

**T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**HABERLEŞME SİSTEMLERİNDE KULLANILAN
TEMEL KODLAMA VE SIKIŞTIRMA TEKNİKLERİ**

Musa ÇIBUK

DOKTORA SEMİNERİ

**ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ
ANABİLİM DALI**

ELAZIĞ

2004

T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HABERLEŞME SİSTEMLERİNDE KULLANILAN
TEMEL KODLAMA VE SIKIŞTIRMA TEKNİKLERİ

Musa ÇIBUK

DOKTORA SEMİNERİ
ELEKTRİK-ELEKTRONİK MÜHENDİSİLİĞİ
ANABİLİM DALI

Bu seminer çalışması/...../..... tarihinde, aşağıda belirtilen jüri tarafından Oybirliği/Oyçokluğu ile Başarılı/Başarısız olarak değerlendirilmiştir.

(İmza)

Danışman

Yrd.Doç.Dr. Hasan H. BALIK

(İmza)

Üye

(İmza)

Üye

Bu seminer çalışmasının kabulü, Fen Bilimleri Enstitüsü Yönetim Kurulu'nun/...../..... tarih ve sayılı kararı ile onaylanmıştır.

TEŐEKKÜR

Bu seminer alıřmamda öncelikle, alıřmam süresince bana her türlü olanađı sađlayan ve elinden gelen yardımı yapan, fikirleriyle bana yol gösteren, ok deđer verdiđim, danıřmanım Sayın Yrd.Do.Dr. Hasan Hüseyin BALIK'a sonsuz teőekkür ederim.

alıřmam süresince stres ve sıkıntılara katlanan sevgili eřim Yonca'ya ve bir anlık sesinin bile bana büyük mutluluk verdiđi biricik kızım İkra'ya ayrıca teőekkür ederim.

Yine bu alıřmam esnasında emeđi geen ve bana yardımcı olan diđer tüm arkadaşlarıma da teőekkürü bir bor bilirim.

İÇİNDEKİLER

İÇİNDEKİLER	I
ŞEKİLLER LİSTESİ	V
TABLolar LİSTESİ.....	VIII
SİMGELER LİSTESİ	IX
KISALTMALAR LİSTESİ.....	XI
ÖZET	XIII
1 GİRİŞ	1
2 HABERLEŞME	3
2.1 Giriş.....	3
2.2 Haberleşme Türleri.....	4
2.2.1 Ortam Türüne Göre Haberleşme Türleri	4
2.2.1.1 Kablolü Haberleşme	5
2.2.1.2 Kablosuz (Wireless) Haberleşme	6
2.2.2 Sinyal Türüne Göre Haberleşme Türleri	6
2.2.2.1 Analog Haberleşme	6
2.2.2.2 Sayısal (Dijital) Haberleşme.....	7
2.3 Haberleşmedeki Kısıtlamalar	7
2.3.1 Gürültü ve Girişim	8
2.3.2 Bant Geniřlięi.....	8
2.3.3 Sinyal Gücü.....	10
2.4 Çoęullama (Multiplexing).....	10
2.4.1 Frekans Bölmeli Çoęullama (FDM).....	11
2.4.2 Zaman Bölmeli Çoęullama (TDM).....	11
3 MODÜLASYON.....	13
3.1 Giriş.....	13
3.2 Sürekli Dalga (Analog) Modülasyonu	14
3.2.1 Genlik Modülasyonu (Amplitude Modulation – AM)	15
3.2.2 Açık Modülasyonu	16
3.2.2.1 Frekans Modülasyonu (Frequency Modulation – FM).....	16
3.2.2.2 Faz Modülasyonu (Phase Modulation – PM)	17
3.2.3 Genlik Kaydırmalı Anahtarlama (Amplitude Shift Keying – ASK).....	17
3.2.4 Frekans Kaydırmalı Anahtarlama (Frequency Shift Keying – FSK)	18

3.2.5	Faz Kaydırmalı Anahtarlama (Phase Shift Keying – PSK).....	18
3.3	Darbe Modülasyonu.....	19
3.3.1	Darbe Modülasyonun Üstünlükleri	20
3.3.2	Örnekleme Teoremi.....	22
3.3.3	Darbe Genlik Modülasyonu (Pulse Amplitude Modulation – PAM).....	22
3.3.4	Darbe Süresi Modülasyonu (Pulse Duration Modulation – PDM).....	23
3.3.5	Darbe Konumu Modülasyonu (Pulse Position Modulation – PPM)	25
3.3.6	Darbe Kod Modülasyonu (Pulse Code Modulation – PCM).....	26
3.3.6.1	Kuantalama.....	26
3.3.6.2	Düzgün Kuantalama	28
3.3.6.3	Kuantalama Hataları	29
3.3.6.4	Boş Kanal Gürültüsü	30
3.3.6.5	Düzgün Olmayan Kuantalama.....	30
3.3.7	Delta Modülasyonu (Delta Modulation – DM).....	33
3.3.7.1	Doğrusal Delta Modülasyonu	33
3.3.7.2	Eğim Sınırlaması	35
3.3.7.3	PCM ile DM'nun Karşılaştırılması.....	36
3.3.8	Adaptif Delta Modülasyonu (Adaptive Delta Modulation – ADM)	36
3.3.8.1	Sabit Basamak Seviyeli Adaptif Delta Modülasyonu.....	36
3.3.8.2	Bir (1) Bit Bellekli Adaptif Delta Modülasyonu	38
3.3.9	Diferansiyel PCM (Differential Pulse Code Modulation – DPCM).....	39
4	SAYISAL KODLAMA.....	40
4.1	Giriş.....	40
4.2	Sıfıra Dönüşsüz (Non-Return to Zero – NRZ) Kodlama	41
4.3	Sıfıra Dönüştü (Return to Zero – RZ) Kodlama	41
4.4	İki kutuplu (Bipolar – HDB3) Kodlama	42
4.5	İki Fazlı (Biphase – Manchester) Kodlama.....	42
4.6	Diferansiyel Manchester Kodlaması	43
4.7	4B/5B Kodlama.....	44
4.8	Çok Seviyeli Eşik – 3 (Multi-Level Threshold – MLT-3) Kodlama	44
5	HATA KAYNAKLARI ve HATA DENETİMİ	45
5.1	Giriş.....	45
5.2	Hata Kaynakları	45
5.2.1	İşaret Zayıflaması (Attenuation)	46
5.2.2	Gecikmeye Bağlı Bozulma (Delay)	47

5.2.3	Gürültü (Noise)	48
5.2.3.1	Termal Gürültü	49
5.2.3.2	Modülasyonlararası Gürültü	50
5.2.3.3	Uyarım Gürültüsü	50
5.2.4	Çapraz Konuşma (Crosstalk).....	50
5.3	Hata Kontrolü.....	50
5.3.1	Hata Bulma Yöntemleri.....	51
5.3.2	Bit Hata Oranı (Bit Error Rate – BER) ve Hata Patlaması.....	52
5.3.3	Eşlik Biti Denetimi (Parity Check)	53
5.3.4	İki-Boyutlu Eşlik Biti Denetimi (2-Dimensional Parity)	54
5.3.5	Dönüşümlü Artıklık Denetimi (Cyclic Redundancy Check – CRC).....	54
5.3.6	Sağlama-Toplamı (Checksum) Algoritması.....	56
5.3.7	Hata Düzeltme Yöntemleri.....	56
5.3.7.1	Hamming Uzaklığı ve Hamming Kodlaması	57
6	VERİ SIKIŞTIRMA.....	59
6.1	Giriş.....	59
6.2	Entropi Kodlaması (Kayıpsız Sıkıştırma)	61
6.2.1	Tekrarlama Uzunluğu Kodlaması (Run-Lenght Encoding – RLE).....	62
6.2.2	İstatistiksel Kodlama	63
6.2.2.1	Huffman Kodlaması	64
6.3	Kaynak Kodlaması (Kayıplı Sıkıştırma)	68
6.3.1	JPEG - Resim Sıkıştırması	69
6.3.2	MPEG- Video Sıkıştırması.....	70
7	UYGULAMA ve MYWAVEGEN Programı	72
7.1	Giriş.....	72
7.2	Program Ana Kontrol Paneli.....	73
7.3	Osiloskop Paneli	74
7.4	Modülasyon ve Parametre Paneli.....	75
7.4.1	Modülasyon Sekmesi	75
7.4.2	Parametre Sekmesi	75
7.4.3	Seçenekler Sekmesi.....	76
7.4.4	Hakkında Sekmesi.....	76
7.4.5	Yardım Sekmesi	77
7.5	Bilgi Paneli.....	77
7.6	Durum Satırı.....	77

8	SONUÇ	78
	KAYNAKLAR	80
	ÖZGEÇMİŞ	81

ŞEKİLLER LİSTESİ

Şekil 2.1 : Bir iletişim sisteminin temel fonksiyonel bileşenleri.....	3
Şekil 2.2 : Elektromanyetik Tayf	3
Şekil 2.3 : Kablolü ortam türleri	5
Şekil 2.4 : Elektromanyetik dalgalar ve kablosuz iletişim	6
Şekil 2.5 : Analog Sinyal	6
Şekil 2.6 : Sayısal (Dijital) Sinyal.....	7
Şekil 2.7 : RC devresinin frekansa bağılı tepkisi	9
Şekil 2.8 : Frekans Bölmeli Çoklama (FDM)	11
Şekil 2.9 : Zaman Bölmeli Çoklama (TDM)	11
Şekil 3.1 : Genlik Modülasyonu (AM)	15
Şekil 3.2 : Frekans Modülasyonu (FM)	16
Şekil 3.3 : Faz Modülasyonu (PM)	17
Şekil 3.4 : Genlik Kaydırmalı Anahtarlama (Amplitude Shift Keying – ASK).....	17
Şekil 3.5 : Frekans Kaydırmalı Anahtarlama (Frequency Shift Keying – FSK).....	18
Şekil 3.6 : Faz Kaydırmalı Anahtarlama (Phase Shift Keying – PSK).....	18
Şekil 3.7 : Darbe sinyali örneği.....	19
Şekil 3.8 : Bilgi sinyalinin darbe dizisiyle çarpımı	20
Şekil 3.9 : Darbe modülasyonunda çoklama özelliği.....	21
Şekil 3.10 : Darbe Genlik Modülasyonu (Pulse Amplitude Modulation – PAM)	23
Şekil 3.11 : Darbe Süresi Modülasyonu (Pulse Duration Modulation – PDM).....	24
Şekil 3.12 : Darbe Konumu Modülasyonu (Pulse Position Modulation – PPM).....	25
Şekil 3.13 : Darbe Kod Modülasyonu (Pulse Code Modulation – PCM)	26
Şekil 3.14 : Örnek kuantalama seviyeleri	27
Şekil 3.15 : Düzgün kuantalama eğrisi	28
Şekil 3.16 : Kuantalama hatasının olasılık dağılımı.....	29
Şekil 3.17 : Boş kanal gürültüsünü önleyici düzgün kuantalama eğrisi.....	30
Şekil 3.18 : A/D ve D/A çeviriciler için Sıkıştırma ve Genleştirme eğrileri.....	31
Şekil 3.19 : Haberleşme sisteminde işaret seviyesinin değiştirilmesi	31
Şekil 3.20 : Çok kanallı sistemde işaret seviyesi değişimi.....	32
Şekil 3.21 : A ve μ tipi eğriler.....	32
Şekil 3.22 : Örnekler arasındaki fark değeri	33
Şekil 3.23 : Doğrusal Delta Modülasyonu; a) verici, b) alıcı.....	34

Şekil 3.24 : Delta modülasyonunun çalışması	34
Şekil 3.25 : Delta modülasyonunda boşa çalışma	35
Şekil 3.26 : Örnek 6 seviyeli Adaptif Delta Modülasyonu	37
Şekil 3.27 : Q -seviyeli Adaptif Delta Modülatörü	37
Şekil 3.28 : Adaptif Delta Modülasyonlu Sistem.....	38
Şekil 3.29 : Adaptif Delta Modülasyonunda gürültü	38
Şekil 3.30 : Diferansiyel PCM (DPCM) sistem; a) verici, b) alıcı.....	39
Şekil 4.1 : Sıfıra Dönüşsüz (NRZ) Kodlama.....	41
Şekil 4.2 : Sıfıra Dönüştü (RZ) Kodlama.....	42
Şekil 4.3 : İki Kutuplu (Bipolar) Kodlama.....	42
Şekil 4.4 : Manchester Kodlama	43
Şekil 4.5 : Diferansiyel Manchester Kodlama	44
Şekil 4.6 : MLT-3 Kodlama.....	44
Şekil 5.1 : İşaret zayıflamasından kaynaklanan hata.....	46
Şekil 5.2 : Yineleyici kullanılarak işaret kuvvetinin yükseltilmesi.....	47
Şekil 5.3 : Gecikmeye bağlı bozulma.	48
Şekil 5.4 : İletişim ortamındaki gürültü.	48
Şekil 5.5 : Hata kaynaklarının toplam etkisi.	49
Şekil 5.6 : Hata Patlaması Örneği	52
Şekil 5.7 : Eşlik Biti Denetim (XOR'lama) işlemi.....	53
Şekil 5.8 : FCS oluşturma ve kontrolü.....	55
Şekil 6.1 : Huffman kodlaması azaltma algoritması.	65
Şekil 6.2 : Bölünme algoritması.....	65
Şekil 6.3 : Örnek Huffman kod ağacı.....	68
Şekil 6.4 : JPEG sıkıştırmasının aşamaları.	69
Şekil 6.5 : DTC algoritması ile matrise dönüştürme.	70
Şekil 6.6 : Nicelikleri bulma tablosu.....	70
Şekil 6.7 : MPEG başvuru sistemi.	70
Şekil 6.8 : MPEG sıkıştırmada I, P ve B çerçeveleri.	71
Şekil 7.1 : myWaveGen Programı ana ekranı.....	72
Şekil 7.2 : Program ana kontrol paneli.....	73

Şekil 7.3 : Osiloskop Paneli	74
Şekil 7.4 : Osiloskop Kontrol Paneli.....	74
Şekil 7.5 : Modülasyon Sekmesi.....	75
Şekil 7.6 : Parametre Sekmesi.....	75
Şekil 7.7 : Seçenekler Sekmesi	76
Şekil 7.8 : Hakkında Sekmesi	76
Şekil 7.9 : Bilgi Paneli	77
Şekil 7.10 : Durum Satırı	77

TABLolar LİSTESİ

Tablo 3.1 : Bilgi ve Taşıyıcı sinyale göre modülasyon türleri	14
Tablo 3.2 : Örnek kuantalama değerleri.....	27
Tablo 3.3 : 8 seviyeli kuantalama için kaynak seviyeleri ve ikilik kod kelimeleri	28
Tablo 4.1 : 4B/5B Kodlama tablosu.....	44
Tablo 6.1 : 8 ayrı karakterin istatistiksel dağılımları	63
Tablo 6.2 : Huffman kodlamasının sonucunda oluşan kod değerleri.....	66
Tablo 6.3 : Örnek sekiz karakterin birbirlerine bağıl sıklıkları.....	67
Tablo 6.4 : Huffman ağaç kodlaması yöntemi	67

SİMGELER LİSTESİ

λ	: Dalga Boyu (m)
c	: Işık Hızı (300.000 km/sn)
f	: Frekans (Hz)
L_{min}	: Minimum Anten Uzunluğu (m)
$e_c(t)$: Zamana bağlı taşıyıcı sinyal genlik fonksiyonu (V)
E_c	: Taşıyıcı sinyal genliği (V)
ω_c	: Taşıyıcı sinyal açısal hızı (rad/sn)
t	: Zaman (sn)
ϕ	: Açı (rad)
$e_m(t)$: Zamana bağlı bilgi sinyali genlik fonksiyonu (V)
E_m	: Bilgi sinyali genliği (V)
ω_m	: Bilgi sinyali açısal hızı (rad/sn)
$e_{AM}(t)$: Zamana bağlı genlik modülasyonu fonksiyonu (V)
m	: Genlik modülasyon indexi (E_m/E_c)
$e_{FM}(t)$: Zamana bağlı frekans modülasyonu fonksiyonu (V)
$e_{ASK}(t)$: Zamana bağlı ASK modülasyonu fonksiyonu (V)
f_H	: Yüksek frekans
f_L	: Alçak frekans
$e_{PSK}(t)$: Zamana bağlı PSK modülasyonu fonksiyonu (V)
$f(t)$: Sürekli-zamanlı işaret
$p(t)$: Darbe katarı dizisi
$f_p(t)$: $f(t)$ ve $f_p(t)$ fonksiyonlarının çarpımı sonucu elde edilen darbe dizisi
W	: Açısal Hız (rad/sn)
T	: Peryot (sn)
f_s	: Örnekleme frekansı (Hz)
$e_{PAM}(t)$: Zamana bağlı PAM modülasyonu fonksiyonu (V)
A	: Genlik (V)
A_{max}	: Maksimum Genlik (V)
A_{min}	: Minimum Genlik (V)
Q	: Kuantalama Seviyesi
a	: Kuantalama Aralığı (V)
S/N	: Sinyal-Gürültü Oranı (dB)
$x(t)$: Zaman bağlı genlik fonksiyonu (V)

$\hat{x}(t)$: Zaman bağılı darbe modülasyonu genlik fonksiyonu (V)
Δ	: PCM'deki kuantalam seviyesi
Δ_n	: PCM'deki n. kuantum seviyesi
Δ_{n-1}	: PCM'deki n-1. kuantum seviyesi
K	: Adaptif delta modülasyonunda sabit
P_1, P_2	: Güç (watt)
\oplus	: XOR işlemi
n	: Dijit Sayısı
H	: Entropi Miktarı (bit/karakter)
H_{max}	: Maksimum Entropi Miktarı (bit/karakter)
L	: Toplam Kod Uzunluğu

KISALTMALAR LİSTESİ

AC	: Alternatif Akım (Alternate Current)
DC	: Doğru Akım (Direct Current)
RF	: Radyo Frekansı (Radio Frequency)
FDM	: Frekans Bölmeli Çoklama (Frequency Division Multiplexing)
TDM	: Zaman Bölmeli Çoklama (Time Division Multiplexing)
AM	: Genlik Modülasyonu (Amplitude Modulation)
FM	: Frekans Modülasyonu (Frequency Modulation)
PM	: Faz Modülasyonu (Phase Modulation)
FAM	: Tam genlik Modülasyonu (Full Amplitude Modulation)
DSBSC-AM	: Çift Yan Bantlı Bastırılmış Taşıyıcı Genlik Modülasyonu
SSB-AM	: Tek Yan Bantlı Genlik Modülasyonu (Single Side Band Amplitude Modulation)
ASK	: Genlik Kaydırmalı Anahtarlama (Amplitude Shift Keying)
FSK	: Frekans Kaydırmalı Anahtarlama (Frequency Shift Keying)
PSK	: Faz Kaydırmalı Anahtarlama (Phase Shift Keying)
PAM	: Darbe Genlik Modülasyonu (Pulse Amplitude Modulation)
PDM	: Darbe Süresi Modülasyonu (Pulse Duration Modulation)
PWM	: Darbe Genişlik Modülasyonu (Pulse Width Modulation – PWM)
PPM	: Darbe Konumu Modülasyonu (Pulse Position Modulation)
PCM	: Darbe Kod Modülasyonu (Pulse Code Modulation)
CCITT	: Uluslar arası Telefon ve Telgraf Danışma Kurulu Komitesi (International Consultative Committee for Telephony and Telegraph)
DM	: Delta Modülasyonu (Delta Modulation)
DPCM	: Diferansiyel Darbe Kod modülasyonu (Differential Pulse Code Modulation)
AGF	: Alçak Geçiren Filtre
ADM	: Adaptif Delta Modülasyonu (Adaptive Delta Modulation)
NRZ	: Sıfıra Dönüşsüz Kodlama (Non Return to Zero)
RZ	: Sıfıra Dönüştü Kodlama (Return Zero)
MLT – 3	: Çok Seviyeli Eşik – 3 Kodlama (Multi-Level Threshold – 3)
BPS	: Saniyedeki Bit Sayısı (Bit Per Second)
NEXT	: Near End Crasstalk
XOR	: Özel YADA işlemi (Exclusive-OR)
BER	: Bit Hata Oranı (Bit Error Rate)
CRC	: Dönüşümlü Artıklık Denetimi (Cyclic Redundancy Check)

FCS	: Çerçeve Kontrol Dizisi (Frame Check Sequence)
ARQ	: Otomatik Tekrar İstemi (Automatic ReQuest)
RLE	: Tekrarlama Uzunluğu Kodlaması (Run-Lenght Encoding)
VLE	: Değişken Uzunlukta Kodlama (Variable Length Coding)
JPEG	: Fotoğraf Uzmanları Birliği Grubu (Joint Photographic Experts Group)
MPEG	: Hareketli Resim Uzmanları Birliği Grubu (Motion Picture Experts Group)
DCT	: Ayrık Cosinüs Dönüşümü (Discrete Cosinus Transform)
CD	: Kompakt Disk (Compact Disc)
ASCII	: American Standard Code for Information Interchange
PC	: Kişisel Bilgisayar (Personel Computer)
MB	: Mega Byte (Veri saklama birimi)
RAM	: Rasgele Erişimli Bellek (Random Access Memory)
OSI	: Açık Sistemler Arası Bağlantı Modeli (Open System Interconnection)
FDDI	: Fiber Dağıtımli Veri Arabirimi (Fiber Distributed Data Interface)

ÖZET

DOKTORA SEMİNERİ

HABERLEŞME SİSTEMLERİNDE KULLANILAN TEMEL KODLAMA VE SIKIŞTIRMA TEKNİKLERİ

Musa ÇIBUK

Fırat Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik – Elektronik Mühendisliği Anabilim Dalı

2004, Sayfa: 81

Bu seminer çalışmasında, bir haberleşme sisteminin temel unsurları ve bunların gerçekleştirimlerdeki teknikler ele alınmıştır. Çalışmada ilk olarak, haberleşme konusu genel hatlarıyla işlenmiştir. İkinci kısımda, modülasyon konusu ve modülasyon teknikleri ele alınmış ve detaylı olarak anlatılmıştır. Üçüncü kısımda da, sayısal kodlama, hata kontrolü ve veri sıkıştırma konuları genel bileşenleri ile ele alınmıştır.

Son olarak da, bu konuları kapsayacak şekilde hazırlanan myWaveGen uygulaması anlatılmıştır. Bu uygulama yazılımında çalışmada söz edilen temel modülasyon teknikleri ve veri kodlama uygulamaları detaylı bir şekilde yapılabilmektedir. Yazılım Delphi görsel uygulama geliştirme programı ile hazırlanmıştır.

Anahtar Kelimeler: Haberleşme, Haberleşme Türleri, Modülasyon Teknikleri, Kodlama Teknikleri, Hata Kontrolü, Veri Sıkıştırma

1 GİRİŞ

İletişim alanının, bir teknoloji toplumunun gelişmişlik düzeyini yansıttığı söylenir. Eğer bu doğru ise, birkaç yılda bir, bir teknoloji devrimine tanık oluyoruz demektir. Yaşadığımız şu dönem içerisinde televizyon, iletişim, uzay, sayısal mikroişlemciler, veri iletişimi kablosuz haberleşme birçok gibi teknolojik devrimler gördük. Bu eğilim her geçen gün devam etmekte ve daha da hızlanmaktadır. Fiber optik ve kablosuz iletişim kanalları üzerinden haberleşmek artık sıradan bir olgu haline gelmiştir. Dikkati çeken bir husus da, belirtilen bu ilgi alanlarının, kolayca anlaşılabilen, temel iletişim ilkeleriyle biri birine bağlı olmasıdır.

