

td=0:1/fsd:1-1/fsd; % zaman aralığı

dirac = zeros(size(td));

dirac(1:fsd/fd:end) = 1;

xs = x.\*dirac;

figure('name',"x\_s");

plot(t,xs),xlabel("Zaman"),ylabel("Genlik"), title("Xs");

% C

x1 = 0:1/f\_s:1;

h1 = rectangularPulse((x1-0.0025)/0.005);%pam1 için süzgeç

xpam1 = conv(xs,h1);

xpam1 = xpam1(1:length(xs));

tpam1 = 0:1/f\_s:1-1/f\_s;

xfpam1 = fftshift(fft(xpam1));

figure('name',"Zaman ve Frekans Domeninde Xpam");

subplot(2,1,1)

plot(tpam1,xpam1),xlabel("Zaman"),ylabel("Genlik"), title("Zamanda Xpam1");

subplot(2,1,2)

plot(f,abs(xfpam1)),xlabel("Frekans"),ylabel("Genlik"), title("Frekansta Xpam1");

% D

x2 = 0:1/f\_s:1;

h2 = rectangularPulse((x2-0.005)/0.01); %pam2 için süzgeç

xpam2 = conv(xs,h2);

xpam2 = xpam2(1:length(xs));

tpam2 = 0:1/f\_s:1-1/f\_s;

xfpam2 = fftshift(fft(xpam2));

figure('name','Zaman ve Frekans Domeninde Xpam');

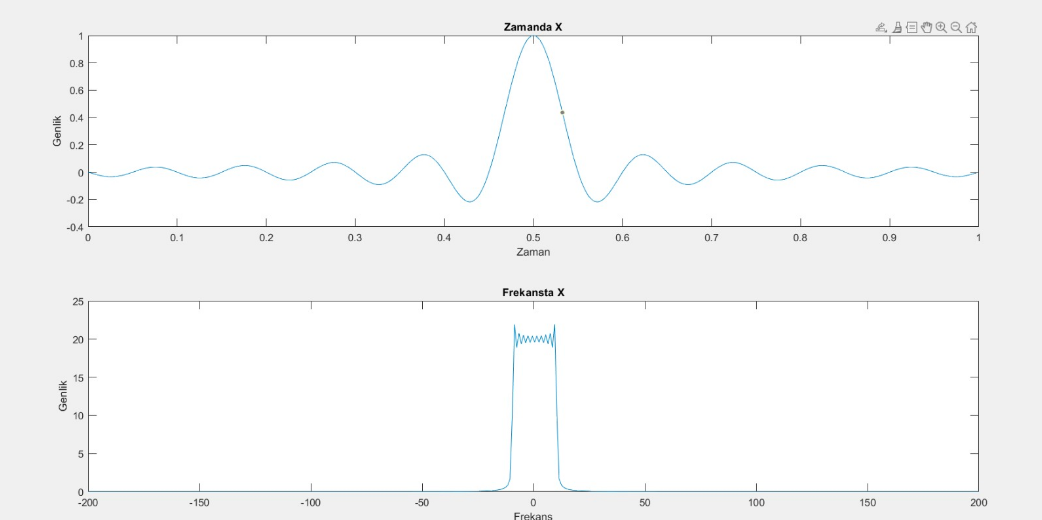
subplot(2,1,1)

plot(tpam2,xpam2),xlabel("Zaman"),ylabel("Genlik"), title("Zamanda Xpam2");

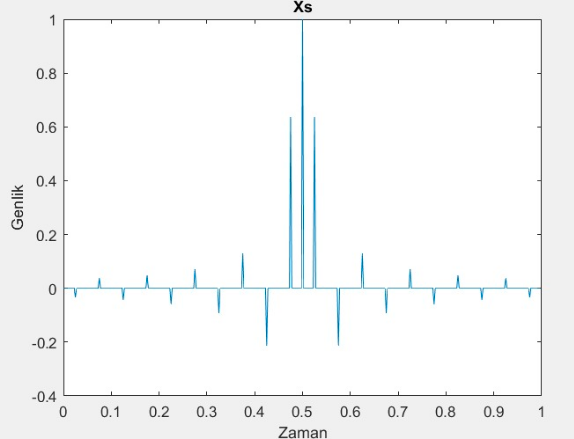
subplot(2,1,2)

plot(f,abs(xfpam2)),xlabel("Frekans"),ylabel("Genlik"), title("Frekansta Xpam2");