İnsanlar, çok eski çağlardan beri bilgiyi, sürekli insanoğlunun normal görme ve duyma sınırlarının ötesine taşıma yollarını, yani bilgiyi iletmenin yollarını araştırmışlardır. Kızılderililerin dumanla, gemicilerin bayrakla işaretlemeleri hepimizce bilinir. Uygarlığın gelişmesi de insanoğlunun doğal duyuları olan görme ve duymayı; kabul edilmiş bir dil yada kodu, yazıyı kullanarak bilgi değiştirme yeteneğine bağlıdır.

İletişim terimi bilgiyi elektriksel yollarla göndermeye, almaya, işlemeye karşılık gelir. İletişimin amacı, herhangi bir biçimdeki bilginin zaman ve uzay içinde kaynak olarak adlandırılan bir noktadan, kullanıcı denilen başka bir noktaya aktarılmasıdır. Bugün telefon, radyo, televizyon gibi elektriksel iletişimin çeşitleri, günlük yaşantımızın vazgeçilmez birer parçası olmuşlardır. Elektriksel iletişimin diğer bazı önemli örnekleri şu şekilde sıralanabilir; radar, teletre dizgeleri, bilgisayarlar arası bilgi aktarımı, askeri amaçlar için kullanılan telsiz. Bu liste istenildiği kadar genişletilebilir. Elektronik devre öğeleri teknolojisindeki yeni ilerlemelere bağlı olarak önümüzdeki yıllarda iletişim dizgelerinde de önemli gelişmelerin olması kaçınılmazdır.

Elektriksel anlamda iletişim, ilk olarak 1840'larda telgraf ile başlamıştır. Birkaç "10 yıl" sonra telefonla ve bu yüzyılın başında da radyo ile daha da gelişmiştir. Elektronik tüpün bulunuşu ile doğan radyo iletişimi, büyük ölçüde 2. Dünya savaşı sırasındaki çalışmalardan kaynaklanmıştır. Transistör, entegre devre ve diğer yarı iletken araçların bulunup, kullanılmasıyla radyo ve TV geliştirilmiş ve yaygın bir şekilde kullanılmaya başlanılmıştır. Özellikle son zamanlarda uydu ve fiber optik teknolojisi, bilgisayarlara ve diğer veri iletişimlerine artan bir önem yüklemiş ve iletişim daha yaygın bir hale gelmiştir.

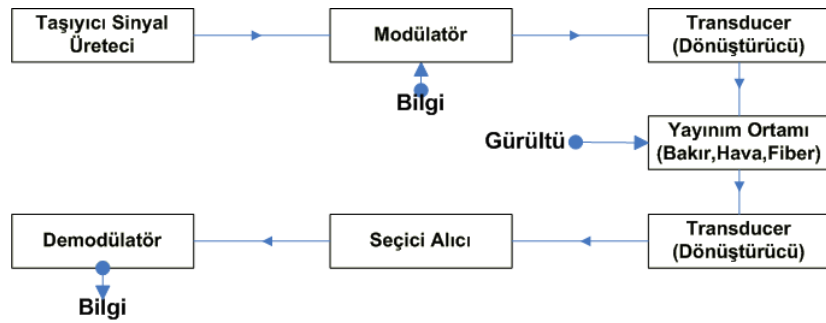
Modern bir iletiřim sistemi, bilgi gndermeden nce onun sıraya koyulmasıyla, iřlenmesiyle ve korunmasıyla ilgilenir. Gerek anlamda gnderme, daha fazla iřleme ve grltnn szlmesiyle gerekleřir. Son olarak, kod zme, mesajı koruma ve bilgi algılama basamaklarından oluřan alma iřlemi gelir.

İletici-Kanal-Alıcı, Claude E. Shannon, btn iletiřim sistemleri iin temel model olan bu triyoyu bir araya getirdiğinde, 1947 senesiydi. Peki, iletilecek veriyi kayıpsız tařması iin kanalın geniřliđi minimum ne olmalı? yada bir sinyalin ierdiđi gerek enformasyon ne kadardır? Bay Shannon, bir matematikiydi. Boolean cebiri kullanarak elektronik devrelerin modellenebileceđini gsterdiđi, 1937 tarihli master tezi yzyılın en nemli tezi sayılmaktadır. Bu matematiki sinyallerin modellenmesi iin istatistiksel zelliklerini kullanarak “Information Theory”nin temelini attı. Bununla da kalmayıp, kanalın geniřliđi sabit olduđu takdirde verinin en ok ne kadarının geebileceđi sorusundan “Rate-Distortion Theory”yi ortaya ıkardı. Dođada analog formda bulunan tm sinyaller iletici tarafından sayısal endeksler ile ifade edilebilen sembollere dnřtrlecekti. Alıcı kanaldan kayıpsız, yada kayıplara karřı korumalı geecek bu endeksleri kendisinin de bildiđi sembollere atayıp sinyali yeniden oluřturacaktı. Kanalın yeterince geniř olmaması halinde ise iletici sinyalin bazı paralarını atmak zorunda kalacaktı. Atılması en az hasara sebep olacak paraların tespiti, “veri sıkıřtırma teknolojisi”nin dođmasına yol atı. [20]

2 HABERLEŞME

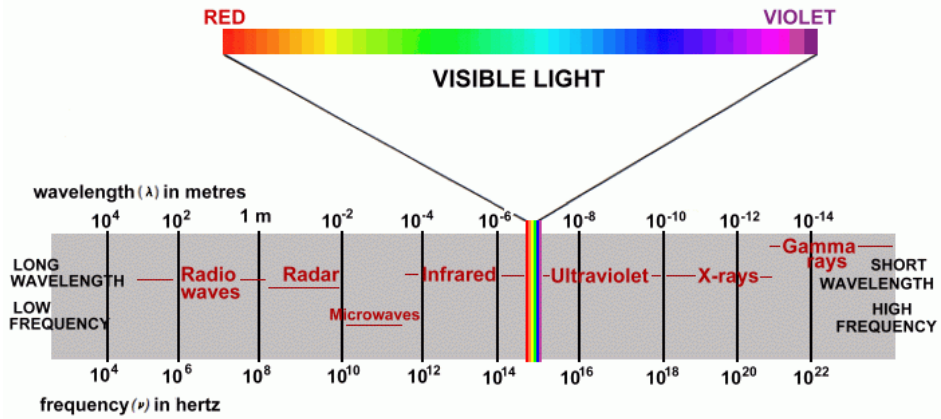
2.1 Giriş

Haberleşme kısaca; bilginin bir yerden başka bir yere hatasız olarak iletilmesi olayına denir. Haberleşmede temel prensip, iletilecek olan bilginin önce elektrik, elektromanyetik veya optik enerjiye dönüştürülerek uzak mesafelere iletilmesi ve karşı tarafta tekrar eski orijinal haline dönüştürülmesidir. Sistemin randımanı, iletim sırasında kaçınılmaz olarak meydana gelen bilgi kaybının miktarı ile ölçülür.



Şekil 2.1 : Bir iletişim sisteminin temel fonksiyonel bileşenleri

Bütün taşıyıcı haberleşme sistemlerinde, Şekil 2.1'de görülen temel fonksiyonel bileşenler bulunur. İletilecek bilgi, modülörde, sinyal üretici tarafından üretilen bir taşıyıcı sinyal üzerine bindirilir. Transdüser (gönderme anteni), modüle edilmiş sinyali, bu sinyalin elektromanyetik dalga şeklinde yayıldığı yayılma ortamına bağlar. Bu sinyalin az bir kısmı alıcıya ulaşır. Burada başka bir transdüser, (alıcı anten), alternatif sinyalle birlikte gürültüyü seçici alıcıya bağlar. Seçici alıcı da, modüle edilmiş taşıyıcı sinyalini filtreden geçirir. Demodülörde ise, bilgiyi modüle edici sinyalin, taşıyıcıdan ayrılması sağlanarak, bilgi yeniden elde edilir.



Şekil 2.2 : Elektromanyetik Tayf

Bir haberleşme sisteminin tipik özellikleri, büyük oranda, elektromanyetik tayfin kullanılan parçasına bağlıdır (Şekil 2.2). Radyo sistemlerinde elektromanyetik taşıyıcı dalga, tayfin radyo frekansı (RF) bölümünden seçilir. Mikrodalga veya milimetre sistemleri, tayfin bu bölümünde bulunan taşıyıcı frekanslarına sahiptir. Optik haberleşme sistemleri ise kızılaltı görünür ışık ve mor ötesi frekanslardan taşıyıcı frekanslarına sahiptir.

Haberleşme sisteminde iletilen bilgi miktarı, modüle edilmiş taşıyıcının bant genişliği (frekans tayfı) ile doğrudan ilgili olup, bu da genelde taşıyıcı frekansının belli bir yüzdesi kadardır. Bu nedenle taşıyıcı frekansının artması, iletim bant genişliğinin ve dolayısıyla bilgi ileten sistemin tüm kapasitesinin artmasını sağlar. Sonuçta, optik haberleşme sistemlerinin özel bir ilgi alanı olarak ortaya çıkmasının nedeni de budur.

En temel tanımıyla bilginin bir noktadan başka bir noktaya iletilmesi aşağıdaki sırayla olur:

- 1) Bilgi üreticinin kafasındaki model düşüncenin oluşması.
- 2) Belli kriterlere uyularak bu modelin ses veya görüntü sembolleri halinde tanımlanması.
- 3) Bu sembollerin kodlanarak iletişim için uygun hale sokulması.
- 4) Kodlanmış sembollerin istenen hedefe iletilmesi.
- 5) Sembollerin çözülmesi ve orijinal haline geri çevrilmesi.
- 6) Alıcının kafasında orijinal düşünce modelinin tekrar meydana getirilmesi.

2.2 Haberleşme Türleri

Haberleşme, çok geniş ve derin bir konudur. Bu bağlamda haberleşmeyi ve bilginin nasıl iletilildiğini iyi kavrayabilmek için sınıflandırılması gerekir. Ancak bu sınıflandırmaların kesinlik arz etmediği belirtilmelidir. Yapılan sınıflandırmada dikkate alınan belirleyiciler;

1. Bilginin iletildiği ortam (Ortam türüne göre) ve
2. İletimde kullanılan sinyaller (Sinyal türüne göre) dir.

2.2.1 Ortam Türüne Göre Haberleşme Türleri

Haberleşmenin temlinde her zaman bilginin iletildiği bir kaynak (source) ve bilginin iletilmek istendiği bir hedef (destination) vardır. Kaynak ve hedef arasında da her zaman bilginin iletildiği bir yayının ortamı (iletişim kanalı) (Şekil 2.1) vardır. Bu ortamlar kimi zaman fiziksel kablolu bir medya (bakır, fiber optik kablo vs.) olabileceği gibi kimi zamanda hava (atmosfer) veya boşluk da olabilir.

Genel olarak ortam türüne göre haberleşme konusu iki sınıfta incelenir;

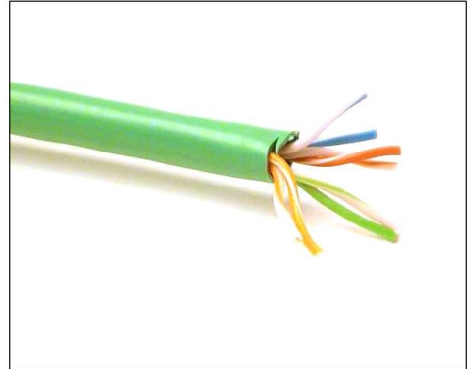
1. Kablolü Haberleşme (Wired Communication) ve
2. Kablosuz Haberleşme (Wireless Communication)

2.2.1.1 Kablolü Haberleşme

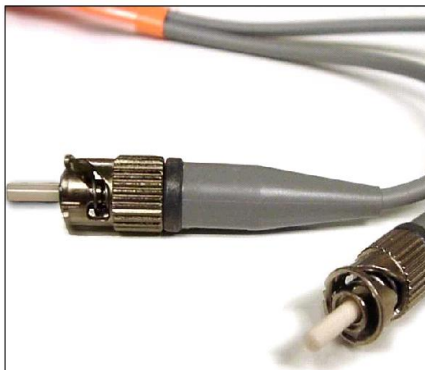
Bilgi iletimi için bakır, dalga klavuzları veya fiber optik kablolardan faydalanılır. İletişim elektrik, elektromanyetik veya optik sinyaller aracılığıyla bu ortamlar üzerinden gerçekleştirilir. Bakır kablolarda (Şekil 2.3-a ve b) haberleşme sinyali elektriksel sinyaller şeklinde iletilirken, fiber optik kablolarda (Şekil 2.3-c) sinyaller ışık demetleri şeklinde iletilir. Bir diğer kablolu iletim ortamı **dalga klavuzları**'dır. Bunlar çok yüksek frekanslı elektromanyetik dalgaları taşımak için düzenlenmiş metal borulardır. Dalga klavuzları sinyal iletime açısından iletkenlere benzer. Ancak iletkenler elektrik akımını iletirken, dalga klavuzları elektromanyetik dalgaları iletir. Ayrıca iletkenlerden akımın geçebilmesi için kapalı bir elektrik devresi gerekirken dalga klavuzlarında buna gerek yoktur. Şekil 2.3-d'de örnek dalga klavuzları görülmektedir.



a) Koaksiyel Kablo



b) Twisted Pair (TP) Kablo



c) Fiber Optik Kablo

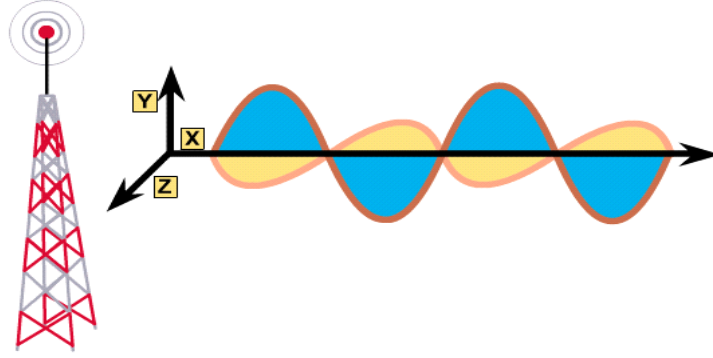


d) Dalga Yönlendirici

Şekil 2.3 : Kablolü ortam türleri

2.2.1.2 Kablosuz (Wireless) Haberleşme

Bilgi iletimi için hava veya boşluk'tan faydalanılır. İletişim elektromanyetik dalgalar şeklinde bu ortamlar üzerinden gerçekleştirilir.



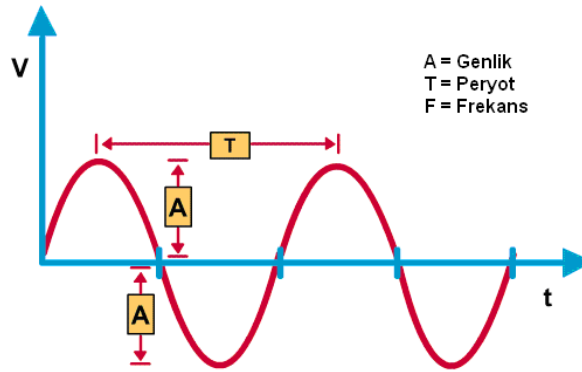
Şekil 2.4 : Elektromanyetik dalgalar ve kablosuz iletişim

2.2.2 Sinyal Türüne Göre Haberleşme Türleri

Sinyal türüne göre sınıflandırma yapılırken temel olarak taşıyıcı sinyalinin durumuna bakılır. Haberleşme alanında temelde iki tür sinyal söz konusudur. Analog ve Sayısal.

2.2.2.1 Analog Haberleşme

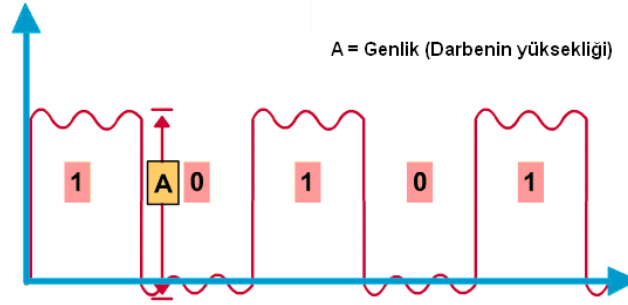
Bazı haberleşme dönüştürücüleri, orijinal bilgi enerjisinin ani değişimlerini doğrudan takip eden elektronik sinyaller üretirler. Bu tip sinyallere “**Analog Sinyal**” adı verilir. Örneğin, bir mikروفon kendisine uygulana ses enerjisinin değişimini takip eden bir elektronik sinyal üretir. Analog sinyaller kullanılarak yapılan haberleşmeye “**Analog Haberleşme**” adı verilir.



Şekil 2.5 : Analog Sinyal

2.2.2.2 Sayısal (Dijital) Haberleşme

Bazı sistemlerde dönüştürücü, sistemin her iki ucundaki insanlar veya makineler tarafından anlaşılabilir ve önceden belirlenmiş kod darbeleri veya değimleri şeklinde elektronik sinyaller üretir. Bu tip sinyallere “**Dijital Sinyal**” adı verilir. Örneğin Network Sistemleri arasındaki haberleşmede kullanılan sinyalizasyon verilebilir. Dijital sinyaller kullanılarak yapılan haberleşmeye “**Dijital Haberleşme**” adı verilir.



Şekil 2.6 : Sayısal (Dijital) Sinyal

Sayısal haberleşmenin, analog haberleşmeye göre bazı üstünlükleri vardır. Bunlar:

1. Dijital Teknoloji kullanır. Bu nedenle geniş ölçekli entegrasyon ve yarı iletkenler sayesinde çok küçük maliyetli sistemlerin oluşturulabilmesine olanak sağlar.
2. Analog iletimde bir sinyal yükseltildiğinde gürültü de onunla birlikte yükselir. Sinyal birçok yükseltme istasyonundan geçtikçe gürültü birikir. Bununla birlikte dijital iletimde her yineleyici istasyonu darbeleri yeniden üretir. Böylece temiz (gürültüsüz) darbeler oluşturulur ve yeni bir temizleme işleminin yer aldığı bir sonraki yineleyiciye gönderilir. Sinyaller gürültülü bir ortamdan geçer fakat daha fazla bozulacağı yerde sürekli yeniden üretildiğinden gürültüden etkilenmez.
3. Dijital teknoloji ile çoğullama (multiplexing) işlemi daha rahat olduğundan iletim kapasitelerinin verimli şekilde kullanılması sağlanır.
4. Şifreleme ve kodlama tekniklerinin dijital sinyallere uygulanması daha kolay ve daha ekonomiktir.

2.3 Haberleşmedeki Kısıtlamalar

Herhangi bir haberleşme sisteminde amaç, kaynaktaki bilgi işaretini alıcı uça olduğu gibi geri elde etmektir. Ancak, bu amaca genellikle erişilemez. Bir başka deyişle, gönderilmiş olan bilgi, bazı kısıtlamalar nedeniyle, olduğu gibi geri elde edilemez. Bu kısıtlamalar bazı

kaçınılmaz doğa olayları yada iletişim tasarımını yapan mühendisler tarafından saptanır. O halde, alıcı tarafta bilgiyi aslına olabildiğince yakın bir biçimde elde etmek daha gerçekçi ve uygulanabilir bir yaklaşım olur. Bu nedenle birçok uygulamada, asıl bilgi işaretinden bazı sapmalara hoşgörü gösterilir.

Bir haberleşme sistemindeki ana kısıtlamalar aşağıdaki başlıklarda toplanabilir;

1. Gürültü ve Girişim
2. Bant genişliği
3. Sinyal Gücü

2.3.1 Gürültü ve Girişim

Modüle edilmiş bir işaret, iletim ortamında istenmeyen bozulmalara uğrar. Bu bozulmaların nedenleri; elektrik fırtınaları, şimşek, yıldırım gibi atmosfer olayları olabileceği gibi, elektrik aygıtları yada başka işaretler nedeniyle oluşan girişim gibi insan yapısı kaynaklar da olabilir. Bozulmaların başka bir kaynağı da haberleşme sisteminin kendi içinde ürettiği gürültüdür. Gürültü ve girişimin etkilerini azaltmak için çeşitli önlemler alınır. Gürültü düzeyi, kullanılan modülasyon türüne bağlıdır. Gürültü etkisinin az olduğu modülasyon türlerinde genellikle daha çok bant genişliği kullanılır.

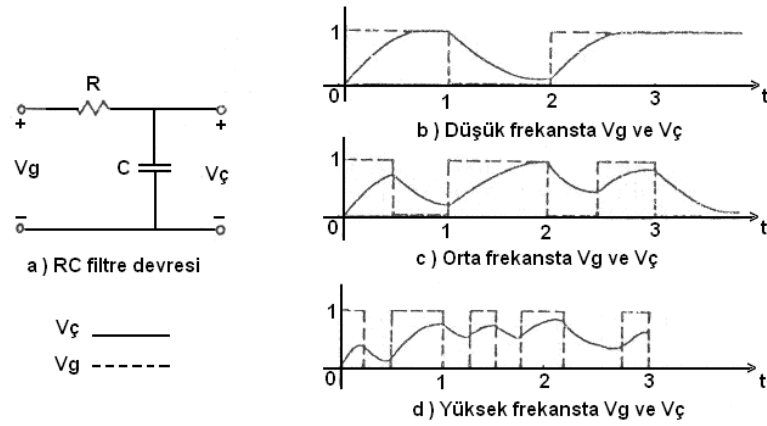
2.3.2 Bant Genişliği

Haberleşme sistemleri için başka önemli bir sınırlama ise bant genişliği kısıtlaması olup bu kısıtlama uluslararası iletişim kuruluşlarınca belirlenir. Elektromanyetik görüngen bir çok kullanıcı tarafından paylaşıldığı için, her kullanıcıya belirli bir bant genişliği verilir ve daha fazlasını kullanmasına izin verilmez. İzin verilen bant genişliği, kullanılan frekans bandına ve modülasyon türüne bağlıdır. Örneğin AM yayınında her kullanıcı 10 KHz'lik bir bant genişliği kullanır. Oysa FM yayınında 200 KHz'lik bir bant genişliği kullanımına izin verilir. Daha çok bant genişliği kullanıldığı için, FM'in gürültü ve girişimi bastırma özelliklerinin daha iyi olması beklenebilir.

Gürültünün azaltılması için bant genişliğinin artırılması, birbirleriyle çelişen iki kavramdır. Ancak, diğer kullanıcılar nedeniyle bant genişliği istenildiği gibi artırılamaz. Bu nedenle iyi bir iletişim sistemi, bu kısıtlamalar arasında kurulan dengenin sonucunda ortaya çıkar. Bu arada, ekonomik etkenlerin de göz önüne alınması gereklidir. Bazı modülasyon türlerinde (PCM) ideal koşullar altında bile bir ölçüde bozulma kaçınılmazdır. Bunun nedeni bu modülasyon türünün tersinir (reversible) olmamasıdır.

Bir kanal üzerinden gönderilebilecek maksimum bilgi miktarı “kanal sığası” olarak adlandırılır ve sistemin bant genişliğine bağlıdır. Belirli bir bant genişliği kaplayan bir işaret, bant genişliği daha dar olan bir sistemden geçerse bozulmaya uğrar. Sistemin bant genişliği azaldıkça bozulma daha da artar. Bozulmanın artması sonucu bilginin güvenli bir şekilde iletilme hızı da azalır. Sistemin bant genişliği, işaretin bant genişliğinden daha fazla ise bozulma olmaz. Bu nedenle herhangi bir işaretin bir sistemden bozulmaya uğramadan geçebilmesi için, sistemin bant genişliğinin sonsuz olması gerekir. Bunun ise pratikte gerçekleşmesi olanaksızdır.

Bilgi iletim hızının, işaretin frekansı ile ilişkili olduğu söylenebilir. Belli bir frekans değerinin üzerinde, haberleşme sistemi içerisindeki elektriksel devreler, doğaları veya yapım unsurları nedeniyle işarettaki anlık değişimleri takip edemezler.



Şekil 2.7 : RC devresinin frekansa bağlı tepkisi

Şekil 2.7 de gösterildiği üzere örneğin Alçak Geçiren bir RC Filtre devresi, işaretin hızı arttıkça bunları takip edemez hale gelir ve bir noktadan sonra işaret bozulmaya uğrar. Birde bu sinyale iletişim kanalı içerisinde gürültü eklendiği düşünülürse alıcı tarafta bu sinyalin doğru olarak alınması çok güç, hatta olanaksız olabilir.

Bu nedenle gönderilen işaretin alıcıda tekrar doğru bir şekilde elde edilebilmesi için bilgi iletim hızına bir üst sınır konmalıdır (konulur).

2.3.3 Sinyal Gücü

Yukarıda açıklananlar dışında bilgi iletim hızını etkileyen başka bir etken daha vardır ve bu etken işaret gücüdür (yada S/N işaret-gürültü oranıdır). Bu etken şu biçimde açıklanabilir. Sistem bant genişliği değiştirilmeden, işaret gücü artırılarak bilgi iletim hızı artırılabilir. Ancak uygulama açısından ve ekonomik nedenlerle, işaret gücünü istenildiği kadar arttırmak olanaksızdır. O halde, sistem bant genişliği yada işaret gücü artırılarak bilgi iletim hızı artırılabilir. Ancak bu nicelikler yukarıda söz edilen nedenlerden dolayı rasgele arttırılmaz. Önemli olan; bu birbiriyle çelişen etmenlere en iyi (yada iyi) ve ekonomik şekilde bir yaklaşım (çözüm) ortaya koymaktır.

2.4 Çoğullama (Multiplexing)

Aktarım sistemlerinin daha verimli kullanılması, var olan olanakların, birden çok aygıt tarafından eşzamanlı olarak kullanılabilmesi ile mümkündür. Çoğullama, var olan kapasiteyi birden çok aygıtta paylaşmak için uygulanan bir tekniktir. Genellikle aktarım olanaklarının kapasitesi, iki aygıt arasında aktarılması istenen verinin gerektirdiği kapasiteden çok daha fazladır. Aynı ortam üzerindeki işaret sayısını çoklayarak (artırarak) bu kapasite birden çok aktarıcı arasında paylaşılabilir. Bu durumlarda asıl aktarım yolu devre (circuit) yada bağlantı (link) olarak adlandırılmaktadır. Her bir alıcı/vericiye ayrılan kapasiteye ise kanal (channel) adı verilir. Zaman Bölmeli Çoğullama (Time Division Multiplexing) ve Frekans Bölmeli Çoğullama (Frequency Division Multiplexing) yaygın olarak kullanılan iki yöntemdir.

Herhangi bir iletim hattının, verilen bir zaman aralığında maksimum bir bant genişliği vardır. Bir elektriksel iletişim hattı sadece iki uç arasındaki konuşmayı iletmek üzere ayrıldığı zaman önemli bir israf yapılmış olunur ve bu nedenle sistem verimi düşük olacaktır. Bu noktada sistemi daha verimli kullanabilmek amacıyla çoğullama (multiplexing) yapılır. Çoğullama, bir hat üzerinden birden fazla bilginin simültane yada sırayla iletilmesi olayıdır. İletim hattı “**kanal**” adı verilen sanal tünellere bölünür.

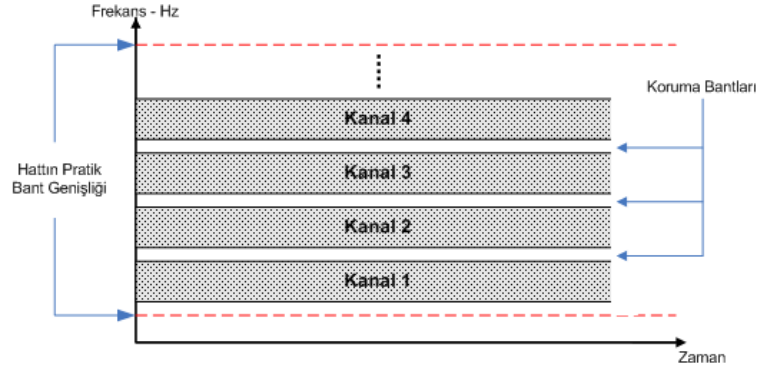
Çoğullama işlemi genelde iki şekilde yapılır;

1. Frekans Bölmeli Çoğullama (Frequency Division Multiplexing - FDM)
2. Zaman Bölmeli Çoğullama (Time Division Multiplexing - TDM)

FDM yönteminde kanal çoklaması yapılırken frekans bandı paylaşılır. TDM yönteminde ise zaman paylaşılır.

2.4.1 Frekans Bölmeli Çoğullama (FDM)

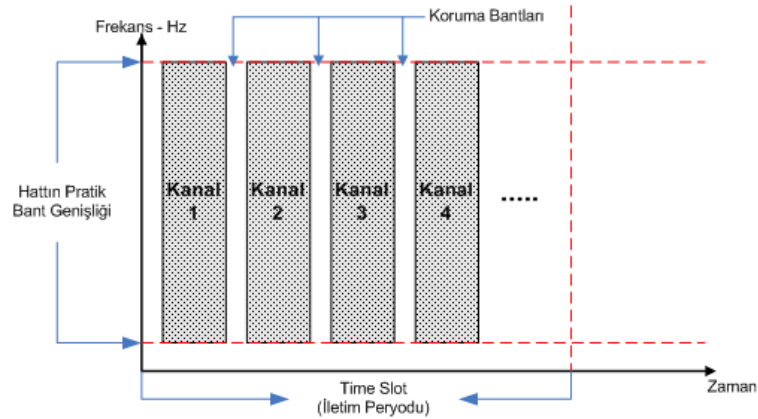
Bu yöntem daha çok analog işaretlerin çoğullanmasında kullanılır. Oldukça sık kullanılan bir çoğullama çeşididir. Bu yöntemde iletim hattı bant genişliği çok sayıda frekans aralıklarına bölünür. Diğer bir deyişle (Şekil 2.8 de gösterildiği üzere) her iletişim kanalına mevcut hat bant genişliğinin bir bölgesi ayrılır ve bütün zamanlarda ayrılan bu bant genişliği kullanılır. Kanal kapasiteleri uygulamaya göre değişkendir. Karışmayı önlemek için kanallar, önlem amaçlı koruyucu bantlar ile birbirlerinden ayırılır. En iyi bilinen örneği kablolu TV sistemleridir.



Şekil 2.8 : Frekans Bölmeli Çoklama (FDM)

2.4.2 Zaman Bölmeli Çoğullama (TDM)

Bu yöntemde, her kanala, düzenli zaman aralıklarında (time slotlarda) hat bant genişliğinin tamamı ayrılır. Şekil 2.9’da gösterildiği gibi zaman aralıkları hattı paylaşan konuşma kanallarının sayısına bağlıdır. Her durumda koruma bantları karşılıklı girişimden (crosstalk) korunmak için komşu kanalları biri birinden ayırır.



Şekil 2.9 : Zaman Bölmeli Çoklama (TDM)

Bu yöntem daha çok sayısal (dijital) işaretlerin çoğullanmasında kullanır. Günümüzde, daha yoğun olarak tercih edilen ve kullanılan sayısal (dijital) haberleşmede, pratik anlamda TDM'in FDM'e karşı aşağıdaki üstünlüklerinden dolayı, daha çok TDM yöntemi kullanılmaktadır;

TDM'in FDM'e göre üstünlükleri şunlardır;

1. Kanal kapasitesi daha yüksektir.
2. Gürültü oranı daha düşüktür.
3. Güvenilirliği daha yüksektir.
4. Fiziki boyutları daha küçüktür.
5. Maliyeti daha düşüktür.
6. İşletme kolaylığı sağlar.

Zaman bölmeli çoğullamanın genelde iki değişik şekli kullanılmaktadır:

1. **Senkron TDM:** İletişim ortamındaki veri oranının, gönderilecek sinyaller için gerekli veri oranını aştığı durumlarda zaman bölmeli çoklayıcılar kullanılır. Çok sayıda veri taşıyan analog veya sayısal sinyal, her sinyali zaman içerisinde parçalara ayırarak eşzamanlı bir biçimde taşıyabilir. Bu yöntemde periyodik olarak her bir kullanıcı bant genişliğinin tamamını kısa bir zaman aralığı içerisinde kullanır. Bu yöntem Round Robin Yöntemi olarak ta bilinir.
2. **İstatistiksel TDM:** Senkron TDM'de bir çerçevenin içerisindeki pek çok zaman aralığı boşa harcanır. İstatistiksel TDM asenkron veya akıllı TDM olarak da bilinir. İstatistiksel TDM zaman aralıklarını isteğe göre dinamik olarak ayırır. İstatistiksel TDM veriyi göndermeden önce tüm bellekleri tarar, çerçeve dolana dek verileri toplar ve çerçeveyi gönderir.

3 MODÜLASYON

3.1 Giriş

Modülasyon, bildiri işaretini iletme için daha uygun bir biçime sokmak için yapılır. Bir bilgi sinyalinin, yayılım ortamında iletilebilmesi için, ortam içerisinde rahatlıkla hareket edebilen başka bir taşıyıcı sinyal üzerine aktarılması olayına “**modülasyon**” adı verilir. Modülasyon yöntemlerinin amacı, bilgi sinyalini var olan bir haberleşme kanalına uydurmaktır. Bu yüzden modülasyon türü; mevcut gürültü, bant genişliği ve verici gücüne bağlı olarak seçilir. Periyodik bir işaretin (taşıyıcı), herhangi bir özelliği, bildiri işaretine (modülasyon işaretine) bağlı olarak değiştirilir. Elde edilen işarete “**modülasyonlu işaret**” denir.

Modülasyon işlemi haberleşme tekniğinin temelini oluşturur. Modülasyon işlemine olan ihtiyaç genel olarak üç grup altında özetlenebilir;

1. **Anten uzunluğu:** Haberleşme sistemlerinde, bir sinyalin transmisyonu için kullanılacak olan antenin minimum uzunluğu, işaretin dalga boyunun en az dörtte biri olmalıdır. Örneğin; eğer 0 ile 3KHz arasında bir konuşma işaretini iletilmek isteniyorsa, bu sinyalin dalga uzunluğu;

$$\lambda = \frac{c}{f} = \frac{300.000 \text{ (km/s)}}{3.000 \text{ (1/sn)}} = 100.000 \text{ metre} \quad (3.1)$$

bulunur. Buradan antenin minimum uzunluğu

$$l_{\min} = \frac{\lambda}{4} = 250.000 \text{ metre} \quad (3.2)$$

olarak elde edilir. O halde, anten uzunluğunun, pratikte gerçekleştirilebilecek bir boyda olması için, işaretin frekans spektrumu değiştirilmelidir.

2. **Birden fazla işaretin transmisyonu (Çoğullama):** Aynı anda bir veya birden çok işaretin, işaretler birbirine karışmaksızın bir noktadan, uzaktaki diğer bir noktaya iletilmesi, modülasyon yapmadan gerçekleştirilemez. Frekans (FDM) veya Zaman (TDM) Çoğullaması yöntemleriyle aynı kanaldan birden çok işaretin iletilmesi mümkün olmaktadır.
3. **Kanal gürültüsünü azaltmak:** Bir haberleşme sisteminde, aktarılan sinyaller iletim ortamının özelliklerine bağlı olarak zayıflamakta ve gürültü adı verilen istenmeyen bozucu etkenlerin tesirinde kalmaktadır. Bu önemli problemi çözmek için en uygun yol, işareti iletileceği ortamın özelliklerine uyan, gürültüden etkilenmeyecek bir biçime dönüştürmektedir.

Modülasyon, bilgi iletim sisteminin verimliliğini arttırmak amacıyla yapıldığından, taşıyıcı sinyal seçilirken bu sinyalin yayılım kanalı için ne kadar uygun olduğu göz önünde bulundurulmalıdır. İşte yukarıda belirtilen nedenlerden dolayı işaretin spektrumu daha uygun bir yere kaydırılır. Bu kaydırma işleminde; taşıyıcı dalga adı verilen ve genellikle yüksek frekanslı sinüzoidal (analog) veya dikdörtgen darbe katarı (dijital) biçiminde işaretlerden yararlanır. Kabaca, iletilecek işaretin bir taşıyıcı dalga yardımı ile iletim işlemine “**modülasyon**” adı verilir.

Modülasyon işlemi sınıflandırılırken, iki farklı bakış açısı kullanılabilir. Bunlar modülasyonda kullanılan taşıyıcı dalga ve bunu modüle eden bilgi sinyalidir. Bu noktada;

1. Taşıyıcı Dalga tipine göre
 - a. Sürekli Dalga Modülasyonu (Continuous Wave Modulation)
 - b. Darbe Modülasyonu (Pulse Modulation)
 2. Bilgi Sinyali tipine göre;
 - a. Analog Modülasyon
 - b. Sayısal (Dijital) Modülasyon
- şeklinde bir sınıflandırma yapılabilir.

Bunlardan darbe modülasyonu, genellikle sayısal haberleşmede kullanılır.

		Taşıyıcı Sinyal	
		Sürekli Dalga Mod.	Darbe Mod.
Bilgi Sinyali	Analog	AM FM PM	PAM PPM PDM
	Dijital	ASK FSK PSK	PCM

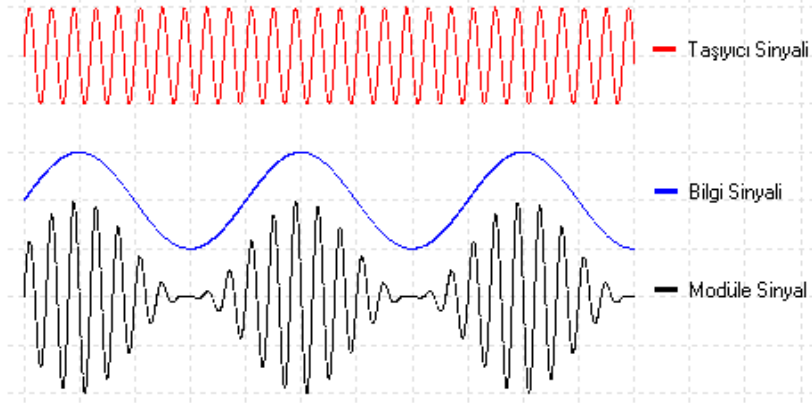
Tablo 3.1 : Bilgi ve Taşıyıcı sinyale göre modülasyon türleri

3.2 Sürekli Dalga (Analog) Modülasyonu

Sürekli dalga modülasyonu tipinde taşıyıcı olarak, belirli bir frekansta sinüzoidal bir dalga kullanılır. Sürekli dalga modülasyonu özellikle, konuşma, müzik veya resim gibi zamanın sürekli bir fonksiyonu olarak değişen analog işaretlerin iletimi için uygun olmaktadır ve kullanılmaktadır.

3.2.1 Genlik Modülasyonu (Amplitude Modulation – AM)

Tarihsel olarak kullanılan en eski modülasyon tekniğidir. Genlik modülasyonunda, taşıyıcının genliği bilgi sinyaline uygun olarak değiştirilir.



Şekil 3.1 : Genlik Modülasyonu (AM)

Genellikle taşıyıcı, kararlı bir sinüzoidal sinyaldir ve şu şekilde ifade edilebilir.

$$e_c(t) = E_c \sin(\omega_c t + \phi) \quad (3.3)$$

Faz açısı herhangi bir değer olabileceği için, ihmal edilebilir. Bu noktada iletilen sinyalde bir analog sinüzoidal sinyal şeklinde düşünülebilir. Öyleyse bilgi sinyali de aşağıdaki gibi ifade edilebilir;

$$e_m(t) = E_m \sin(\omega_m t) \quad (3.4)$$

$e_m(t)$ sinyali E_c 'yi modüle ettiğinde 3.3 nolu denklem şu şekle dönüşür:

$$e_{AM}(t) = (E_c + E_m \sin \omega_m t) \sin \omega_c t \quad (3.5)$$

E_c 'yi paranteze alınıp $\sin \omega_c t$ ile çarpıldığında 3.5 nolu denklem şu şekilde yeniden yazılabilir:

$$e_{AM}(t) = E_c \left(1 + \frac{E_m}{E_c} \sin(\omega_m t) \right) \sin(\omega_c t) \quad (3.6)$$

Burada E_m/E_c oranına modülasyon indeksi (m) adı verilir.

Genlik modülasyonun üç türü vardır:

1. Tam genlik modülasyonu (FAM)
2. Çift yan bantlı bastırılmış taşıyıcı genlik modülasyonu (DSBSC-AM)
3. Tek yan bantlı genlik modülasyonu (SSB-AM)

3.2.2 Açı Modülasyonu

$$V_c = E_c \cdot \cos \theta_c = E_c \cdot \cos(\omega_c t + \phi) \quad (3.6.1)$$

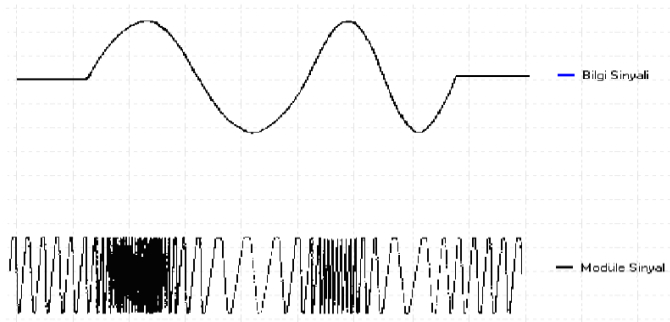
Yukarıdaki bağıntıda E_c 'yi değiştirerek yapılan modülasyona genlik modülasyonu (AM) denir. FM ve PM modülasyonları ise baseband θ_c 'de yapılan değişiklikler sonucu oluşur. ω_c 'de değişiklik yapılıncsa frekans modülasyonu, ϕ_c 'de yapılan değişiklikler de faz modülasyonuna neden olur. Son iki modülasyon tipi kullanım açısından sık sık birbirine çok benzer görünür ve bunların benzer analitik işlemleri olmasına karşın pratik olarak etkileri tamamen farklıdır. Bu fark baseband kare dalga için net bir şekilde görülebilir ancak sürekli değişen bir baseband sinyal için bu farkı görmek zordur.

Bu modülasyon tekniği daha fazla bant genişliği gerektirmesine karşın, daha üstün gürültü özellikleri nedeniyle yaygın olarak kullanılmaktadır. İki türü vardır;

1. Taşıyıcının frekansı ($\omega_c t$) bilgi sinyaline uygun olarak değiştirildiğinde, oluşan modülasyona “**Frekans Modülasyonu (FM)**” denir
2. Taşıyıcının fazı (ϕ_c) bilgi sinyaline uygun olarak değiştirildiğinde, oluşan modülasyona “**Faz modülasyonu (PM)**” denir.

3.2.2.1 Frekans Modülasyonu (Frequency Modulation – FM)

İlk olarak 1920’lerde ortaya çıkmış ve matematiksel analizi ise ilk olarak Bell Telefon Laboratuvarları’nda çalışan ünlü matematikçi John R. Carson tarafından gerçekleştirilmiştir. Frekans modülasyonunda, taşıyıcının frekansı bilgi sinyalinin genliğine orantılı olarak modüle edilir.



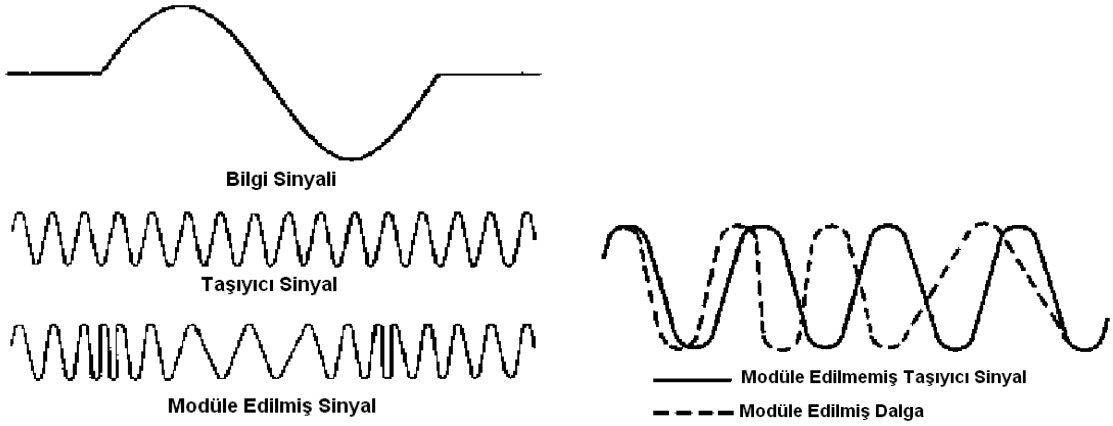
Şekil 3.2 : Frekans Modülasyonu (FM)

FM Modüle edilmiş sinyal matematiksel olarak aşağıdaki gibi ifade edilir;

$$e_{FM}(t) = E_c \cos \left[\omega_c t + \omega_c \int_0^t g(t) dt \right] \quad (3.7)$$

3.2.2.2 Faz Modülasyonu (Phase Modulation – PM)

FM modülasyonuna çok benzediği için çoğunlukla “**Dolaylı (indirect) FM**” olarakta bilinir. Taşıyıcı sinyalin faz açısı, bilgi sinyaline uygun olarak modüle edilir.



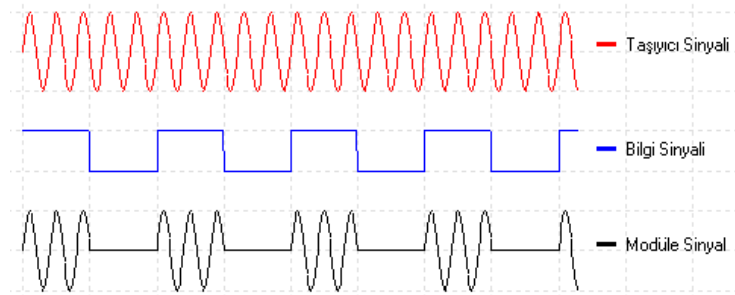
Şekil 3.3 : Faz Modülasyonu (PM)

PM Modüle edilmiş bir sinyal matematiksel olarak aşağıdaki gibi ifade edilir;

$$e_{PM}(t) = E_c \cos \left(\omega_c t + \frac{\Delta \omega_{eq}}{\omega_m} \cos \omega_m t \right) \quad (3.8)$$

3.2.3 Genlik Kaydırmalı Anahtarlama (Amplitude Shift Keying – ASK)

İkilik mantıkla kodlanmış, Sıfır (0) ve Bir (1)'lerden oluşmuş bir bilgi sinyalinin sinüzoidal bir taşıyıcının genliğine bindirilmesi tekniğidir. Klasik Genlik Modülasyonuna benzer. Tek fark bilginin sayısal (dijital) bir sinyal olmasıdır. Özellikle PCM kodlanmış temel bant sinyallerin iletilmesinde kullanılır.