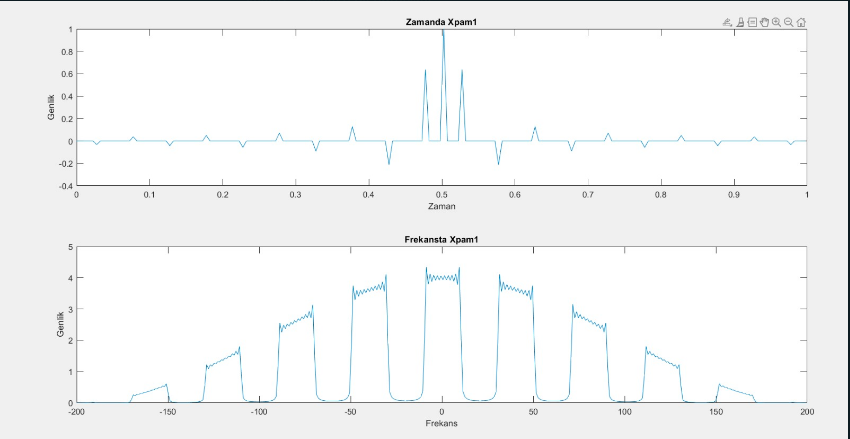
**a şıkkındaki yazılan koda göre alınan çıktı:**

****

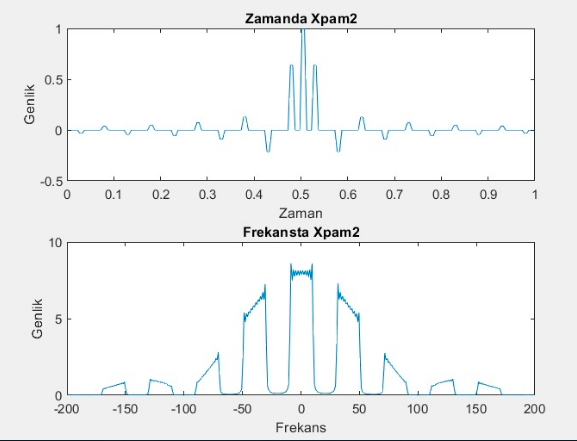
**b şıkkındaki yazılan koda göre alınan çıktı:**

****

**C şıkkındaki yazılan koda göre alınan çıktı:**

****

**D şıkkındaki yazılan koda göre alınan çıktı:**

****

**e-)** D şıkkındaki durum seçilmelidir, d şıkkındaki filtre daha iyidir. D şıkkındaki Xpam2 c şıkkındaki Xpam1’ e göre daha kararlıdır, c şıkkına göre frekansta daha az yer kaplar ve daha çabuk sönümlenir. Bundan dolayı d şıkkındaki durum seçilmelidir.

**2-) Konu: Sayısal Haberleşmede Kanal Kodlama (Channel Coding)**

1. **Kodlama Teorisi Nedir?**

Kodlama teorisindeki amaç tasarımların üzerindeki uygunluğunu ve [kodların](https://tr.wikipedia.org/wiki/Kod) özelliklerini incelemektir. [Kriptografi](https://tr.wikipedia.org/wiki/Kriptografi), [veri sıkıştırma](https://tr.wikipedia.org/wiki/Veri_s%C4%B1k%C4%B1%C5%9Ft%C4%B1rma), veri iletimi, veri depolama vb. uygulamalarda kodlar kullanılabilir. En optimize şekilde (güvenli olması, verimli olması vb.) veri aktarımı yapılabilmesi için kodların belli kontrollerden geçmesi gerekir, bu ortamı sağlamak amacıyla kodlar, belli bilimsel disiplinler tarafından kontrole tabi tutulur. Bu bilim disiplinlerine matematik mühendisliği ve bilgisayar mühendisliği örnek verilebilir. Kodlardaki fazlalığın kaldırılması, iletilen verilerdeki hataların düzeltilmesini ve tespit edilmesi az önce bahsedilen bilimsel disiplinlerin katkısıyla gerçekleşir.