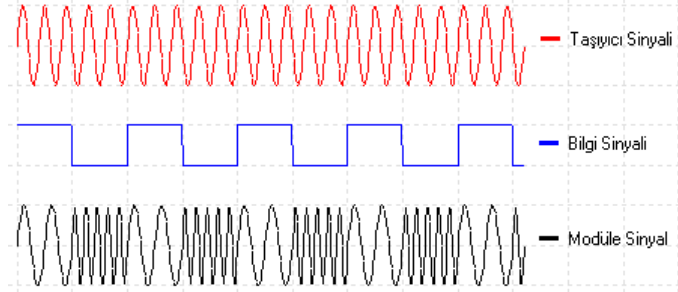
Şekil 3.4 : Genlik Kaydırmalı Anahtarlama (Amplitude Shift Keying – ASK)

Genelde ASK'da ikilik 0 iletmek için 0V genlik seviyesi ve ikilik 1 iletmek için E_c gibi bir seviye kullanılır.

$$e_{ASK}(t) = \begin{cases} 0 \rightarrow "0" \\ E_c \sin \omega t \rightarrow "1" \end{cases} \quad (3.9)$$

3.2.4 Frekans Kaydırmalı Anahtarlama (Frequency Shift Keying – FSK)

İkilik mantıkla kodlanmış, Sıfır (0) ve Bir (1)'lerden oluşmuş bir bilgi sinyalinin sinüzoidal bir taşıyıcının frekansına bindirilmesi tekniğidir. Klasik Frekans Modülasyonuna benzer. Tek fark bilginin sayısal (dijital) bir sinyal olmasıdır.



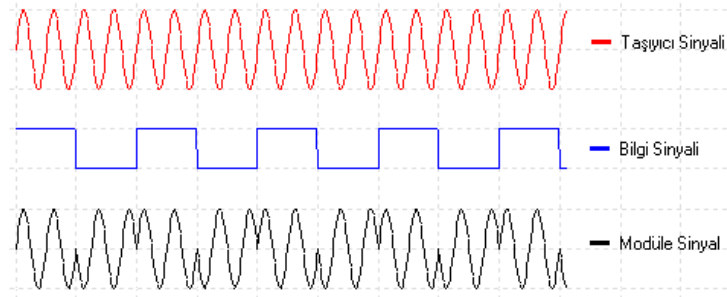
Şekil 3.5 : Frekans Kaydırmalı Anahtarlama (Frequency Shift Keying – FSK)

Genelde FSK'da, ikilik 0 ve 1 ifadeleri için, biri birinden rahatça ayırt edebilen iki farklı taşıyıcı frekansı (f_L ve f_H) kullanılır.

$$e_{FSK}(t) = \begin{cases} E_c \sin(2\pi f_L t) \rightarrow "0" \\ E_c \sin(2\pi f_H t) \rightarrow "1" \end{cases} \quad (3.10)$$

3.2.5 Faz Kaydırmalı Anahtarlama (Phase Shift Keying – PSK)

İkilik mantıkla kodlanmış, Sıfır (0) ve Bir (1)'lerden oluşmuş bir bilgi sinyalinin sinüzoidal bir taşıyıcının fazına bindirilmesi tekniğidir. Klasik Faz Modülasyonuna benzer. Tek fark bilginin sayısal (dijital) bir sinyal olmasıdır.



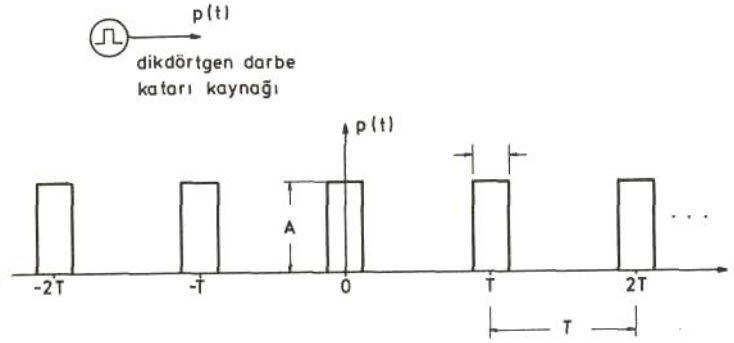
Şekil 3.6 : Faz Kaydırmalı Anahtarlama (Phase Shift Keying – PSK)

Genelde PSK'da ikilik 0 ve 1 için biri birinden rahatça ayırt edilebilir iki faz açısı kullanılır. Buna en iyi yaklaşım ikilik bir sinyal için biri birine (180°) zıt fazlı açılar kullanmaktır. Örneğin pratikte 0 için 0° ve 1 içinde π (180°) sıklıkla kullanılır.

$$e_{PSK}(t) = \begin{cases} E_c \sin(\omega t) \rightarrow "0" \\ E_c \sin(\omega t + \pi) \rightarrow "1" \end{cases} \quad (3.11)$$

3.3 Darbe Modülasyonu

Darbe modülasyonu genel olarak belirli bir darbe katarının parametrelerinin bilgi işaretinin bir fonksiyonu olarak değiştirilmesi sonucu elde edilir. Sürekli dalga modülasyonunun aksine darbe modülasyonu sürekli olmayan ayırık-zamanlı bir işlemdir. Şekil 3.7'den görüleceği gibi, darbeler zamanın ancak belli T aralıklarında T süresince vardır. Bu özelliğinden dolayı, darbe modülasyonu doğal olarak ayırık mesaj işaretleri için uygundur.



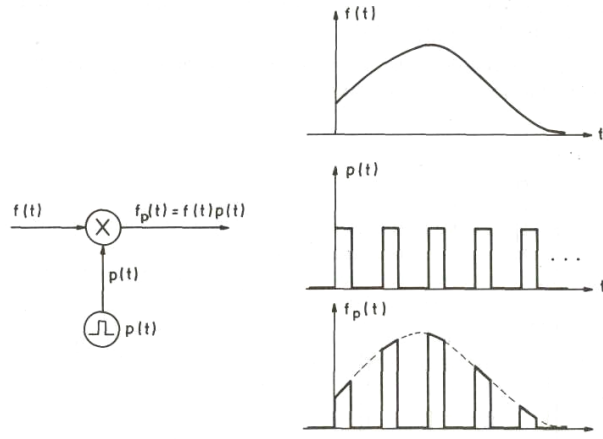
Şekil 3.7 : Darbe sinyali örneği

Mesaj işaretine göre sınıflama yapıldığında iki tip modülasyon ortaya çıkmaktadır.

1. Analog modülasyon: Sürekli bir mesaj işaretinin modülasyonu ile elde edilen modülasyon türüdür.
2. Sayısal modülasyon: Ayırık bir mesaj işaretinin modülasyonu sonucu elde edilen modülasyon türüdür.

Analog haberleşmede incelenen genlik, faz ve frekans modülasyonlarında işaret türü belirtilmemiştir. Bu nedenle, sürekli dalga modülasyonu için elde edilen sonuçlar hem sürekli-zamanlı hem de ayırık-zamanlı işaretler için geçerlidir.

Örneğin sürekli-zamanlı bir $f(t)$ işaretinin, $p(t)$ darbe katarı ile çarpımı sonucu bulunan işaretin değişimi Şekil 3.8'de gösterilmiştir. Buna göre çarpım sonucu elde edilen $f_p(t)$ darbe dizisinin genliği $f(t)$ işaretine uygun olarak değişmektedir. Yani, örnekleme darbelerinin orijinal dikdörtgen biçiminin bozulduğu görülmektedir. Oysa darbe genlik modülasyonunda darbelerin tepesi düzdür. Darbelerin genlikleri işaretlerle orantılı değişirken biçimleri bozulmamalıdır.



Şekil 3.8 : Bilgi sinyalinin darbe dizisiyle çarpımı

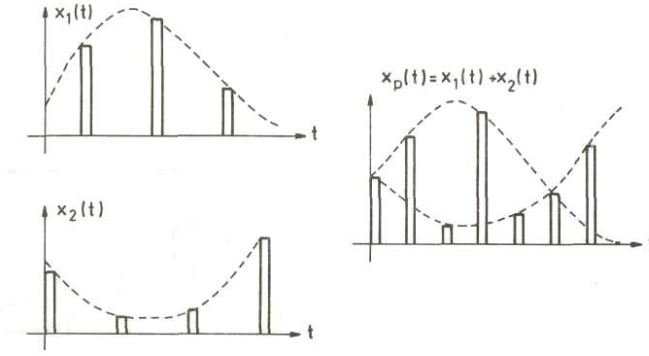
Bu örnekten görüleceği üzere, ayrık-zamanlı işaretlerin modülasyonu için darbe modülasyonu doğal olarak uygun bir modülasyon türüdür. Ancak, sürekli zamanlı işaretler için de darbe modülasyonu yöntemleri kullanılabilir. Sürekli bir $f(t)$ işareti verildiğinde bu işareti örnekleme işlemiyle bir ayrık-zamanlı işarete 'dönüştürür ve daha sonra darbe modülasyonu uygulayabiliriz.

3.3.1 Darbe Modülasyonun Üstünlükleri

Darbe modülasyonunda, taşıyıcının darbe katarı olması nedeniyle, sürekli dalga modülasyonuna göre üstünlükleri şöyle özetlenebilir:

1. Darbe modülasyonunda iletilen güç yalnız kısa darbeler içinde yoğunlaşmıştır. Sürekli dalga modülasyonundaki gibi sürekli olarak dağılmamıştır. Bu özellik tasarımcılara önemli kolaylıklar sağlamaktadır. Örneğin, yüksek güçlü mikrodalga tüpleri ve lazerler darbe biçiminde çalışmaya elverişli elemanlardır.
2. Darbeler arasındaki boşluklar, diğer mesajlara ait örneklerle doldurularak, tek bir haberleşme sistemi üzerinden birden fazla mesaj işaretinin iletilmesi sağlanabilir.

Şekil 3.9'da gösterilen bu işleme zaman bölmeli çoğullama (Time Division Multiplexing-TDM) adi verilir.



Şekil 3.9 : Darbe modülasyonunda çoklama özelliği

3. İşlemler ayrık türden işaretlerle yapıldığı için, son yıllarda tümleşik devre teknolojisindeki büyük gelişmeler, sayısal haberleşme devrelerinin gerçekleşmesini kolaylaştırmıştır.
4. Sayısal işaret işleme tekniklerindeki ilerlemeler, sayısal işaretlerin daha yaygın kullanılmasına neden olmuştur.
5. Bazı darbe modülasyonlu sistemler gürültü ve diğer bozucu işaretler açısından sürekli dalga haberleşmesinden daha güvenilir bulunmaktadır.

Sürekli dalga modülasyonu ile darbe modülasyonu arasındaki bir diğer önemli bir fark da, modüle edilmemiş dalga katarının çok önemli miktarda alçak frekans bileşenleri hatta doğru akım bileşeni içermesidir. Bu nedenle, modüle edilmiş dalgaların belirli bir frekans bandına sahip kanallardan iletilebilmesi için ikinci bir modülasyon işlemine ihtiyaç duyulur. Bu yüksek frekanslı kanallarda bir sürekli dalga modülasyonu, örneğin FM kullanılabilir. Bir kablo üzerinden yapılan iletimde ise, daha farklı modülasyon yöntemleri kullanılır.

Özellikle Darbe Modülasyonu tipinde, taşıyıcı sürekli bir sinyal olmayıp darbeleri bir kare dalgaya benzer. En büyük özelliği bilginin temel bantta kalması ve daha yüksek bir frekanslı bir taşıyıcıya çevrilmemesidir. Genellikle nicelemeli ve nicelemesiz olmak üzere iki kategoriye ayrılır.

Her iki kategoride de darbenin değiştirilebilen 3 parametresi vardır:

1. Darbenin uzunluğu,
2. Yüksekliği (genliği) ve
3. Konumu

Darbe modülasyonu gerçekleştirilirken, ilk önemli adım, verilen bir mesaj işaretini ayırık örneklerden oluşan değerlere dönüştürme (örnekleme) işlemidir.

3.3.2 Örnekleme Teoremi

Bu teorem Shannon örnekleme teoremi veya Nyquist örnekleme teoremi olarak da adlandırılır. Teorem temel olarak sınırlı bantlı bir $x(t)$ analog işaretinin yeniden ve bozulmadan elde edilebilmesi için örnekleme işleminin nasıl olması gerektiğini belirtmektedir. Örnekleme teoreminin ifadesi, ideal örnekleme durumu için şöyle verilebilir.

Örnekleme teoremi; W frekansı ile bantı sınırlı bir $x(t)$ işaretinden eşit T zaman aralıklarında alınan $x(nT)$ örnek değerleri kullanılarak, işaretin tek ve bozulmasız olarak elde edilmesi için gerek ve yeter koşul

$$f_s = \frac{1}{T} \geq 2W \quad (3.12)$$

olmasıdır. Bu koşul altında $x(t)$ işareti

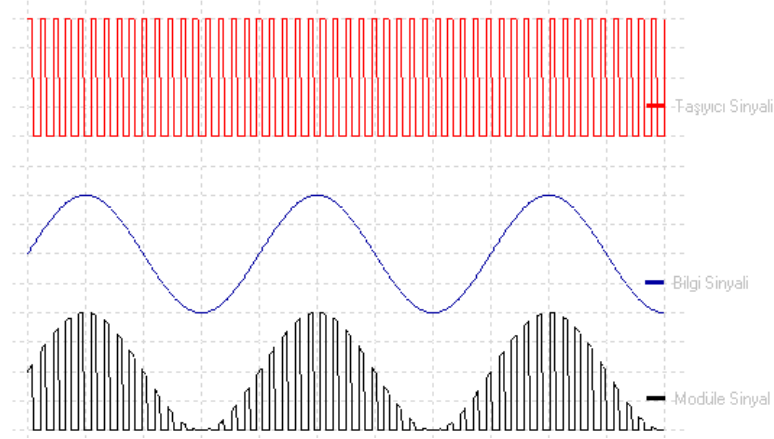
$$x(t) = \sum_{k=-\infty}^{\infty} x(kT) \frac{\sin \left[\pi \left(\frac{t}{T} - k \right) \right]}{\frac{t}{T}} \quad (3.13)$$

ifadesinden elde edilebilir. Burada T [saniye] örnekleme aralığı, $f_s=1/T$ [örnek/saniye] “örnekleme hızı (Nyquist hızı) veya örnekleme frekansı” olarak adlandırılır. Gerekenden daha düşük bir frekansla örnekleme yapılırsa bu durumda “katlanma” oluşur.

3.3.3 Darbe Genlik Modülasyonu (Pulse Amplitude Modulation – PAM)

Darbe Genlik Modülasyonunda analog bir sinyal, bir kare dalga taşıyıcı sinyalin genliğine modüle edilir. Çıkış bilgi sinyali ile taşıyıcı sinyalin çarpımıdır. Çıkış sinyali aşağıdaki gibi ifade edilebilir.

$$e_{PAM}(t) = f(t)f_s(t) = E_m \cos \omega_m t \cdot f_s(t) \quad (3.14)$$



Şekil 3.10 : Darbe Genlik Modülasyonu (Pulse Amplitude Modulation – PAM)

3.3.4 Darbe Süresi Modülasyonu (Pulse Duration Modulation – PDM)

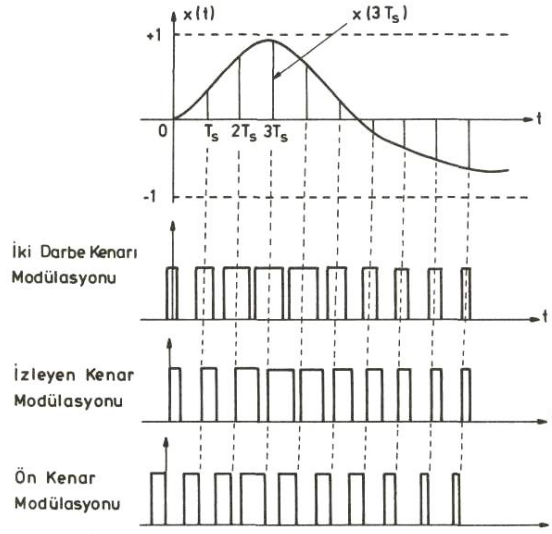
Darbe Genişlik Modülasyonunda bilgi sinyali, taşıyıcı kare dalga sinyalin darbe genişliğine modüle edilir. Bilgi sinyalinin genliği ile orantılı olarak taşıyıcı sinyalindeki kare dalgaların genişliği artar veya azalır. Bir PDM de kare dalganın ön, arka veya her iki kenarı birden yer değiştirebilir.

Bu modülasyon tekniği Darbe Genişlik Modülasyonu (Pulse Width Modulation – PWM) olarak bilinir. Şekil 3.11’de gösterildiği gibi, bu modülasyon türü üç farklı biçimde gerçekleştirilebilir.

PDM işareti yaklaşık olarak, sürekli dalga modülasyonunun bir türü olan açılı modülasyonuna benzemektedir. Bu nedenle, zaman ve frekans domenindeki ifadelerini kesinlikle analitik olarak ifade etmek mümkün değildir. Ancak, büyük bir yaklaşıklıkla, Fourier serisi açılımından yararlanarak bazı sonuçlar elde edilebilir.

Bu yaklaşıklıkların dayandığı iki gözlem sonucu şunlardır;

1. Mesaj işareti $x(t)$ 'nin komşu örnek değerleri arasında büyük değer farkları yoktur. Yani, $x(t)$ 'nin frekans bileşenleri genellikle, W bant genişliğinin çok altında yoğunlaşmıştır.
2. Pratikte, modülasyonlu darbeler için izin verilen maksimum darbe genişliği darbeler arasındaki süreden çok küçüktür. Bu gözlemlerin sonucunda, yaklaşık olarak PDM dalgası periyodu T_s olan periyodik darbe katarı biçiminde düşünülebilir.



Şekil 3.11 : Darbe Süresi Modülasyonu (Pulse Duration Modulation – PDM)

Genliği A , darbe süresi x ve periyodu T_s olan dikdörtgen darbe treninin Fourier serisine açılımı

$$x_p(t) = \frac{A\tau}{T_s} + \sum_{n=1}^{\infty} \frac{A}{\pi n} \left[\sin \frac{\pi n}{T_s} (2t + \tau) - \sin \frac{\pi n}{T_s} (2t - \tau) \right] \quad (3.15)$$

olur. Bu ifadelerde, τ darbe süresi $x(t)$ mesaj işaretiyle orantılı olarak değişecektir. Yukarıda belirtildiği gibi, darbe süresinin değişimi T_s periyodu yanında küçük olduğundan, m . darbenin süresi

$$\tau_m = \tau_0 + t_0 x(mT_s); t_0 < \tau_0 \ll T_s \quad (3.16)$$

biçiminde yazılabilir. Bu gösterilimde, τ_0 ve t_0 modülasyon sabitlerini göstermektedir. Ayrıca, $|x(mT_s)| \leq 1$ varsayıldığından maksimum darbe süresi

$$\tau_0 - t_0 \leq \tau_m \leq \tau_0 + t_0 \quad (3.17)$$

olur. $\tau_m \ll T_s$ yaklaşıklığının yapılması durumunda, τ_m darbe süresi t 'nin sürekli bir fonksiyonu biçiminde

$$\tau_m = \tau = \tau_0 + t_0 x(t) \quad (3.18)$$

olarak ifade edilebilir. (3.15) ve (3.18)'den

$$x_p(t) = \frac{A}{T_s} \tau_0 + \frac{A}{T_s} x(t) + \sum_{n=1}^{\infty} \frac{A}{\pi n} \left[\sin(n\omega_s t + n\phi(t)) - \sin(n\omega_s t - n\phi(t)) \right] \quad (3.19)$$

elde edilir. Burada,

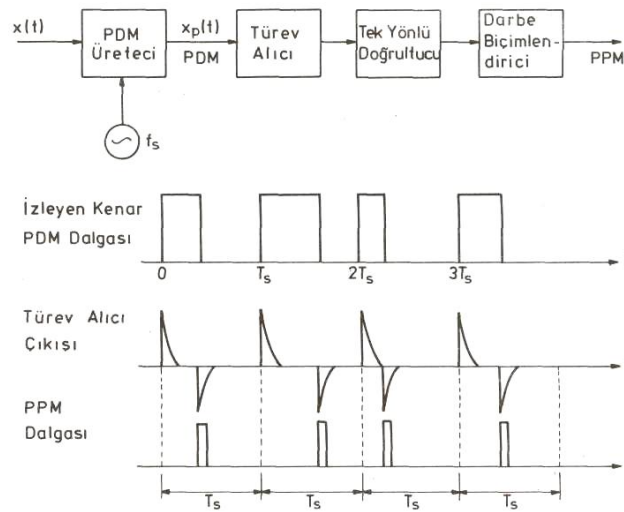
$$\phi(t) = \frac{\pi}{T_s} (\tau_0 + t_0 x(t)) \quad (3.20)$$

(3.19) denkleminde $x_p(t)$ 'nin spektrumu için şu gözlemler yapılabilir;

- (1) Bir doğru akım bileşeni vardır.
- (2) $x(t)$ mesaj işaretinin kendisi bulunmaktadır.
- (3) $f_s = 1/T_s$ 'nin katlarında $x(t)$ 'nin faz modülasyonlu dalgalarının süperpozisyonundan oluşmaktadır.

3.3.5 Darbe Konumu Modülasyonu (Pulse Position Modulation – PPM)

PPM modülasyonu ve PDM modülasyonu biri birine çok yakın iki modülasyon türüdür. Genellikle PPM dalgası, PDM modülasyonundan sonra ilave bir işlemle üretilir. Aslında, tıpkı faz modülasyonunda olduğu gibi, PDM'in temel kullanım alanlarından birisi de PPM üretmektir. PPM mesaj iletme bakımından PDM den daha üstündür.



Şekil 3.12 : Darbe Konumu Modülasyonu (Pulse Position Modulation – PPM)

PPM'de, $x(t)$ mesaj işaretine ilişkin bilgi, T_s periyodu içinde darbelerin konumlarında saklıdır. Bu nedenle, genlik modülasyonunda taşıyıcı bileşenin boşuna güç taşımaya benzer biçimde PAM ve PDM dalgalarındaki güçte, boşuna harcanan güçtür. Dolayısıyla, PPM'in düşük güçlerde çalışabilmesi, PDM'e göre en belirgin özelliğini ve üstünlüğünü oluşturmaktadır.

PPM doğrusal olmayan bir modülasyon olması nedeniyle, PPM işaretinin frekans spektrumu son derece zordur. PPM dalgası PDM dalgasına dönüştürülerek demodüle edilir.

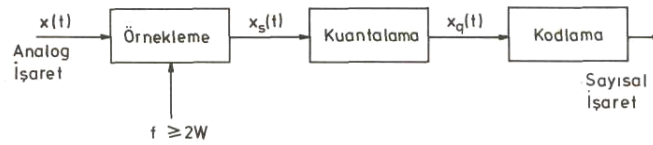
3.3.6 Darbe Kod Modülasyonu (Pulse Code Modulation – PCM)

Eğer sinyal sürekli ise sürekli sinyallerin yerine ayrık sinyal yerleştirme işlemine “Niceleme (quantizing)” adı verilir.

Darbe Kod Modülasyonu nicelemeli bir modülasyondur. Yani bilgi sinyalinin örneklenmiş değeri değil onun sayısal karşılığı kodlanır ve gönderilir. Nicelenmiş sinyalin yerine onu belirten sayısal bir değer ifade eden sinyal kullanmaya “**Kodlama**” denir.

Darbe modülasyonunda, analog bilgi sinyalinin ayrık-zamanda iletimi söz konusudur. PAM, PDM ve PPM modülasyonlarıyla darbenin sırasıyla genliğinin, genişliğinin ve bir periyod içindeki pozisyonunun sürekli olarak tüm işaret değerleri için değişimine izin verilmektedir. Bu aşamadan sonra bir iyileştirme de, zamanda ayrık duruma getirilmiş (örneklenmiş) bilgi işaretinin genliğinin de belirli sayıda ayrık seviyelere ayrılarak kuantalanmasıdır. M seviye sayısını göstermek üzere, PAM sistemlerinde kullanılan bu yönteme “**M-li PAM**” adı verilmektedir. Bu noktada her bir örnekleme seviyesi için ikilik (binary) bir kod kullanılması sonucu elde edilen modülasyona “**Darbe Kod Modülasyonu (Pulse Code Modulation – PCM)**” denir.

PCM’de bilgi taşıyan $x(t)$ işareti önce uygun bir örnekleme frekansıyla örneklenir. Daha sonra bu örnek değerler belirli kuantalama seviyelerine kuantalanır. Buna kuantalama işlemi adı verilir. Son olarak, her kuantalama seviyesi bir ikili kod kelimesiyle, yani sonlu sayıda $\{0,1\}$ dizisiyle gösterilir. İkilik kod kelimeler dizisine dönüştürülen bu işarete PCM dalgası adı verilir. Şekil 2.1’de PCM sistemin verici bölümünün blok diyagramı görülmektedir.



Şekil 3.13 : Darbe Kod Modülasyonu (Pulse Code Modulation – PCM)

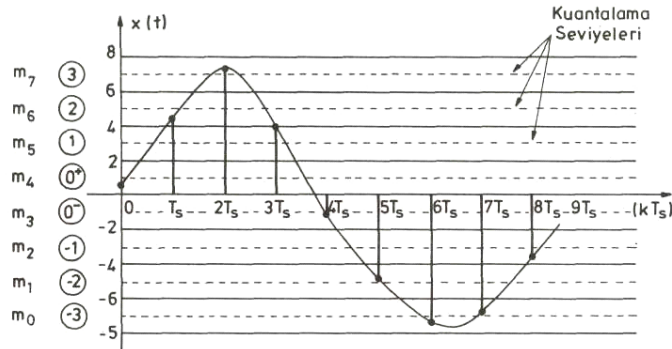
3.3.6.1 Kuantalama

Darbe genlik modülasyonunda örneklenmiş değerler belirli kuantalama seviyelerine yuvarlatılmadan iletilmektedir. Ancak, bu işlem işaretin gürültüye olan bağımsızlığı açısından bir yarar sağlamayacaktır. Bunun yerine, işaret genliğini belirli kuantalama seviyelerine yuvarlatmak ve her kuantalama seviyesine uygun bir kod kelimesi karşı düşürmek daha uygun olmaktadır.

$x(t)$ işaretinin maksimum ve minimum genlikleri A_{max} ile $-A_{max}$ arasında değişiyorsa ve bu aralıkta değişen genlik değerleri $Q = 2^n$ adet eşit kuantaya seviyesine bölünmek isteniyorsa, kuantalama aralığı veya adımı

$$a = \frac{2A_{max}}{2^n} \quad (3.21)$$

olarak tanımlanmaktadır. Kuantalama işleminde örnek değerlerin bulunduğu dilim belirlenir. Örneğin, -8 ve +8 volt arasında değişen bir $x(t)$ işaretini ele alalım. Bu aralık, 8 kuantaya seviyesine ayrılırsa, kuantalama aralığı $a=(16/8)= 2$ birim olacaktır.



Şekil 3.14 : Örnek kuantalama seviyeleri

Şekil 3.14'de gösterildiği gibi, her örnek değer 8 seviyeden birisine yuvarlatılır. Bu örnek için kuantaya seviyeleri $\pm 0, \pm 1, \pm 2, \pm 3$ olmaktadır. Her örnekleme anında elde edilen değer, en yakın kuantaya seviyesine kuantalanır. Tablo 3.2'de çeşitli işaret genliklerine karşı düşen kuantaya seviyeleri ve kod kelimeleri görülmektedir.

Giriş işareti genliği	Kuantaya seviyesi	Kod kelimesi
2.768	+ 1	001
2.051	+ 1	001
6.767	+ 3	011
-0.025	-0	100

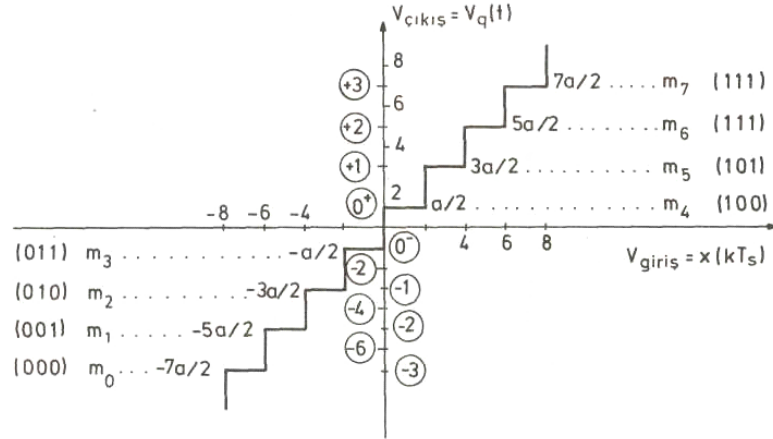
Tablo 3.2 : Örnek kuantalama değerleri

Kuantalama dilim sayısı Q arttıkça kuantalama gürültüsü de azalacaktır. Buna karşılık bir örneği belirlemek için kullanılması gerekli bit sayısı da artacaktır.

PCM sistemin alıcı bölümünde Şekil 3.13'deki işlemlerin tam tersi yapılarak analog mesaj işareti elde edilir.

3.3.6.2 Düzgün Kuantalama

Şekil 3.13’de gösterilen türden kuantalamaya düzgün kuantalama adı verilir. 8 seviyeli düzgün bir kuantalayıcıya ilişkin giriş-çıkış eğrisi Şekil 3.15’de gösterilmiştir.



Şekil 3.15 : Düzgün kuantalama eğrisi

-3, -2, -1, -0, +0, +1, +2, +3 deki 8 kuant seviyesi sırasıyla $m_0, m_1, m_2, \dots, m_7$ olarak simgelenilmektedir. Bu simgeler biti kod kelimeler kullanılarak kodlandırılır. Bu kodlama işleminde, üretilecek kod kelime uzunluklarının en kısa uzunlukta olması arzu edilir. Örneğin, verilen örnekte 8 kuant seviyesi için $8 = 2^3$ olduğundan, biti kodlamada kelime uzunluğu $n = 3$ olacaktır. Tablo 3.3’te kaynak seviyeleri ve ikilik kod kelimeleri görülmektedir.

Kaynak Seviyeleri	İkili Kod Kelimesi	Kaynak Seviyeleri	İkili Kod Kelimesi
m_0	000	m_4	100
m_1	001	m_5	101
m_2	010	m_6	110
m_3	011	m_7	111

Tablo 3.3 : 8 seviyeli kuantalama için kaynak seviyeleri ve ikilik kod kelimeleri

Darbe dizisi biçiminde oluşacak PCM dalgaları doğrudan kablolar üzerinden veya analog modülasyon yöntemleri kullanılarak iletilir. Alıcı tarafta PCM işareti demodüle etmek için yapılması gereken işlem oldukça basittir. Dalganın biçimine veya genliğine bakılmaksızın sadece bir dalganın varlığının veya yokluğunun belirlenmesi yeterli olmaktadır. İkilik işaret dizisi elde edildikten sonra, kod çözülerek kuantalanmış örnek değerler bulunabilir.

3.3.6.3 Kuantalama Hataları

Kuantalanmış örnek değerlerden, kuantalanmamış orijinal örnek değerlerin elde edilmesi mümkün değildir. Yani, tersine bir işlem olmayan kuantalama sonucunda, orijinal bilginin bir kısmı kaybolmaktadır. Kuantalanmış örnek işaret $x_q(t)$ mesaj işareti $x(t)$ 'nin yaklaşık bir değeri olduğundan bir bozulma söz konusudur. Bu bozulmaya “**kuantalama hatası**” adı verilmektedir.

$$e(t) = x(t) - x_q(t) \quad (3.22)$$

ifadesiyle gösterilen kuantalama hatasının etkisi bir toplamsal gürültünün etkisine eşdeğerdir. Bu nedenle, bu bozulma “**kuantalama gürültüsü**” olarak adlandırılır. Bu gürültü tamamen ortadan kaldırılamamakla beraber çeşitli yöntemlerle etkisi azaltılabilir.

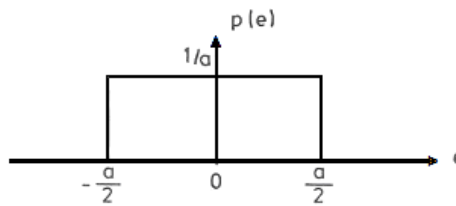
Giriş-çıkış karakteristiği Şekil 3.15'deki gibi düzgün kuantalayıcı kullanılması durumunda, kuantalama gürültüsü istatistiksel olarak modellenabilir. Kuantalama seviyesi (adımı) “ a ” olan kuantalama gürültüsünün olasılık yoğunluk fonksiyonu

$$p(e) = \begin{cases} \frac{1}{a} & , \quad -\frac{a}{2} \leq e \leq \frac{a}{2} \\ 0 & , \quad \text{diğer} \end{cases} \quad (3.23)$$

olur. $p(e)$ 'nin değişimi Şekil 3.16'da gösterilmiştir. (3.23) nolu ifadeden kuantalama gürültüsünün karesel ortalaması:

$$\langle e^2 \rangle = E[e^2] = \int_{-a/2}^{a/2} e^2 p(e) de = \frac{a^2}{12} \quad (3.24)$$

olarak bulunur. Ayrıca, ortalamannın $E[e] = 0$ olduğu görülmektedir.



Şekil 3.16 : Kuantalama hatasının olasılık dağılımı

İşaretin maksimum genliği A_{max} ise, n-bitlik bir kodlamada, kuantalama adımı (3.21) nolu ifadeden bulunabilir. Örneğin, en büyük gerilimin 2 volt olması durumunda 8 bit için her bir adım $a=(4/8)=0.5$ volt olur.

A_{max} genliğine göre ayarlanmış n-bitlik bir kuantalayıcı genliği A olan bir sinüzoidal işarete uygulanırsa, işaretin gürültüye oranı (S/N) şöyle hesaplanabilir:

$$\left(\frac{S}{N}\right) = \frac{\langle x^2(t) \rangle}{\langle e^2(n) \rangle} = \frac{A^2 / 2}{a^2 / 2} = \frac{3}{2} 2^{2n} \left(\frac{A}{A_{max}}\right)^2 \quad (3.25)$$

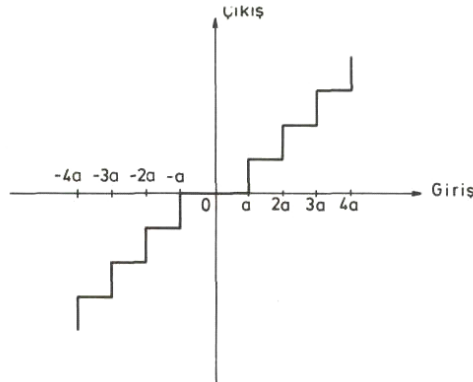
veya desibel cinsinden;

$$\left(\frac{S}{N}\right)_{dB} = 10 \log_{10} \left(\frac{S}{N}\right) = 1.76 + 6.02n + 20 \log_{10} \left(\frac{A}{A_{max}}\right) \quad (3.26)$$

olarak bulunur.

3.3.6.4 Boş Kanal Gürültüsü

(3.26) nolu denklem ifadesinden görüldüğü üzere, işaret gürültü oranı (S/N), işaretin genliği küçüldükçe azalmaktadır. Eğer işaret, en küçük dilimden daha küçük ise, ($A < a/2$), gürültü işareten daha büyük olur. Bu durum, özellikle kanal boş olduğu zaman çok rahatsız edicidir. Boş kanal gürültüsünü önlemek için, kuantalama eğrisi Şekil 3.17'deki gibi ortası yatay olacak şekilde yeniden ayarlanır. Bu yeni kuantalamada, $x(t)$ "a" adımından küçükse daima sıfır çıkışı elde edilir.



Şekil 3.17 : Boş kanal gürültüsünü önleyici düzgün kuantalama eğrisi

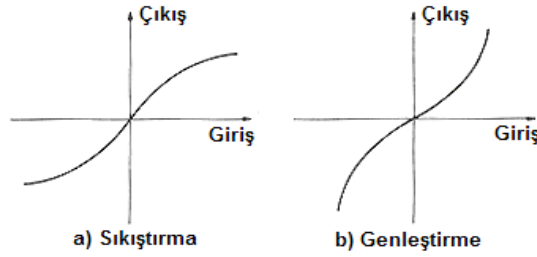
3.3.6.5 Düzgün Olmayan Kuantalama

Özellikle ses işaretlerinin istatistikleri incelendiğinde, küçük genliklere daha sık rastlandığı görülmektedir. Oysa, yukarıda küçük işaretlerde, kuantalama gürültüsünün rahatsız edici boyutlarda olacağı gösterilmiştir.

Bu gürültüyü azaltmak için başvurulacak ilk yöntem, adım büyüklüğünün azaltılması veya dilim sayısının artırılmasıdır. Ancak, bu durumda her bir örneği göstermek için kullanılması gereken bit sayısı artacağından, bu yöntem her zaman uygun ve ekonomik değildir. Diğer taraftan, çok seyrek olarak ortaya çıkan yüksek genlikli işaretler için gereksiz yere bir

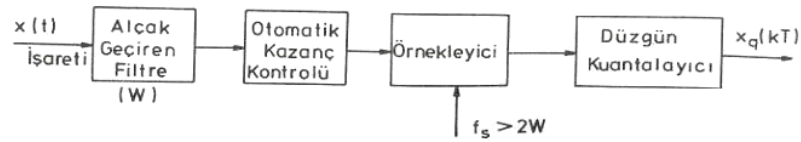
miktar dilim ayrılmış olacaktır. Eğer en büyük genlik küçük tutulursa, bu defa da kırılmalar meydana gelecektir.

Bununla beraber, büyük işaretler için büyük adım, küçük işaretler için de küçük adım kullanılarak işaret gürültü oranının aynı olması sağlanabilir. Bunu gerçekleştirmek için, haberleşme sistemlerinde, gönderici tarafında Sıkıştırma (Compressing) ve alıcı tarafında da Genleştirme (Expanding) işlemi yapılmaktadır. Şekil 3.18'de bu sıkıştırma ve genleştirme işlemlerinin lineer olmayan karakteristikleri görülmektedir.



Şekil 3.18 : A/D ve D/A çeviriciler için Sıkıştırma ve Genleştirme eğrileri

Bazı sistemlerde sıkıştırma işlemi doğrudan analog ses işareti üzerinde yapılır. Şekil 3.19'da gösterilen sistemde, otomatik kazanç ayarı kontrolüyle işaret seviyesi yaklaşık olarak kodlayıcının genlik seviyesine yakın tutulmaktadır. Bunun sonucu, kuantalama seviyelerinin büyük çoğunluğunun kullanılması sağlanarak, belirli bir kuantalama seviyesi ayısında sistemin en iyi biçimde çalışması gerçekleştirilir.



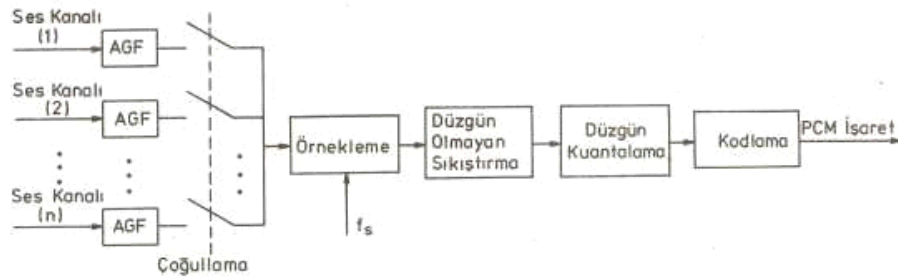
Şekil 3.19 : Haberleşme sisteminde işaret seviyesinin değiştirilmesi

Ancak, haberleşme sisteminde, birden fazla kanal varsa ve bu işaretler çoğullanıp tek bir kuantalayıcı ve kodlayıcıya uygulanıyorsa, sıkıştırma işleminin her kanal için ayrı ayrı yapılması yerine, kuantalayıcı ve kodlayıcı girişinde yapılması sistem tasarımı açısından daha verimlidir (Şekil 3.20).

Yukarıda belirtilen türden lineer olmayan sıkıştırma ve genleştirme devrelerinde genellikle logaritmik gerilim-akım karakteristiği olan diyotlar veya birkaç diyot ile birbirinden ayrılmış değişik ağırlıklı zayıflatıcıların örneklenmiş işaretin genliğine bağlı olarak devreye girip çıkmasından yararlanılır. Ancak, diyot karakteristikleri ile elde edilen bu sıkıştırma ve

açma eğrileri birbirine tam olarak uymadıklarından, çok doğru sonuç vermesi istenen kuantalayıcı ve kodlayıcılarda kullanışlı olmazlar.

CCITT¹ tarafından sayısal ses iletimi için önerilen başlıca iki tür sıkıştırma eğrisi vardır. Bu eğrilerin biçimi sinyalin istatistiksel özelliklerine bakılarak en uygun bir biçimde belirlenmiştir. Eğriler sıfırdan geçmekte ve sıfır civarındaki eğimi uçlardaki eğimden fazladır. Bu iki eğimin oranı sıkıştırma oranı olarak adlandırılır. Bu oran arttıkça işaretin dinamiği artmaktadır.

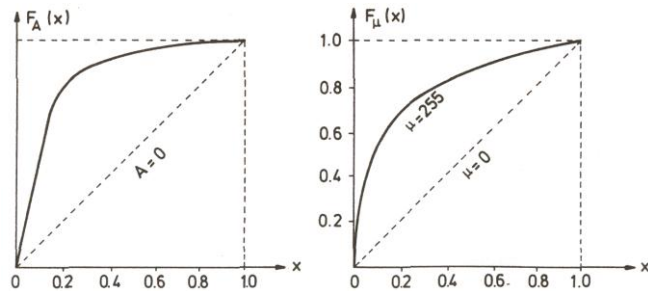


Şekil 3.20 : Çok kanallı sistemde işaret seviyesi değişimi

Standartlaştırılmış iki tip sıkıştırma eğrisi vardır:

- (1) Amerika ve Japonya'da kullanılan μ -tipi eğri;
- (2) Avrupa'da kullanılan A-tipi eğri

Şekil 3.21'de μ -tipi ve A-tipi eğriler görülmektedir.



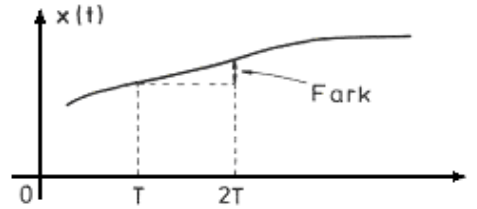
Şekil 3.21 : A ve μ tipi eğriler

¹ CCITT, Uluslar arası Telefon ve Telgraf Danışma Kurulu Komitesi (International Consultative Committee for Telephony and Telegraph). Uluslararası standartlar koyan bir kuruluştur.

3.3.7 Delta Modülasyonu (Delta Modulation – DM)

Bir önceki bölümde incelenen Darbe Kod Modülasyonunda (PCM), $x(t)$ analog mesaj işaretinden alınan örnekler belirli kuanta seviyelerine yuvarlatıldıktan sonra kodlanmaktadır.

Ancak görüntü işaretlerinde olduğu gibi uygulamada karşılaşılan bazı işaret türlerinde birbirine komşu örnekler arasında önemli ilişkiler vardır. Bu ilişkilerin büyük olması nedeniyle, bir örnek değerinin bilinmesi onu takip eden örnek değerlerinin de büyük olasılıkla belirlenebilmesini sağlar. O halde, her örneğin değerini ayrı ayrı göndermek gerekmez. Gerçek değerler yerine işaretteki değişimler (farklar) gönderilebilir (Şekil 3.22). Böylece, her bir örnek için gönderilen bit sayısında büyük bir azalma sağlanabilir.



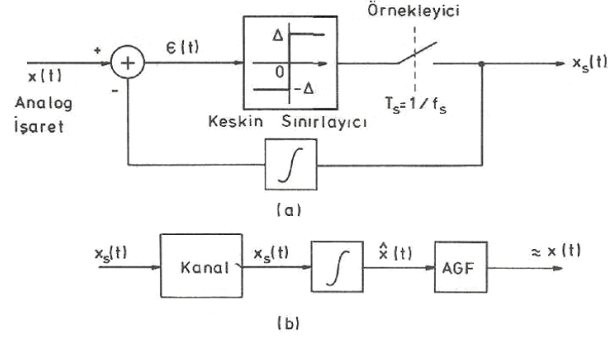
Şekil 3.22 : Örnekler arasındaki fark değeri

Bu yöntemde birbirini izleyen örnekler arasındaki genlik farklarının kodlanması öngörüldüğünden, elde edilen kodlama tekniğine diferansiyel PCM veya DPCM adı verilir. DPCM'nin en basit şekli, iki örnek arasındaki farkın sadece pozitif veya negatif olduğu dikkate alınarak yapılan kodlamadır. Buna delta modülasyonu (DM) denir. Diğer bir anlatımla delta modülasyonunda arda arda gelen iki işaret örneğinin genlik farkı bir bit ile kodlanır. Bu biçimde tanımlanan delta modülasyonuna “**Doğrusal Delta Modülasyonu**” adı verilir.

Çok hızlı değişen işaretler için, bu yöntem uygun değildir. Bunun nedeni DM'nin eğim izleme yeteneğinin sınırlı olmasıdır. DM'nin eğim izleme yeteneği, sıkıştırma-genleştirme (companding) metotlarıyla artırılabilir.

3.3.7.1 Doğrusal Delta Modülasyonu

Doğrusal delta modülasyonuna ait blok diyagramı Şekil 3.2'de verilmiştir. Doğrusal DM, bir keskin sınırlayıcı, bir örnekleme ve geri besleme yolunda yer alan bir integral alıcıdan oluşmaktadır. Alıcı tarafta ise, bir integral alıcı ve birde alçak geçiren filtre vardır.

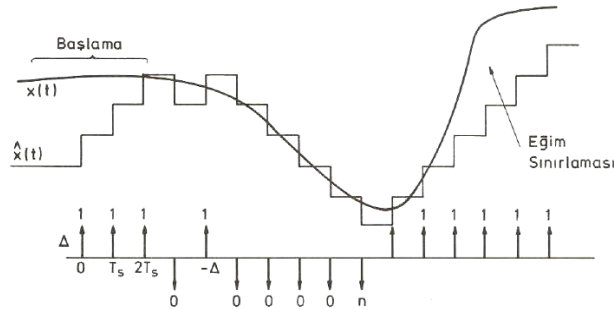


Şekil 3.23 : Doğrusal Delta Modülasyonu; a) verici, b) alıcı

Şekil 3.23a'da giriş işaretiyle geri besleme katında oluşturulan yaklaşık işaret arasındaki pozitif veya negatif olmasına göre bit çıkış darbeleri üretilmektedir. Bu bit darbelerden geri besleme yolunda bulunan integral yardımıyla yaklaşık işaret elde edilir. Giriş işaretiyle yaklaşık işaret arasındaki fark, negatif veya pozitif oluşuna göre iki mümkün seviyeden birine kuantalanır. Keskin sınırlayıcı çıkışı Nyquist frekansının çok üstünde bir f_s frekansı ile örneklenerek belirli çıkış darbeleri elde edilir.

DM'lu işaretin demodülasyonu Şekil 3.23b'de gösterilmiştir. İntegral alıcının çıkışındaki $\hat{x}(t)$ 'nin analog mesaj işareti $x(t)$ bir yaklaşığı olduğu Şekil 3.24'den anlaşılmaktadır. $\hat{x}(t)$ nin bir alçak geçiren filtreden geçirilmesiyle yaklaşık olarak $x(t)$ elde edilecektir. Burada AGF'nin görevi $x(t)$ 'deki süreksizlikleri düzgünleştirmektir. Filtre çıkışındaki işaretin $x(t)$ ile aynı olmamasının nedeni, modülasyon sırasında yapılmış olan kuantalama hatalarıdır. Ayrıca, $\hat{x}(t)$ darbe modülasyonunda gördüğümüz örneklenip ve tutulmuş işarete benzemektedir. Bu nedenle, $\hat{x}(t)$ 'den $x(t)$ işaretini elde etmek için girişi $\hat{x}(t)$ olan AGF'nin örnekleme ve tutma devresinin oluşturduğu bozulmayı giderecek biçimde tasarlanması gerekmektedir.