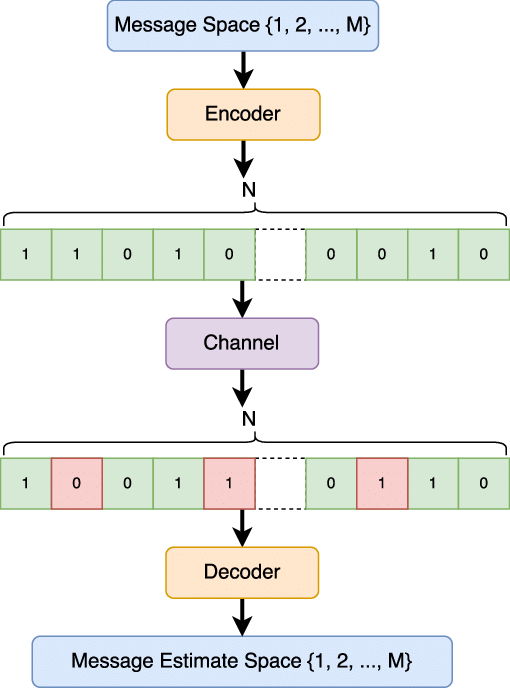
Yukarıdaki kısaca bahsedilen kodlamanın 4 tipi vardır:

**1-) Veri sıkıştırma (Kaynak kodlama)**

**2-) Hata kontrolü (Kanal kodlaması)**

**3-) Kriptografik kodlama**

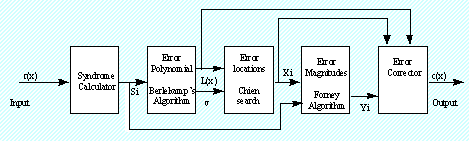
**4-) Hat kodlama**

Aşağıdaki ***Şekil 1***’de haberleşme kanalına ait kodlama teorisinin görünümü verilmiştir.

***Şekil 1***

Verileri daha optimize (içindeki bulunan fazlalıkları kaldırma) bir halde kullanmak ve iletmek için veri sıkıştırma metodu kullanılır. Örnek vermek gerekilirse her birimizin kullandığı WINRAR dosyaları bu yöntemden sayılabilir.  [RAR veri sıkıştırması](https://tr.wikipedia.org/wiki/Zip) internet trafiğini azaltmak ve verilerin gönderilecek yere daha hızlı iletme gibi amaçlarla verileri küçültür.

Veri gönderimi işlemi anında, mevcut kanaldaki bozucu etkenlere karşı daha güvenli hale getirmek için hata düzeltme işleminde fazladan bitler eklenir. Bu işlem gerekli şartları sağlanıp yapılırsa herhangi bir kullanıcı kullanılan hata düzeltme işlemlerinin farkına varamayabilir. Bu durumda örnek verilmesi gerekirse müzik veya film CD’lerindeki tozların ve çiziklerin bozucu etkilerini ortadan kaldırmak için için “Reed-Solomon” hata düzeltme kodu kullanılır. 1960 yılında Irving S. Reed ve Gustave Solomon tarafından çıkartılan bu uygulamada iletim kanalına direkt olarak CD'nin kendisi denilebilir. Aşağıda belirtilen blok diyagramları uygulamanın nasıl gerçeklediğini belirtir.



***Şekil 2***

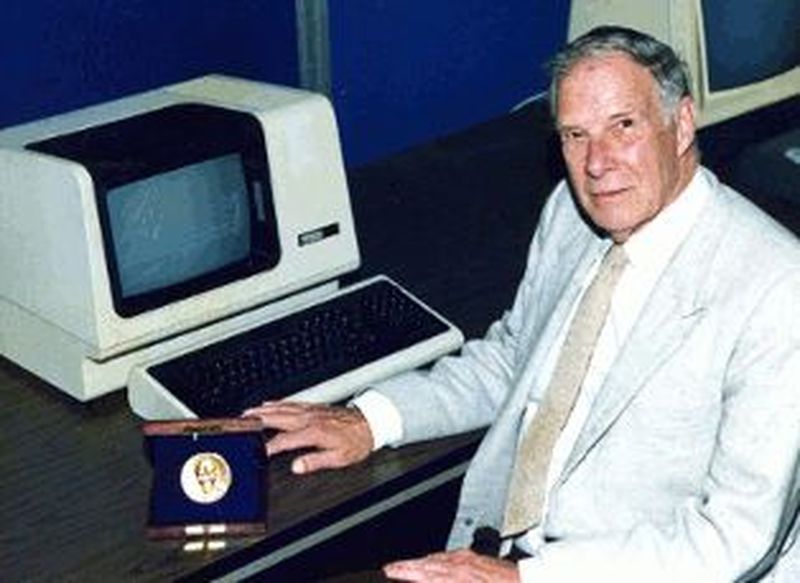
Hata düzeltme uygulamaları sadece üstte anlatılan uygulama ile sınırlı değildir, daha çok alanda ayrıca uygulamaları vardır. Hata düzeltme uygulamasının çeşitli alanlarda farklı yaklaşımlara sahip olsa da genel amaçları birbirlerine paraleldir. Hata düzeltmeye “Reed-Solomon” hata düzeltme uygulamasının haricinde de örnek gösterilebilir. Mesela cep telefonlarındaki radyo bağlantısının zayıflamasını ve mevcut olan gürültülerini onarmak amacıyla ayrıca kodlama teknikleri kullanılır.