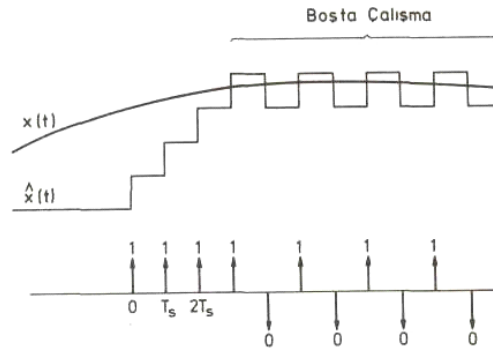
Kuantalama gürültüsüne ek olarak, delta modülasyonundaki kısıtlamalar nedeniyle bazı durumlarda $\hat{x}(t)$ işareti $x(t)$ 'yi yakından izleyemez. Sonuç olarak, AGF'nin çıkışı $x(t)$ 'den çok farklı olabilir.



Şekil 3.24 : Delta modülasyonunun çalışması

Şekil 3.24 incelendiğinde DM'nun çalışmasında şu istenmeyen durumlarla karşılaşılabilir.

1. Başlangıçtaki $\hat{x}(t)$ 'nin $x(t)$ işaretine yetişmesi için birkaç basamaklık bir süre gereklidir. Buna “**başlama süresi (start-up period)**” adı verilmektedir.
2. $x(t)$ sabit kaldığında, $\hat{x}(t)$ 'nin bir + bir – basamaklarla $\hat{x}(t)$ 'yi izlemeye çalıştığı görülmektedir. Şekil 3.25’de gösterilen bu duruma “**boşta çalışma gürültüsü (idling noise)**” adı verilir.
3. $x(t)$ hızlı değiştiğinde ise, $\hat{x}(t)$ sadece Δ kadar değişebildiğinden $x(t)$ 'yi izleyemez. Bu durum eğim sınırlaması (slope-overload) olarak adlandırılır.



Şekil 3.25 : Delta modülasyonunda boşta çalışma

3.3.7.2 Eğim Sınırlaması

Şekil 3.24’de gösterilen sınırlamayı azaltmak için çeşitli yollara başvurulabilir. Ancak, bunun sonucunda daha hızlı örnekleme hızı gerekecek ve sistem karmaşıklaşacaktır.

Bu noktada, eğim sınırlamasını önlemek için bir hesaplama yapılabilir. Buna göre $x(t)$ mesaj işaretinin W frekanslı bir sinüzoidal işaret olduğunu varsayalım. Bu işaretin en büyük eğimi,

$$m = 2\pi W \quad (3.27)$$

olur. Diğer taraftan, $\hat{x}(t)$ 'nin eğimi ise Δ/T_s veya Δf_s olmaktadır. O halde, eğim sınırlamasını önlemek için

$$\Delta f_s > 2\pi W \quad (3.28)$$

koşulu sağlanmalıdır. Bu ifadedeki Δ değeri, PCM’deki kuanta seviyesine karşı gelmektedir. $x(t)$ ile $\hat{x}(t)$ arasındaki farkın küçük olması için Δ küçük tutulmalıdır. (3.28) koşulunun sağlanması ancak f_s 'nin çok büyük seçilmesiyle mümkün olmaktadır.

3.3.7.3 PCM ile DM'nun Karşılaştırılması

Sistem yapısı açısından DM'lu sistemler çok basittir. Buna karşılık DM'lu işaret PCM'e göre çok daha büyük bir iletim bant genişliği gerektirir. Delta modülasyonunda örnekleme hızı hem Nyquist hızından büyük olmalı hem de (3.28) ilişkisini sağlamalıdır.

Örneğin, örnekleme hızı 8 KHz olan 8 bitlik bir PCM işaretinde saat frekansı 64 KHz olmasına karşılık buna denk bir DM'lu işaretin saat frekansı 100 KHz civarındadır. Diğer taraftan, bant genişliğini azaltmak için anlaşılabilirlikten fedakârlık edilirse, DM'nin başarısının daha iyi olduğu söylenebilir.

3.3.8 Adaptif Delta Modülasyonu (Adaptive Delta Modulation – ADM)

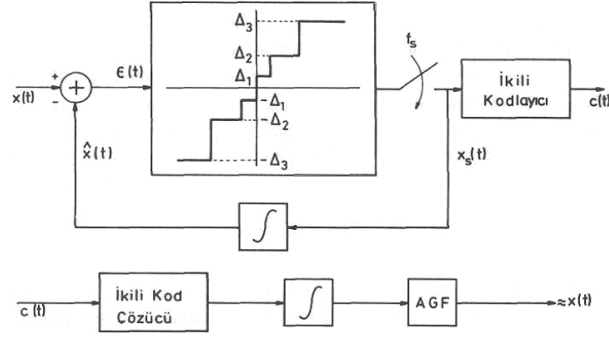
Delta modülasyonun uygulamadaki kolaylıkları yanında bazı eksiklikleri de vardır. Örneğin, doğrusal delta modülasyonundaki sabit basamak boyu (Δ), eğim sınırlaması, başlama süresi ve boşa çalışma gürültüsü bu olumsuzluklardan bazılarıdır. Örnekleme hızı, f_s , sabit tutulursa eğim sınırlamasını önlemek için Δ 'yı büyütmek gerekir. Ancak bu durumda kuantalama gürültüsü de boşa çalışma gürültüsüyle birlikte artacaktır. Δ küçük tutulduğunda ise, eğim sınırlamasını önlemek için örnekleme frekansını artırmak gerekir, f_s 'in artırılması da delta modülasyonlu işaretin iletim bant genişliğinin artmasına neden olmaktadır.

Yukarıda açıklanan sorunlara bir çözüm olarak basamak boyu Δ 'nın genliği, $x(t)-x^{\wedge}(t)$ 'nin farkına bağlı olarak değiştirilir. Böylece, örnekleme frekansını artırmadan $x^{\wedge}(t)$ 'nin mesaj işareti $x(t)$ 'yi daha yakından izlenmesi sağlanabilir.

$x(t)-x^{\wedge}(t)$ küçüldüğünde Δ da küçüleceği için boşa çalışma gürültüsü azalacaktır. $x(t)-x^{\wedge}(t)$ büyürken de Δ büyüyeceğinden eğim sınırlaması önlenmiş olacaktır. Basamak boyunun değişken olduğu bu modülasyon yöntemine “**Adaptif Delta Modülasyonu (Adaptive Delta Modulation – ADM)**” adı verilir.

3.3.8.1 Sabit Basamak Seviyeli Adaptif Delta Modülasyonu

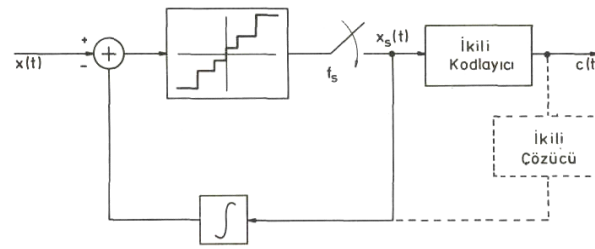
Şekil 3.23'deki 2 seviyeli ($\pm\Delta$) keskin sınırlayıcı yerine, başka bir doğrusal olmayan eleman konulur. Örnek olarak Şekil 3.26'da 6 seviyeli ($\pm\Delta_1, \pm\Delta_2, \pm\Delta_3$) keskin sınırlayıcı bir adaptif delta modülasyonlu sistem görülmektedir.



Şekil 3.26 : Örnek 6 seviyeli Adaptif Delta Modülasyonu

Bu sistemin çıkışı ikilik bir darbe dizisi olmayacaktır. Şekildeki eleman için 6 değişik değer alabilen darbelerden oluşan bir darbe dizisi olacaktır. Bu adaptif delta modülasyonunda herhangi bir değişiklik yapılmazsa, kanala 2'li bir dizi yerine 6'lı bir darbe dizisi gönderilir. Genellikle, çok seviyeli işaretlerin bir iletim ortamından gönderilmesinde türlü zorluklar ortaya çıkmaktadır. Bu çeşit çok seviyeli haberleşmede işaret gürültü oranı artmaktadır.

Böyle bir adaptif delta modülasyon sisteminin istenmeyen özelliklerini azaltmak için, Q -seviyeli darbe dizisini bir ikilik dizi oluşturacak şekilde kodlamak gerekir. Bu kodlanan işaret kanala gönderilir. Şekil 3.26'da gösterildiği gibi alıcı biti darbe dizisini Q -seviyeli darbe dizisine dönüştürecek bir kod çözücü gerekmektedir. Ayrıca, vericideki integral alıcıyı Q -seviyeli darbelerle beslemek yerine biti dizisinin kodunu çözerek bununla beslemek gerekebilir. Şekil 3.27'de bu türden bir adaptif delta modülatörünün blok diyagramı görülmektedir



Şekil 3.27 : Q -seviyeli Adaptif Delta Modülatörü

Doğrusal delta modülasyonuna göre daha üstün olan adaptif delta modülasyonunun olumsuz yanları da vardır. Bunlar;

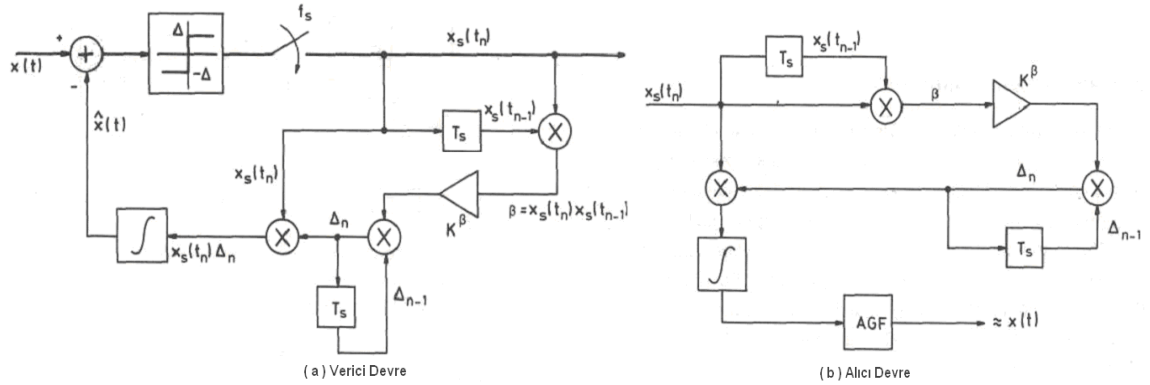
1. Sistemde kullanılan kodlayıcı ve kod çözücüleri sistemi karmaşıklığını artırır.
2. Q -seviyeli dizinin ikilik dizi olarak kodlanması bant genişliğini artırır.

3.3.8.2 Bir (1) Bit Bellekli Adaptif Delta Modülasyonu

Basamak boyunun her bir örnekleme anında adaptif olarak yenilenmesi için birçok metod vardır. Kuantalanmış fark dizisi $x_s(t_n)$, dalga biçiminin eğimiyle ilgili bilgi verir. $x_s(t_n)$ birbirini takip eden örneklerde işaret değiştirirse, dalganın eğiminin o yerel bölgede fazla olmadığı anlaşılır. Diğer taraftan, dalga biçiminin eğiminin dik olması durumunda birbirini takip eden $x_s(t_n)$ değerleri aynı işarettedir. Bu gözlemlerden yararlanarak $x_s(t_n)$ 'nin işaretine bağlı olarak adaptif basamak seviyeli bir algoritma oluşturulabilir. Jayant [1] tarafından geliştirilen basit kurala göre Δ_n basamak seviyesi

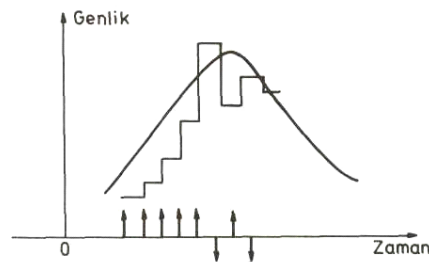
$$\Delta_n = \Delta_{n-1} K^{x_s(t_n)x_s(t_{n-1})} \quad n = 1, 2, \dots \quad (3.29)$$

ilişkisiyle değiştirilir. $K \geq 1$ bir sabit olup yerel distorsiyonu minimum yapacak biçimde seçilir. Bu adaptif algoritmayı kullanan sistemin blok diyagramı Şekil 3.28'de gösterilmiştir.



Şekil 3.28 : Adaptif Delta Modülasyonlu Sistem

Eğer $x_s(t_n) = x_s(t_{n-1})$ ise, kodlayıcı aşırı eğim yükü uyarısı alıyor demektir. Basamak boyu büyütülür. (3.28) ifadesinden $\Delta_n > \Delta_{n-1}$ olur. Benzer şekilde, eğer $x_s(t_n) \neq x_s(t_{n-1})$ ise, basamak boyu küçültülür. $K^\beta < 1$ ve $\Delta_n < \Delta_{n-1}$ elde edilir ($\beta = x_s(t_n)x_s(t_{n-1})$).



Şekil 3.29 : Adaptif Delta Modülasyonunda gürültü

Bu yöntemde, ani değişimleri izleyen aşırı yükselmeler ve yakalama süresi nedeniyle yeni bir gürültü oluşmaktadır (Şekil 3.29). Ayrıca, basamak boylarının sayısı istenildiği gibi değiştirilebilir. Bunun sonucu, $\hat{x}(t)$ 'nin $x(t)$ mesaj işaretini yakından izlemesi sağlanır.

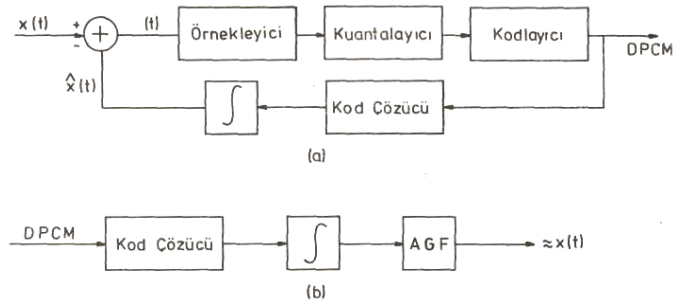
3.3.9 Diferansiyel PCM (Differential Pulse Code Modulation – DPCM)

Doğrusal olmayan elemanla bunu izleyen örnekleyci yerine bir PCM örnekleyci ve kuantalayıcı kullanarak özel bir adaptif delta modülasyon sistemi yapılabilir. Şekil 3.30'daki kodlayıcı ve kod çözücü birer PCM kodlayıcı ve PCM kod çözücüsüdür. Bu sisteme, delta modülasyonu ve PCM karışımı olduğu için “Diferansiyel PCM (DPCM)” adı verilir.

DPCM işareti, $x(t)-x^{\wedge}(t)$ fark işaretinin bir PCM işareti olarak gösteriliminden başka bir şey değildir. Ancak, burada adım boyları değişkendir. $\pm\Delta, \pm2\Delta, \dots, \pm Q\Delta/2$ arasında değişir. Q kuanta seviyesini gösterir. Bu nedenle, $x^{\wedge}(t)$ işareti $x(t)$ 'yi daha yakından izler. $Q=2$ için, DPCM basit bir delta modülasyona indirgenir.

Düzgün olmayan kuantalama kullanılması durumunda, boşa çalışma gürültüsü çok düşük olur. Başlama süresi ve eğim sınırlama olasılığı azalır.

$Q>2$ için DPCM'de devre karmaşıklığı PCM'dekine eşdeğerdir. $x(t)$ mesaj işaretinin örnekten örneğe çok değişmediği uygulamalarda $x(t)-x^{\wedge}(t)$ fark işareti daha az sayıda kuantalamayla uygun biçimde iletilebilir. Örneğin, DPCM ile görüntü işaretinin yeniden üretiminde $N=2^3=8$ kuanta seviyesi kabul edilebilir sonuçlar vermektedir. Aynı performansı elde etmek için, basit PCM'de $N=2^8=256$ kuanta seviyesine ihtiyaç vardır. DPCM işaretin bant genişliği PCM'e göre 3/8 oranında küçülmüştür.



Şekil 3.30 : Diferansiyel PCM (DPCM) sistem; a) verici, b) alıcı

4 SAYISAL KODLAMA

4.1 Giriş

Genel olarak saf ikilik bit akışları ile yapılan kodlama aşağıdaki nedenlerden ötürü pekte uygun değildir.

1. Hattaki işaretlerin büyük bir DC bileşeninin olması.
2. Bu nedenle hat işaretlerinin tekrarlayıcılar için gönderilmesi gerekecek DC güçten trafo köprüleri ile ayırmak mümkün olmayacaktır.
3. İşaret güç spektrumu alçak frekanslarda büyük bir bileşen içerecek ve bu da aynı kablo üzerindeki ses devrelerinde karışmaya neden olabilir.
4. Tekrarlayıcılar bit akışından bir saat işareti elde etmek zorundadır. Frekansın değiştiği ve özellikle ardarda gelmiş pek çok sıfırın bulunduğu durumda bunu elde etmek çok zor olacaktır.

Bu nedenlerden dolayı sayısal iletimde sinyaller iletilirken yukarıda bahsedilen unsurların ortadan kaldırılması için sayısal kodlama teknikleri kullanılır.

Sayısal işaret, ikilik bir verinin voltaj değişimleri kullanılarak kodlanmasıdır. Verinin iletimi sırasında kodlama tekniğine göre iki yada üç farklı voltaj düzeyi kullanılabilir: Pozitif yüksek voltaj düzeyi, negatif düşük voltaj düzeyi ve sıfır voltaj düzeyi. Kodlama sistemleri bu üç düzeyi bir arada kullanabilecekleri gibi bunlardan bir çiftini de kullanabilir. Hem yüksek hem de düşük voltaj düzeyinin bir arada kullanıldığı kodlama teknikleri ile oluşan işaretler, pozitif ve negatif değerler arasında gidip gelirler. Bu işaretlere iki kutuplu işaretler (bipolar) adı verilir. Yalnızca yüksek (ya da düşük) ve sıfır voltaj düzeyleri kullanarak yapılan kodlama sonucu oluşan işaretler ise tek bir düzeyde gidip gelir ve tek kutuplu (unipolar) işaretler olarak adlandırılır. İşaretlerin adlandırılmasını belirleyen bir diğer değişken ise kodlamanın sıfır düzeyini kullanıp kullanmamasıdır. Kodlamada sıfır düzeyi kullanılmıyorsa oluşan işaretler, hiçbir zaman sıfır noktasına sürekli değillerdir ve “**sıfıra dönmeyen (NRZ) işaretler**” olarak adlandırılırlar. Kodlamada sıfır düzeyi kullanılıyor ise oluşan işaret “**sıfıra dönen (RZ) işaret**” olarak adlandırılır.

Kullanılacak voltaj düzeylerini ve bir bitin 1 yada 0 olmasını belirleyen kurallar, kodlama tekniği tarafından belirlenir.

Günümüzde sıklıkla kullanılan kodlama teknikleri şunlardır:

1. Sıfıra Dönüşsüz (Non-Return to Zero – NRZ) Kodlama.
2. Sıfıra Dönüştü (Return to Zero – RZ) Kodlama.
3. İki Kutuplu (Bipolar – HDB3) Kodlama.
4. İki fazlı (BiPhase – Manchester) Kodlama.
5. Diferansiyel Manchester Kodlaması.
6. 4B/5B Kodlama
7. Çok Seviyeli Eşik – 3 (Multi-Level Threshold – MLT-3) Kodlama

4.2 Sıfıra Dönüşsüz (Non-Return to Zero – NRZ) Kodlama

Sıfıra dönüşsüz kodlama tekniğinde her bir bit belli bir düzeydeki işaret ile gösterilir. Örneğin ikilik 1 yüksek voltaj düzeyi ile, 0 ise düşük voltaj düzeyi ile gösterilir. İkili veri ardarda gönderilirken sürekli olarak düşük ve yüksek voltaj düzeyleri arasında geçiş olur, hiçbir zaman sıfır voltaj düzeyi kullanılmaz. Bir bitin gönderim aralığı boyunca voltaj düzeyi sabittir. Şekil 4.1’de, sıfıra dönüşsüz kodlama tekniği kullanılarak aktarılan biti bir veri gösterilmektedir.



Şekil 4.1 : Sıfıra Dönüşsüz (NRZ) Kodlama

Sıfıra dönüşsüz kodlama tekniği, hem eşzamanlı (senkron) hem de eşzamansız (asenkron) iletimde kullanılır. Genellikle düşük hızlı aktarımlar için uygundur. Kolay uygulanabilen bu tekniğin dezavantajı artarda aynı tür biti veri gönderildiğinde (sürekli ikilik 0 yada 1) voltaj düzeyinde hiçbir değişiklik olmamasıdır. Örneğin bir Ethernet yerel ağında binlerce ikilik 1 ardarda gönderilebilir. Bu durumda alıcı ve vericinin senkronizasyonu arasındaki en küçük bir kayma bile, gelen işarete bakılarak düzeltilemez.

4.3 Sıfıra Dönüştü (Return to Zero – RZ) Kodlama

Sıfıra dönen kodlama tekniğinde ikilik 1, voltaj düzeyinin yükselmesi ve daha sonra sıfıra geri dönmesiyle, 0 ise herhangi bir değişikliğin olmaması ile gösterilir. İkilik 1’in gösteriminde bit başlangıcında voltaj düzeyi artar, bitin gönderim aralığının yarısına kadar yüksek düzeyde kalır, tam yarısında 0 düzeyine düşer ve 0 düzeyinde kalır. Şekil 4.1’de verilen

bit dizisinin sıfıra dönen kodlama tekniği kullanılarak kodlanmış hali Şekil 4.2’de gösterilmektedir.

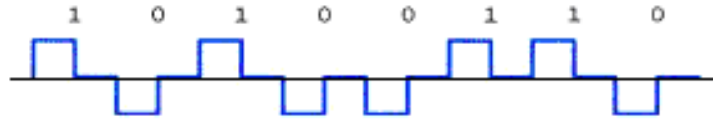


Şekil 4.2 : Sıfıra Dönüşlü (RZ) Kodlama

Sıfıra dönen kodlama tekniği, sıfıra dönmeyen kodlama tekniği ile karşılaştırıldığında önemli bir kazanım sağlamaz. Sıfıra dönmeyen kodlama tekniğinin aynı türden uzun bit dizilerinin gönderimi konusundaki dezavantajı, sıfıra dönen kodlama tekniğinde de vardır.

4.4 İki kutuplu (Bipolar – HDB3) Kodlama

İki kutuplu kodlama tekniği, her bir bitin yüksek yada düşük voltaj düzeyinden 0 düzeyine dönmesi ile kodlanır. İkilik 1, bitin başlangıcında voltaj düzeyinin yükselmesi, bit gönderim genişliğinin tam ortasına gelindiğinde ise tekrar sıfıra düşerek bitin sonuna kadar 0 düzeyinde kalması ile kodlanır. 0 ise aynı şekilde bitin gönderilmeye başlanması ile voltaj düzeyinin düşmesi, bit genişliğinin tam ortasında ise tekrar sıfıra yükselmesi ile kodlanır. Bu kodlama tekniği, Şekil 4.3’te gösterilmiştir. Her bir bitin kodlanmasında sıfır düzeyine gelinmesi, oluşan işaretin dönüş genişliğinin yarıya düşmesine neden olur.



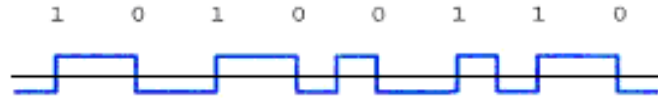
Şekil 4.3 : İki Kutuplu (Bipolar) Kodlama

İki kutuplu kodlamada her bir bitin tam ortasında sıfıra geri dönüş vardır. Bu sıfıra dönüş, senkron iletimde saat bilgisini taşımak için kullanılır.

4.5 İki Fazlı (Biphase – Manchester) Kodlama

Evre (phase) kodlaması olarak da bilenen Manchester kodlamasında bitler iki ayrı voltaj düzeyi arasında bir geçiş oluşturularak kodlanır. Bu nedenle sıfır düzeyine gerek yoktur, pozitif ve negatif genlik düzeyi yeterlidir. Manchester kodlamasında ikilik 1 gönderimi, düşük voltaj düzeyinden başlar ve bit gönderim genişliğinin tam ortasında voltaj düzeyi yükselir. Tam orta

noktada yukarı doğru olan bu yükselme (0→1) gönderilen verinin 1 olduğunu belirtir. bitin ikinci kısmında voltaj düzeyi yüksek kalır. Aynı şekilde 0 kodlanırken voltaj düzeyi yüksekten başlar, bitin tam ortasında aşağı doğru düşer (1→0) ve ikinci yarıda da düşük düzeyde devam eder. 1 ve 0 arasındaki geçişlerde voltaj düzeyi bir sonra gelen bite göre değişir. Örneğin biti 0 gönderildikten sonra gelen biti 1 ise düşük olan voltaj düzeyi 1'in başlangıç noktasını oluşturur, ancak gelen yine 0 ise (1→0) geçişinin sağlanabilmesi için voltaj düzeyinin yüksekten başlaması gerekir, dolayısıyla iki bit arasında bir geçiş olur. Şekil 4.4'te, Manchester kodlaması ile kodlanmış bir veri örneği gösterilmektedir.



Şekil 4.4 : Manchester Kodlama

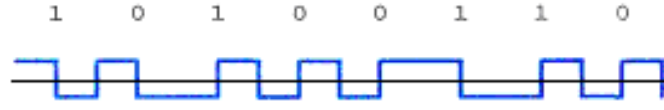
Manchester kodlamasında da biti kodlamada olduğu gibi bitin tam ortasında bir geçiş vardır. Ancak bu geçiş 0 düzeyine değil, yüksek düzeyden düşüğe yada tam tersinedir. Dolayısıyla Manchester kodlamasında eşzamanlı iletimde saat bilgisini taşımak için kullanılır.

4.6 Diferansiyel Manchester Kodlaması

Ayrımsal Manchester kodlamasında, Manchester kodlamasında olduğu gibi her bir bitin ortasında yüksekten düşüğe yada tam tersi yönde bir voltaj değişimi vardır. Ancak gönderilen verinin biti 1 yada 0 olduğunu belirleyen, bu değişimin pozitiften negatife yada negatiften pozitive olması değildir, bitin sonunda bir değişim olup olmaması, bir sonraki bitin ne olduğunu gösterir. Ancak bir sonra gelen bit 0 ise gönderilen bitin sonunda voltaj düzeyi değişir, yoksa aynı kalır. Ayrımsal Manchester kodlamasının bir örneği Şekil 3.16'da gösterilmektedir. Her bir bitin ortasında dönüşüm olduğuna ama ancak bir sonra gelen bit 0 ise sonunda bir dönüşüm gerçekleştiğine dikkat ediniz. Dolayısıyla biti 1 kodlandığında, bulunduğu konuma göre tam ortasında 0→1 yada 1→0 dönüşümünü gerçekleştirebilir, ancak bu verinin ne olduğunu belirlemez.

Ayrımsal Manchester kodlamasında bir veri dizisinin kodlanmış hali, işaretin düşük düzeyden yada yüksek düzeyden başlamasına göre değişir. Örneğin ilk bit 0 ise ve işaret düşük düzeyde ise, bitin başında bir dönüşüm gerekeceğinden yüksek düzeye çıkar. Ancak aynı veri için hattaki işaret yüksek düzeyde ise yine bir dönüşüm gerekeceğinden düşük düzeyden başlar. Oluşan bu iki form basitçe birbirinin tersidir.

Ayrımsal Manchester kodlamasında her bitin ortasında mutlaka bir dönüşüm olması, eşzamanlı iletimdeki saat bilgisinin kodlanması için kullanılır.



Şekil 4.5 : Diferansiyel Manchester Kodlama

4.7 4B/5B Kodlama

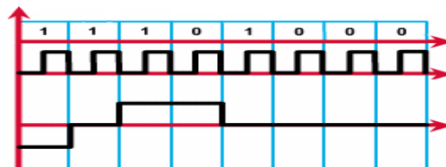
Bu kodlama tekniğinde 4 bitlik bilgiler 5 bit şeklinde kodlanır. Günümüzde daha çok FDDI ağlarında kullanılır. Aşağıda örnek olarak 4 bit ve 5 bitlik sembol kodlamaları görülmektedir.

4 bitlik grup	5 bitlik sembol grubu
0000	11110
0001	01001
0010	10100
0011	10101
0100	01010
0101	01011
0110	01110
0111	01111
1000	10010
1001	10011
1010	10110
1011	10111
1100	11010
1101	11011
1110	11100
1111	11101

Tablo 4.1 : 4B/5B Kodlama tablosu

4.8 Çok Seviyeli Eşik – 3 (Multi-Level Threshold – MLT-3) Kodlama

Üç seviyeli $(-V, 0, V)$ bir kodlama tekniğidir. Bu teknikte ikilik 0 için her hangi bir geçiş olmazken ikilik 1 için seviye döngüsü $(-V, 0, V, 0, -V, 0, V \dots\dots)$ içerisindeki bir sonraki seviyeye geçiş yapılır. Bu kodlama tekniği günümüzde özellikle Fast Ethernet (100Base-TX) Teknolojisinde kullanılmaktadır.



Şekil 4.6 : MLT-3 Kodlama

5 HATA KAYNAKLARI VE HATA DENETİMİ

5.1 Giriş

Bilginin verici ile alıcı arasında aktarılması, iletişim ortamları aracılığı ile gerçekleşir. Bilginin sağlıklı olarak aktarılabilmesi için, oluşabilecek hataların ve bozulmaların denetim altında tutulması gerekir. Haberleşmede hangi yöntem kullanılırsa kullanılsın, az yada çok, gönderilen bilginin hatalı olma olasılığı vardır. Bilgi, işaretler halinde iletişim ortamından aktarılırken, işaret, kuvvetinin azalması yada elektromanyetik etkileşim nedeni ile bozulabileceği gibi, alıcı tarafından alınırken senkronizasyon sorunu olması ve her bir biti temsil eden işaretlerin hatalı dönüştürülmesi sonucunda da bozulabilir. Kaynağı ne olursa olsun bu hataları saptamak ve düzeltmek için çeşitli teknikler kullanılmaktadır.

Bilgilerin iletimi esnasında bit hatalarının oluşma olasılığı çok fazladır. Bundan dolayı pek çok uygulama için, sadece hatanın olduğunu bulmakla yetinmeyip, ayrıca verinin hatasız bir başka kopyasının elde edilmesinin sağlanması gerekmektedir. Buna “**hata kontrolü**” denir.

Bir hata olması durumuna karşın, buna 4 farklı şekilde tepki verilebilir:

1. Hata için hiç bir şey yapmamak
2. Hatasız kopyayı yeniden göndermek
3. Hatayı fark edip aynı zamanda da hatasız kopyayı yeniden göndermek
4. Hatayı fark edip düzelterek göndermek (Hamming Kod).

Eğer veri de hata saptanmışsa verinin hatasız bir kopyasını elde etme gereği vardır.

5.2 Hata Kaynakları

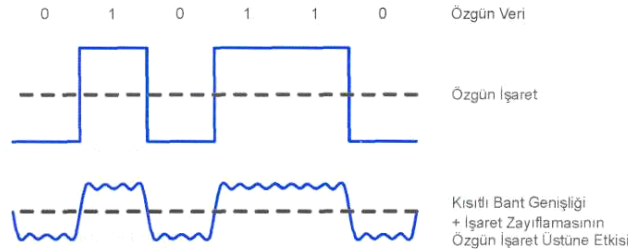
Herhangi bir iletişim sisteminde, verici tarafından gönderilen işaretler ile, alıcı tarafından alınan işaretler, iletişim ortamındaki çeşitli hata kaynaklarının neden olduğu bozulmalardan dolayı tam olarak aynı değildir. İşaretlerdeki bu değişimler, hem analog hem de sayısal işaretler için geçerli olmakla birlikte, özellikle sayısal işaretler için daha önemlidir. Çünkü analog bir işaret, iletişim ortamındaki hatalardan dolayı fazla değişime uğramasa da sayısal işaretler için bu değişimler bit hatası ve bit kayıplarına yol açar. Yani sayısal 1 olarak gönderilen bir işaret, sayısal 0 olarak (veya tam tersi) algılanır.

İletişim ortamlarından kaynaklanan en önemli hata kaynakları şunlardır:

1. İşaret zayıflaması ve bu zayıflamadan kaynaklanan bozulma,
2. Gecikmeden kaynaklanan bozulma,
3. İletişim ortamındaki gürültü,
4. İletişim ortamının kısıtlı bant genişliğinden kaynaklanan hatalar.

5.2.1 İşaret Zayıflaması (Attenuation)

Bir elektromanyetik işaret kablolu bir iletişim ortamından uzak mesafelere iletilirken enerji kaybı gerçekleşir ve iletilen işaretin gücü gittikçe zayıflar. Kablolu ortamlarda işaret gücü, mesafeyle birlikte logaritmik olarak düşer. Bu kayıp, her km için desibel cinsinden ölçülür. Kaybedilen enerji miktarı, işaretin frekansı ile ilişkilidir. İşaret gücündeki zayıflama (attenuation), frekansın bir işlevi olarak artar. Özellikle analog işaretlerde yüksek frekanslarda zayıflama daha fazla olur. Analog bir işaret, farklı frekans bileşenlerinden oluşur. Dolayısıyla her bileşenin bozulma ve zayıflama oranı farklı olacaktır. Bu da işaretle bozulmaya ve alıcı tarafından hatalı algılanmasına neden olur (Şekil 5.1). Bu durum, sayısal işaretlerde daha az rastlanan bir sorundur.

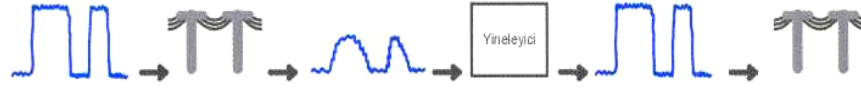


Şekil 5.1 : İşaret zayıflamasından kaynaklanan hata.

İşaretteki güç kaybı çok fazla olursa, alıcı işareti fark edemez yada işaret iletişim hattının gürültü düzeyinin altına düşer. Bu nedenle, alıcıya ulaşan işaretin, alıcıdaki elektronik devreler tarafından doğru olarak işlenebilmesi için, yeterli bir işaret kuvveti düzeyinde tutulması gerekir. Yeterli kuvvet düzeyi ise, işaret kuvvetinin, işaretin iletiildiği hattın gürültü düzeyinin belli bir miktar üstünde olması demektir.

İletişim ortamlarının özelliğine bağlı olarak işaret zayıflama oranları genellikle bilinir. Bu nedenle işaretin, kullanılan iletişim ortamının özelliğine bağlı olarak belli mesafelerde yükseltilmesi ve özgün durumuna yeniden getirilmesi gerekir. Bu amaçla, analog işaretler için yükseltici (amplifier), sayısal işaretler içinse yineleyici (repeater) kullanılır. Yineleyiciler,

kendilerine gelen sayısal işareti alır, yeniden oluşturur ve gönderir (Şekil 5.2). Her ne kadar işaretin zayıflamasından oluşabilecek bir hatayı önlemek için kullanılsalar da, yineleyici ve yükselticiler de hata kaynağı olabilir. Örneğin, yineleyiciye düşük bir işaret kuvvetiyle gelen sayısal işaret yineleyici tarafından hatalı oluşturulursa, bu aygıttan iletişim hattına giden işaret de hatalı olur ve düzeltilemez. Bir başka deyişle, yineleyicinin biti veriyi yeniden oluşturma işlemi sırasında yapılan herhangi bir bit hatası, iletişim boyunca devam eder.



Şekil 5.2 : Yineleyici kullanılarak işaret kuvvetinin yükseltilmesi.

İşaret kuvvetindeki zayıflama, bir iletişim ortamından çıkan özgün işaretle (P_1 watt) alıcıya ulaşan işaret kuvvetlerinin (P_2 watt) göreceli olarak birbirlerine oranları ile desibel cinsinden ölçülebilir.

$$\text{Zayıflama (Attenuation)} = 10 \log_{10} P_1/P_2 \text{ dB} \quad (5.1)$$

$$\text{Yükseltme (Amplification)} = 10 \log_{10} P_2/P_1 \text{ dB} \quad (5.2)$$

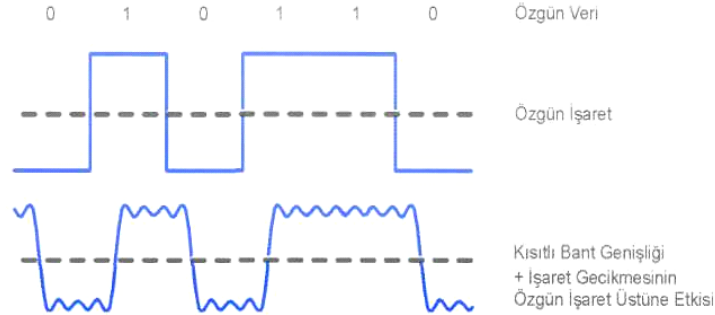
5.2.2 Gecikmeye Bağlı Bozulma (Delay)

İşaret gecikmesinden kaynaklanan hatalar “**gecikmeye bağlı bozulma (delay distortion)**” olarak adlandırılır. Bu durum kablolu iletişim ortamlarından kaynaklanan doğal bir sonuçtur. Bu nedenle de antenler aracılığıyla gerçekleştirilen kablosuz (wireless) iletişim ortamlarında gecikme oluşmaz.

Gecikmeye bağlı bozulma, bir işaretin kablo içerisindeki yayılma hızının, farklı frekanslarda farklı hızda olmasından kaynaklanır. Herhangi bir frekans bandındaki bir işaretin hızı, orta frekansta en yüksek, frekans bandının her iki tarafında ise en düşük olur. Böylece işaret bileşenleri, yani işaretin farklı Fourier bileşenleri farklı hızlarda yol alırlar ve alıcıya aynı zamanda ulaşamazlar. Böylelikle alıcıya ulaşan işaret, farklı frekans bileşenlerindeki farklı gecikmeler sonucunda değişime uğrar ve işaret bozulmasına neden olur.

Bu tür bozulmalar özellikle sayısal veriler için çok daha önemlidir. Gönderilen bit hızı arttıkça, bozulmanın miktarı da artar. Bu durum şöyle açıklanabilir: Gönderilen bit oranı arttıkça, her bitin gönderimi sırasında biti oluşturan frekans bileşenleri gecikecek ve daha sonraki bitin frekans bileşenleriyle karışacaktır. Alıcı tarafta ise bit, her bit periyodunun ortasında

örneklendiğinden, bir bit konumunun, diğer bit konumuna kaymasına, bu da alınan bitin yanlış örneklenmesine ve dolayısıyla hataya neden olur.



Şekil 5.3 : Gecikmeye bağlı bozulma.

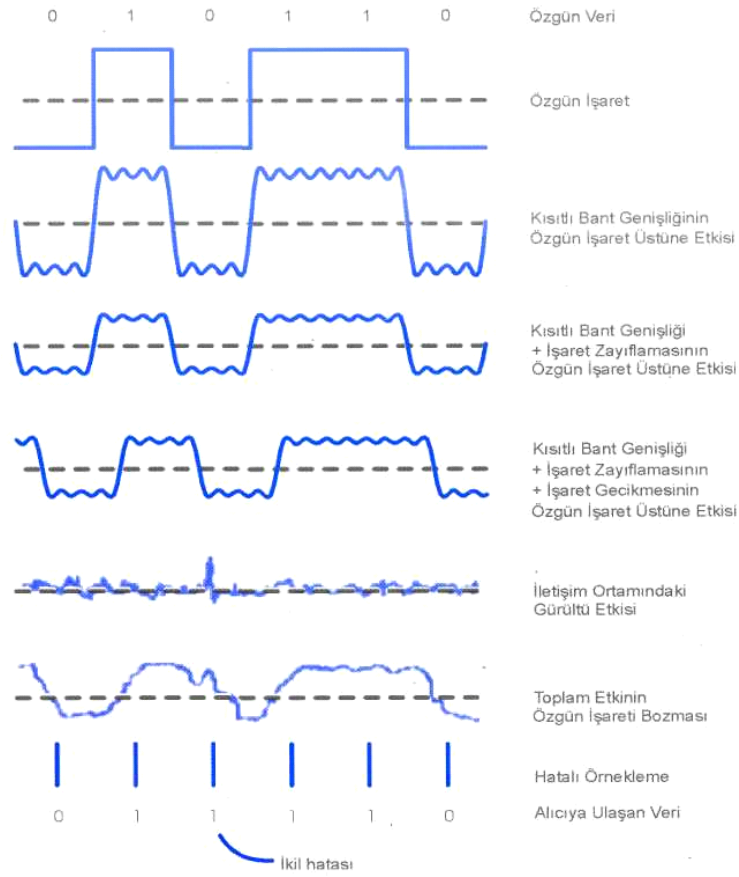
5.2.3 Gürültü (Noise)

Alıcı tarafından alınan bir işaret, özgün işarete göre, işaret gücünün zayıflaması ve iletişim sisteminden kaynaklanan çeşitli kayma ve bozulmaların yanı sıra, dış etkilerden kaynaklanan ve istenmeyen ani elektromanyetik enerji etkileşimleriyle de değişikliğe uğrar. Alıcı ile verici arasındaki iletişim sırasında ortaya çıkan bu istenmeyen elektromanyetik enerji, “gürültü” olarak adlandırılır (Şekil 5.4).



Şekil 5.4 : İletişim ortamındaki gürültü.

Bir iletişim ortamında işaret iletilmediği durumlarda bile, hat üzerinde rasgele elektromanyetik etkileşimler bulunur. Hattın gürültü düzeyi olarak bilinen bu etkileşimle, zayıflama yada gecikme gibi diğer etkenlerin birleşmesi, gönderilen veride bir yada daha fazla bitlik bozulmalara neden olabilir. Örneğin Şekil 5.5, bir iletişim ortamı üzerinden gönderilen verinin gürültü, zayıflama ve gecikme etmenlerinin toplam etkisi sonucunda nasıl bozulduğunu göstermektedir.



Şekil 5.5 : Hata kaynaklarının toplam etkisi.

Her iletişim hattının belirli bir gürültü düzeyi vardır. Gürültü ve nedenleri kendi içinde 4 ana başlıkta ele alınabilir:

1. Termal sıcaklığa ait gürültü (Thermal Noise).
2. Modülasyonlar arası gürültü (Intermodulation Noise).
3. Çapraz konuşma (Crosstalk).
4. Uyarım gürültüsü (Impulse Noise).

5.2.3.1 Termal Gürültü

Bu gürültü türleri arasında termal gürültü, dış etkilerden bağımsızdır. Termal gürültü, bir iletkenin veya kablunun içindeki elektronların rasgele hareketinden kaynaklanır, bu nedenle de kaçınılmazdır. Bu gürültü türü, tüm elektrikli aygıtlarda ve iletişim ortamlarında bulunup sıcaklığın bir fonksiyonudur. Diğer bir deyişle, 0° 'nin üstündeki tüm sıcaklıklarda tüm iletişim ortamları termal gürültü ile karşı karşıyadırlar. Termal gürültü, işaretin sıklık aralığı boyunca aynı şekilde ve dümdüz dağılır. Bu nedenle de beyaz gürültü (white noise) olarak da adlandırılır.

5.2.3.2 Modülasyonlararası Gürültü

Farklı frekans sahip işaretler aynı iletişim ortamını paylaştığında, örneğin FDM yönteminin kullanıldığı geniş bantlı iletişim ortamlarında, modülasyonlararası gürültü ortaya çıkabilir. Modülasyonlararası gürültü genelde birbirine karışan iki farklı işaretin frekanslarının toplamı yada farkı biçiminde ortaya çıkar.

5.2.3.3 Uyarım Gürültüsü

Buraya kadar anlattığımız gürültü türleri, sabit ve tahmin edilebilir büyüklüktedir. Bu nedenle de bir iletişim sisteminin bu tür gürültülerin üstesinden gelebilmek için önlem alabilmesi olasıdır. Uyarım (impulse) gürültüsü ise, sürekli olmayan ve kısa aralıklardaki güçlü işaretlerin özgün işareti etkilemesiyle ortaya çıkar. Bu tür gürültünün çeşitli nedenleri vardır. Bunlar arasında, iletişim sistemindeki dış elektromanyetik dağılımlar yer alır. Uyarım gürültüsünden kaynaklanan hatalar, sayısal bir iletişim ortamının en önemli hata kaynaklarından biridir.

Örneğin, 0.01 saniye süren ani ve keskin bir elektromanyetik enerji etkileşimi, herhangi bir ses iletişimini pek de fazla değiştirmezken, 4800 bps hızda gönderilen bir verinin yaklaşık 50 bitini silebilir. Bu nedenle, bu tür hatalar önlem alınabilmesi en zor olan ve bilgi kaybına yol açan hata kaynaklarıdır.

5.2.4 Çapraz Konuşma (Crosstalk)

Çapraz konuşma, birbirine yakın iki iletken veya anten arasındaki istenmeyen elektromanyetik etkileşim/karışma sonucunda ortaya çıkan gürültüdür.

Birkaç tür çapraz konuşma vardır. Bunlardan NEXT, oldukça yaygın olarak görülen bir hata biçimidir. NEXT, bir göndericiye girdi olarak zayıf bir işaret girerken, aynı göndericiden çıktı olarak güçlü bir işaretin iletilmesi durumunda ortaya çıkan işaret etkileşiminden kaynaklanır. Bu tür hataları düzeltmek için, adaptif NEXT önleyici olarak bilinen aygıtlar kullanılır.

5.3 Hata Kontrolü

Bir önceki bölümde anlatıldığı gibi, iletişim ortamları hata potansiyeline sahiptir. Bundan dolayı da bir veri iletişim sisteminin en önemli görevlerinden biri, bu hataları denetleme ve düzeltebilme yeteneğidir.

Bir iletişim sisteminde hata saptama ve düzeltme teknikleri, sistem başarımını büyük ölçüde etkiler. Uygulamaların hiçbiri hataya karşı toleranslı değildir. Bu nedenle de birçok çok uygulama için, hatayı saptamakla yetinmeyip, verinin hatasız bir başka kopyasının elde edilmesinin de sağlanması gerekmektedir. Buna, veri iletişim terminolojisinde “**hata denetimi (kontrolü)**” denir.

Hata denetim süreçleri iki aşamada gerçekleştirilir:

1. Hatanın saptanması.
2. Hatanın düzeltilmesi.

5.3.1 Hata Bulma Yöntemleri

Bir haberleşme sistemi üzerinden iletilen bilginin, alıcı tarafından hatasız olarak alındığından veya alınan bilgide hata olmadığından emin olabilmek için çeşitli yöntemler kullanılır. Veride hata saptanmışsa, verinin hatasız yeni bir kopyasını elde etme gereği duyulur.