* 1. **Kodlama Teorisinin Tarihi**

**I-)** 1948'de yılında matematikçi ve kriptografici olan Claude Shannon, Bell System Technical Journal adlı dergide Temmuz ve Ekim sayılarında iki bölümden oluşan makalesini yayınladı ("A Mathematical Theory of Communication"). Bu çalışma özünde vericideki iletilmek istenen bilginin en iyi şekilde nasıl kodlanabilir sorununa odaklanır. Ayrıca bu çalışmada Norbert Wiener tarafından geliştirilmiş olup, o zamanlarda iletişim teorisine daha kullanımının başlarında olan olasılık teorisindeki malzemeleri kullandı. Shannon aslında bilgi teorisi alanını ortaya çıkarttı ve böylece bir mesajdaki belirsizliğin bir ölçüsü olan bilgi entropisini geliştirdi. Aşağıda Cluade Shannon’un fotoğrafı verilmiştir.

***Şekil 3***

**II-)** 1949 yılında geliştirilen ikili golay kodu, her 24 bitlik kelimedeki üç hatayı düzeltir ve dördüncü hatayı da tespit eder.

**III-)** Richard Hamming, Bell Lab’larındaki hata tespit ve hata düzeltici kodlar üzerine olan çalışmaları sayesinde 1968 yılında Turing Ödülü’ne layık görülmüştür. Richard Hamming çalışmaları sonucunda, Hamming kodları, Hamming pencereleri, Hamming sayıları ve Hamming mesafesi olarak bilinen terimleri icat etmiştir. Aşağıdaki görselde Richard Hamming’e ait bir fotoğraf verilmiştir.

***Şekil 4***

**IV-)** 1972'de Nasir Ahmed’in katkılarıyla, 1973'de ise T. Natarajan ve K. R. Rao ile geliştirilen ayrık kosinüs dönüşümü önerildi. JPEG, MPEG ve MP3 gibi çoklu medya formatlarının temeli olan çok yaygın olarak kullanılan kayıplı sıkıştırma algoritmasıdır.

* 1. **Kodlama Teorisinin Uygulamaları**

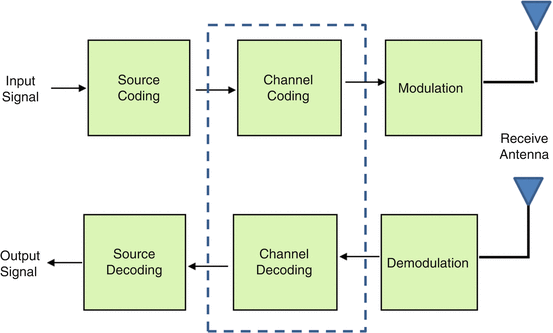
Kodlama teorisinin yukarıda anlatılandan başka kullanım alanları ve amaçları vardır. Bu teorideki bir başka kullanım alanına örnek vermek gerekilirse, senkronizasyona yardımcı olan kodlar tasarlanması söylenebilir. Tasarlanacak kod faz kaymasını kolayca tespit edecek ve düzeltebilecek şekilde tasarlanıp aynı zamanda aynı kanal üzerinde birden fazla sayıdaki sinyalin gönderilebileceği şekilde tasarlanabilir. Bir başka kod uygulaması da piyasadaki bazı cep telefonlarında bulunun kod bölmeli çoklu erişimdir (CDMA). Bu uygulamayı kısaca açıklanmak gerekirse; her telefona, diğer telefonların kodları yardımıyla yakın olarak ilgisiz olan bir kod dizisi atanır. Bu kodlar iletim esnasında sesli mesajı temsil eder ve veri bitlerini modüle etmek amacıyla kullanılır. Alıcı kısmında ise verileri kurtarmak amacıyla demodülasyon işlemi gerçekleştirilir. Bu kod dizisinin sağladığı imkânlar arasında birçok kullanıcının aynı radyo kanalı aynı zamanda kullanmasına izin vermesi vardır. Alıcı kısmındaki demodülatöre diğer kullanıcıların sinyalleri çok düşük miktarda bir gürültü olarak yansıyacaktır.

1. **Kanal Kodlama**

Kanal kodlama teorisinin amacı bir hatayı veya birden çok hatayı düzeltebilen kodları bulmak veya en azından hatayı tespit eden kodları bulmaktır. Bu uygulama için birbirleriyle bağlantıları gözetmeksizin, bu alanlardaki performanslar aslında eksileriyle ve artılarıyla bir çeşit değiş-tokuştur. Bu sebeple her kodun her alandaki uygulama verimi eşit olmaz, bu kodların yüksek verimle çalıştığı alanlar birbirinden farklıdır. Bu kodda belirtilen özellikler temelde iletim sırasında gerçekleşen hataların olasılığına bağlıdır. Herhangi bir CD'de meydana gelen bozulmanın kaynağı toz veya çiziklerdir.

Kanal kodlama uygulaması çok iyi bir kod uygulaması olmasa da, tekrarlayan kod kolay bir şekilde anlaşılabilir bir örnek olarak gösterilebilir. Mesela vericiden çıkan sesi temsil eden bir veri bitini alıp üç kez gönderelim. Bu işlemi alıcı kısmında üç tekrarı inceleyeceğiz ve çoğunluk oyu alacağız. Buradaki işlem bitleri yalnızca sırayla göndermeyip onları araya koymamızdır. Bu işlem sırasında ilk olarak veri bitleri bloğu 4 küçük bloğa bölünür ve sonra bloklar boyunca dolaşıyoruz. Birinci parçadan bir bit, ardından ikinciden (bu sırayla) gönderiyoruz. Bu işlemi 3 kere tekrar ederiz ve bu işlemi yapmamızdaki amaç verileri diskin yüzeyine yaymaktır.

Üstte bahsettiğimiz üzere her kodun efektif olarak kullanıldığı alan farklıdır ve uygulanan her kodun ideal olduğu bir alan vardır. Derin uzay iletişimi alıcının sürekli bir yapıya sahip olan termal gürültüsü ile sınırlıdır ve buna benzer şekilde, dar bant modemler, telefon şebekesinde bulunan gürültü ile sınırlıdır.

Sonuç olarak kanal kodlama uygulaması, giriş dizilerini iletime fazlalık ekleyen kod dizilerine eşleyerek dijital iletişim sistemlerinde gelişmiş hata performansı sağlar. Kanal kapasitesinin (yani güvenilir iletişimin mümkün olduğu maksimum hız), iletilen veri bloğunun uzunluğu arttıkça rastgele kodlarla ulaşılabilir olduğu iyi bilinmektedir. Ancak blok uzunluğu arttıkça kod çözücü karmaşıklığı yönetilemez hale gelir. Bu nedenle, mütevazı kodlama ve kod çözme karmaşıklığı sunarken, kapasiteye yakın performans gösteren kodlar tasarlamak hedeflenmiştir. Aşağıda kanal kodlamaya ait bir diyagram verilmiştir.

***Şekil 5***

**KAYNAKÇA**

<https://tr.wikipedia.org/wiki/Kodlama_teorisi>

<https://en.wikipedia.org/wiki/Coding_theory>

<https://www.accelercomm.com/overview-channel-coding>

<https://www.sciencedirect.com/topics/computer-science/channel-coding>

<https://e-bergi.com/y/richard-hamming/>

<https://tr.wikipedia.org/wiki/Claude_Elwood_Shannon>

<https://link.springer.com/chapter/10.1007/978-3-319-21170-1_1>