Hata saptama yöntemlerinde kullanılan iki yaklaşım vardır:

1. İleri hata denetimi (forward error control): Bu yaklaşımda, gönderilen her karakter veya çerçeve bilgisinin yanı sıra, fazla ve yinelenen bilgi içeren bitler gönderilir. Öyle ki, alıcı hatayı fark etmekle kalmaz, hatanın hangi bit veya bitlerde olduğunu da saptar (çerçeve ile birlikte gönderilen ek bilgi bitleri bunu saptayacak bilgiler içerir). Daha sonra hatasız veri, ek bilgi bitlerinden elde edilir. Güvenilir bir ileri hata denetimine erişebilmek için gereken ek bit sayısı, gönderilen bilgideki bit sayısı arttıkça hızla artar.
2. Geri beslemeli hata denetimi (feedback error control): Bu yaklaşımda, gönderilen her karakter veya çerçeve, alıcının yalnızca veride hata bulunduğunu saptamasına yetecek kadar ek bilgi biti içerir. Alıcı, bu bilgi bitlerini kullanarak veride hata olup olmadığını denetler. Veride hata saptandığında ise, ek bilgi bitleri hatanın düzeltilmesine olanak tanımayacağından, alıcı, gönderilmiş olan hatalı verinin doğru bir kopyasının gönderilmesi için istekte bulunur. Bu nedenle geri beslemeli hata denetimi iki aşamada ele alınır:
 - a. Güvenli bir şekilde hatayı saptamayı sağlayan teknikler.
 - b. Hata durumunda, yeniden gönderim denetim tekniklerini sağlamak için kullanılan algoritmalar.

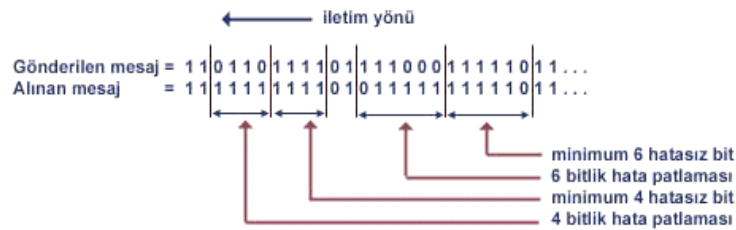
5.3.2 Bit Hata Oranı (Bit Error Rate – BER) ve Hata Patlaması

Hata saptama türlerini belirleyen etkenlerden biri BER, diğeri ise Hata Patlaması (Error Burst)’ kavramlarıdır.

BER (Bit Error Rate), bir iletişim sistemi üzerinden iletilen tek bir bitin belli bir zaman aralığındaki bozulma olasılığıdır. Örneğin, BER 10^{-5} demek 100.000 bit arasından 1 bitin belli bir zaman aralığındaki bozulmasıdır.

Hata Patlaması, yıldırım ve benzeri doğal olayların, çevre koşullarının veya iletişim ortamındaki şiddetli bir elektromanyetik etkileşimin, olay süresince iletişim ortamından geçen bitleri etkileyip bozmasıyla oluşan hatalardır. Hata patlaması, yanlış alınmış bir bitle başlar, bir süre doğru veya yanlış bit dizisi ile devam eder ve hatalı bir bit ile biter. Yani, iki hatalı bit arasındaki bit sayısıdır. Aradaki bitler bozulmuş veya bozulmamış olabilir.

Diğeri bir deyişle hata patlaması, birbirini izleyen hatalı 2 bit arasındaki bitlerin sayısı olarak tanımlanır. Bu sayıya hatalı 2 bit de dahildir. Ayrıca bir hata patlamasının uzunluğunu belirlerken patlamadaki son hatalı bit (1. hata patlamasının son biti) ve onu izleyen ilk patlamadaki (2. hata patlamasının başı) ilk hatalı bit ve N de 1. hata, hata patlamasının uzunluğu olmak üzere N veya daha fazla doğru bitle ayrılmalıdır. Bir başka deyişle, bit dizisi içerisinde hata patlamasının son bitinden sonra, en az hata patlamasındaki bit sayısı kadar doğru bit gelmelidir. Bunu bir örnekle açıklayacak olursak, Şekil 5.6’da iki farklı hata patlamasının bir örneği gösterilmektedir.



Şekil 5.6 : Hata Patlaması Örneği

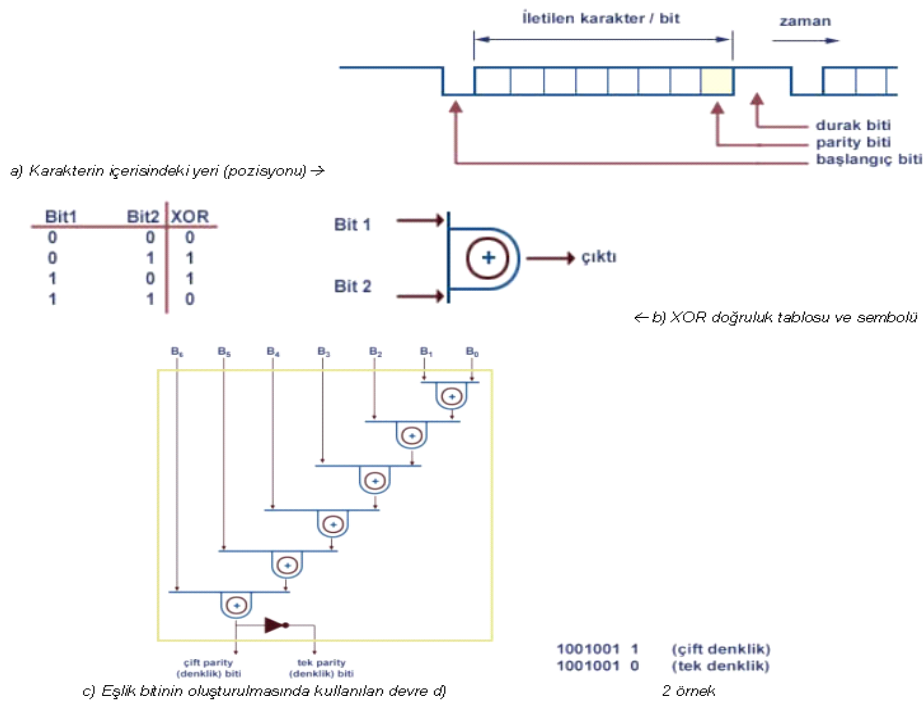
Hata türü önemlidir, çünkü farklı hata türleri için farklı hata bulma yöntemleri kullanılır. Ayrıca, bazı yöntemlerde kullanılan bit sayısı, fark edilecek patlama uzunluğunu tanımlar. En fazla kullanılan yöntemlerden bazıları, Eşlik Biti Denetimi (Parity Check) ve Dönüşümlü Artıklık Denetimi (Cyclic Redundancy Check – CRC)’dir.

5.3.3 Eşlik Biti Denetimi (Parity Check)

Eşzamansız ve karakter yönelimli eşzamanlı gönderimde, eşlik biti en sık kullanılan hata saptama yöntemlerinden birisidir.

Eşlik biti, iletim sırasında her karaktere (byte), bu bitlerde herhangi bir değişiklik olup olmadığını denetlemek amacıyla eklenen tek bir bittir. Eşlik, çift veya tek olarak belirlenebilir. Çift eşlik, eşlik biti de içinde olmak üzere bir karakterin içerisindeki 1 bitlerinin sayısının çift olması durumudur. Tek eşlik ise, yine eşlik biti de içinde olmak üzere, karakterdeki 1'lerin sayısının tek olması durumudur.

Karakteri yollayacak olan bilgisayar tek eşlikte ise karakterdeki 1'lerin sayısının tek, çift eşlikte ise karakterdeki 1'lerin sayısının çift olmasını sağlamak üzere, karakterin sonuna 0 veya 1 eşliğini ekler. Karakteri alan bilgisayar, kullanılan eşliğe göre, 1'lerin sayısını denetler. Teklik veya çiftlik sayısı tutmazsa, karakterin iletişimde bir hata olduğu anlaşılır ve göndericinin karakteri yeniden göndermesi istenir.



Şekil 5.7 : Eşlik Biti Denetim (XOR'lama) işlemi

Bir karakter için eşlik biti hesaplanırken iki bit mod-2 aritmetiğine göre toplanır (Şekil 5.7). Bu hesaplama, bir dizi exclusive-OR (XOR) anahtarından oluşur.

5.3.4 İki-Boyutlu Eşlik Biti Denetimi (2-Dimensional Parity)

İki boyutlu eşlik, bir çerçevenin içerdiği karakterlerin her birine karşı gelen her bit konumu için benzer bir hesaplama yapar. Bu işlem, her karakter için bir eşlik bitine ek olarak, fazladan bir eşlik sekizinin oluşmasına neden olur?

Örnek:

```
0 1 0 0 1 0 1 0
1 0 0 0 1 1 0 0
0 0 1 0 0 0 1 1
1 0 1 1 0 1 0 1
0 0 0 1 0 1 1 0
0 1 0 0 1 0 1 0
1 1 0 1 0 1 0 1
0 0 1 0 0 1 1 0
```

5.3.5 Dönüşümlü Artıklık Denetimi (Cyclic Redundancy Check – CRC)

Eşlik biti hata patlamalarına karşı güvenilir bir hata saptama yöntemi sağlayamaz. Bu gibi durumlarda alternatif olarak polinom kodlar kullanılır.

Polinom kodlar çerçeve veya blok iletim yöntemleriyle kullanılırlar. Bu yöntemde, gönderici sistem tarafından çerçevenin içeriğine göre ve çerçevenin sonuna eklenecek şekilde her çerçeve için biti tek bir kontrol seti (check digits) oluşturulur (hesaplanır). Daha sonra alıcı sistem, tamamlanmış çerçeve ve kontrol sayıları üzerinde benzer bir hesaplama yapar. Eğer hiç hata yapılmamışsa bilinen bir sonuç elde edilmelidir. Eğer farklı bir sonuç bulunursa bu bir hata olduğunu gösterir. Her çerçeve için kontrol sayılarının sayısı beklenen iletim hatalarının tipine uyacak şekilde seçilir. Ancak 16 ve 32 bit kontrol sayıları en sık kullanılanlarıdır. İşlenen kontrol sayılarına çerçeve kontrol dizisi (frame check sequence) FCS veya CRC, sayıları denir.

Polinom kodlarının altında yatan metod, eğer mod-2 aritmetiği kullanılırsa 2'li düzendeki sayıların aşağıdaki özelliğini kullanır. Diyelimki,

$M(x)$: gönderilecek k bit sayılı mesaj olsun.

$G(x)$: $(n+1)$ bitli sayı (bölen)

$R(x)$: n bitli sayı ($k > n$) olsun, bu durumda aşağıdaki eşitliği yazılabilir.

$$(M(x) \cdot 2^n + R(x)) / G(x) = Q(x) + (R(x) / G(x)) \quad (5.2)$$

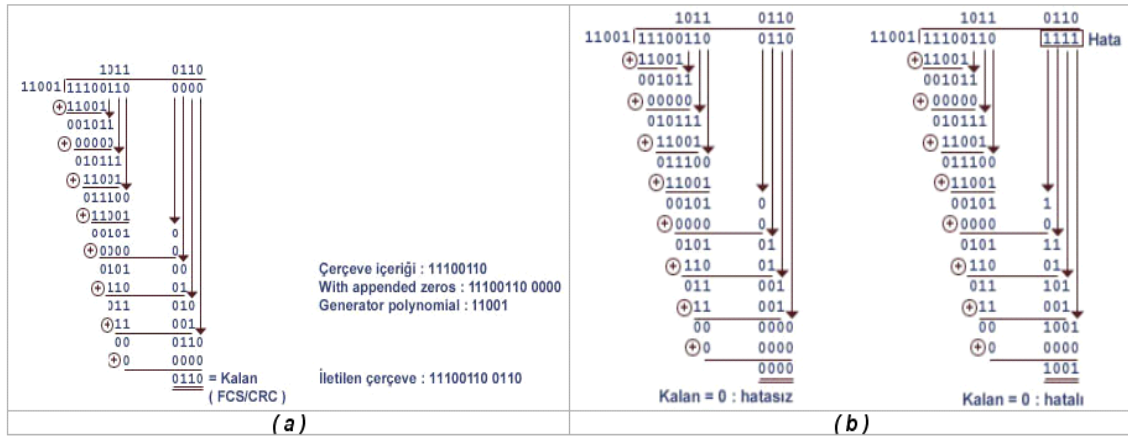
Bu da $Q(x)$ 'e eşittir. Çünkü ikilik düzende bir sayının kendisiyle toplamı 0 eder. Bu da kalanın 0 olacağını gösterir.

Bunu kullanmak için çerçeve içeriklerinin tamamı, $M(x)$, sayıları oluşturulacak FCS digitlerinin sayısına eşit olarak eklenmiş bir 0 lar setiyle (n FCS digitlerinin sayısına eşitken, mesajı 2^n ile çarpmaya eşdeğerdir) FCS den 1 digit daha fazla digit içeren ikinci bir ikilik sayı ($G(x)$, oluşturucu polinom (generator polinomial)) tarafından mod-2 ye bölünür.

Bölünme işlemi, XOR işlemini çerçevedeki her bit işlenirken bitten bite paralel olarak gerçekleştirmeye eşdeğerdir. Daha sonra artan $R(x)$ FCS'dir ve mesaj bitlerinin sonuna eklenir. Benzer şekilde alıcı tarafından FCS digitlerini de içeren bit akımı tekrar aynı polinom tarafından bölünür yani $(M(x).2^n + R(x))/G(x)$. Eğer hiç hata yoksa artan sadece 0'dır. Ancak bir hata varsa artan 0 değildir.

Bu işlem bir örnekle açıklanacak olursa, 8 bitlik bir mesaj bloğu 11100110 ve 11001 oluşturucu polinomunu kullanarak CRC yöntemi ile hata saptadığımızı varsayalım.

FCS'yi önce oluşturalım ve sonra da FCS'yi kontrol edelim.



Şekil 5.8 : FCS oluşturma ve kontrolü

Şekil 5.10'da görüldüğü gibi mesaja önce 4 sıfır eklendi çünkü FCS 4 bitli olacaktır (oluşturucu polinomun digit sayısı $n=5$). Bu sayı daha sonra şekilde görüldüğü gibi oluşturucu polinoma bölünür. Yukarıda da belirtildiği gibi mod-2 bölme işlemi bit bit paralel bir şekilde XOR işlemine eşdeğerdir. Bu işlemden artan sayı FCS sayısıdır. Alıcı, mesaja eklenmiş FCS sayısını alır ve oluşturucu polinoma böler (Şekil 5.10-b). Eğer hata yoksa 0 sayısı elde edilir. Bu teknikte oluşturucu polinomun seçimi çok önemlidir çünkü polinom hangi tip hataların fark edileceğini tanımlar.

Yani bu yöntemle,

1. Tüm tek-bit hatalarını
2. Tüm çift-bit hatalarını
3. Tüm tek-sayıli bit hatalarını
4. R'den küçük tüm hata patlamalarını
5. R'den büyük çok fazla hata patlaması fark edilebilir.

5.3.6 Sağlama-Toplamı (Checksum) Algoritması

Bu yöntem, CRC ve eşlik biti ile aynı işlevi sağlar. CRC yöntemi kadar kesin koruma yöntemi sağlayamaz, fakat OSI başvuru modelinin ağ katmanı için güvenle kullanılabilir. Çünkü veri bağlantı katmanında hata bulma teknikleri (genelde CRC) zaten uygulanmaktadır.

Sağlam toplamı algoritmasının temeli oldukça basittir. Şöyle ki;

1. Gönderdiğin tüm kelimeleri topla
2. Elde ettiğın toplamı sağlama toplama olarak gönder

Alıcı da aynı işlemleri tekrar ederek, bulduğu sağlama toplamını gönderilenle karşılaştırır. Bir hata durumunda sonuçlar farklı çıkacağından, alıcıyı iletimde bir hata olduğu konusunda uyarır.

5.3.7 Hata Düzeltme Yöntemleri

Haberleşme sistemlerinde hata düzeltme teknikleri iki yaklaşımla gerçekleştirilir.

1. Hatanın saptanıp, iletinin yeniden gönderilmesinin sağlanması.
2. Hatanın alıcıda düzeltilmesinin sağlanması.

Bunlardan ilki, alıcı tarafında hatanın saptanmasını izleyen ve hatalı çerçevelerin yeniden gönderilmesini sağlayan Otomatik Tekrar istemi (ARQ) aracılığıyla gerçekleştirilir.

İkinci seçenek ise hatanın alıcıda düzeltilmesinin sağlanmasıdır. Göndericinin bir hata durumunda, aynı çerçeveyi yeniden göndermesinin çok güç olduğu, iletişim ortamlarının çok pahalı olduğu veya bir çerçevenin yeniden iletilmesinin çok fazla zaman kaybına yol açtığı

durumlarda, hatanın alıcıda düzeltilebilmesini sağlayan çeşitli teknikler geliştirilmiştir. Bu gibi durumlarda, çerçevedeki hata çok büyük bir hata patlaması değilse, alıcıda düzeltilmeye çalışılır. Bu teknikler arasında en sık kullanılan yöntem, Hamming Kodlamasıdır.

5.3.7.1 Hamming Uzaklığı ve Hamming Kodlaması

Bir iletişim ortamı üzerinden iletilen bit dizisinin en fazla k bitinin bozulması durumunda, bit dizisi içerisindeki hata düzeltilebiliyorsa, bu koda “**k_ikil hata bağışıklığı**” denir.

Kodlama terminolojisinde veri bitleri ile veriye eklenen hata saptama bitlerini birleştiren birleşik ileti birimi “**kod sözcük (codeword)**” olarak adlandırılır. Örneğin, m bitten oluşmuş bir veri, r bitlik hata denetim veya tekrar bitleri ile ($n=m+r$) bitlik bir kod sözcük oluşturur. İki geçerli kod sözcüğün farklılaştığı en düşük bit konumu sayısı, kodun Hamming uzaklığı olarak bilinir. 2 kod sözcük arasında kaç iki! konumunun değiştiğini bulmak için, önce iki kod sözcüğe XOR işlemi uygulanır ve sonra bu işlem sonucunda elde edilen 1’lerin sayısı toplanır.

Örneğin, her karakterin 7 veri biti ve bir eşlik biti ile gösterildiği bir kod sözcük düşünelim. Çift eşlik kullanıldığını varsayarsak bu şemayı izleyen kod sözcükler:

0000000	0
0000001	1
0000010	1
0000011	0

Buradan Hamming uzaklığının 2 olduğunu çıkarabiliriz. Çünkü geçerli her kod sözcük en az iki bit konumunda farklıdır. Bu da, bu şemanın 2 bitli hataları saptayamayacağını gösterir. Çünkü sonuçtaki bit yapısı farklı, ancak geçerli bir kod sözcük olacaktır. Bununla birlikte bütün tek bitli hataları saptayabilir. Çünkü bir kod sözcükteki tek bir bit bozulursa, geçersiz bir kod sözcük ortaya çıkacaktır. Örneğin, Hamming uzaklığı 3 olan bir kod sözcük tek bitli hataları düzeltebilir, 2 bitli hataları bulabilir.

Eğer iki kod sözcük arasındaki Hamming uzaklığı u ise, bir kod sözcüğü diğerine çevirmek için, u , tek bit hatası gerekecektir. u hatayı saptamak için, $u + 1$ koda, u hatayı düzeltmek için ise $2u + 1$ koda gerek duyulur.

Hamming kodu şu şekilde oluşturulur;

1. Kod sözcükteki bitler 1'den başlayarak sırayla numaralanır.
2. Bit sıra numarası 2^n 'nin kuvveti olan bitler (1, 2, 4, 8, 16, ...) denetim bitleri,
3. Diğer sıradaki bitler (3, 5, 6, 7, 9, ...) veri bitleriyle doldurulur.
4. Her denetim biti, kendisi de içinde bazı bit setlerinin eşliği olarak hesaplanır. Bir bit, birkaç eşlik biti hesaplamasına dahil olabilir.
5. k konumundaki (sirasındaki) veri bitlerinin, denetim bitlerinin hangisi olduğunu görebilmek için k , 2^n 'nin kuvvetlerinin toplamı şeklinde yazılır.

Örneğin, $7 = 1+2+4$ (7. bit, 1., 2. ve 4. bitler tarafından denetlenir), $11 = 1+2+8$ (11. bit, 1., 2., ve 8. bitler tarafından denetlenir).

6. Bir kod sözcük geldiğinde, alıcı her denetim bitinin doğru eşlikte olup olmadığını denetler. Tüm denetim bitlerinin paritesi (eşliği) doğru ise kod sözcük geçerli olarak kabul edilir.

Örnek:

```
100100000110010000
110000110111001001
110110111101010101
110110111101010101
110100101101011001
110111001101010110
```

Hamming kodu, aslında tek bit hatalarını düzeltebilir. Fakat bu yöntem, bir kod sözcük matrisi oluşturularak, hata patlamalarında da kullanılabilir.

6 VERİ SIKIŞTIRMA

6.1 Giriş

Saklamak yada iletmek amacıyla sinyallerin sayısallaştırılmasının iki ana sebebi, hatalara karşı dayanıklılık katabilmek ve sıkıştırıp artık verilerden arındırabilmektir. Enformatik açıdan ise sinyalleri kaynaklarından ve izleyicilerinden bağımsız düşünemeyiz. Oysa enformasyon, günümüzde, bu öznel çabasını yitirmiştir. Bilgiyi atomize edip küçük mesajlara dönüştürdüysek suçu, “binary information unit - bit”leri tanımlayıp, ilk defa kullandığı için Bay Shannon’a atamayız. Enformasyon matematiksel niceliklerle ifade edildiği sürece anlamsal içerik kaybolacaktır. Günümüzde bilgi teknolojileri bilgiden ziyade sinyaller, veriler gibi fiziksel olgularla ilgilidir. Doğa ile bütünleşik sistemler kurmak ve olası verimin tümünü elde edebilmek için, bilgiye öznel varlığını geri vermemiz gerekiyor. Şimdiye kadar üretilen sıkıştırma araçlarını gözden çıkarmak istemediğimize göre, onları anlamsal çözümleme yeteneğine sahip sistemlerde kullanmak çıkar bir yoldur. Enformasyonu henüz cihazlar tarafından edinme aşamasında sinyaller yerine anlamsal bütünlüğü olan sembollerle ifade etmeliyiz. Bu semboller birer ihbar, birer gösterge kabul edilip hem tahmin hem tahmin hatası kodlama aşamalarında doğru araçların seçilebilmesini sağlamalıdır. [20]

Tipik bir mesaj, istatistiksel olarak incelenirse belirli karakterlerin diğerlerinden daha fazla kullanıldığı gözlemlenebilir. Bir mesaj iletilemeden önce analiz edilerek, sıkça kullanılan karakterlerin yerine kısa kodlar atanır ve uzun kodlar yalnızca seyrek karşılaşılan karakterlerde kullanılır. Bunu yapmakla mesajdaki bilgiyi değiştirmeden gönderilecek toplam karakter sayısını azaltmak mümkündür. Alıcıdaki uygun çözümleyici mesajı asıl şekline geri döndürecektir. Bu işlem “**veri sıkıştırma**” olarak bilinir. Mesaj iletmeden önce analiz edilmesi için zamana ihtiyaç duyulmasında rağmen, sağlanan tasarruf sayesinde sıkıştırma, iletim ve sıkıştırılan veriyi açma için gerekli olan toplam zaman sıkıştırılmamış mesajı iletmekten daha kısa olabilir.

Bazı tür veriler diğerlerinden daha fazla sıkışırlar. Örneğin resimleri oluşturan veriler asıl boyutunun %20’sine düşürebilirler. Ancak bir bilgisayar programı ancak %20 veya %25 sıkıştırılabilir. Ortalama oranda bir sıkıştırıcı ile bu dosyayı sıkıştırdığımızda yaklaşık %50 oranda bellek kazancı olur. Bu gönderme süremizi %50 oranda düşürmemiz demektir. Tüm internet hareketlerini düşünersek binlerce terabyte seviyede kazanç sağlanmaktadır.

Özetle bilgisayarda veri sıkıştırma işleminin uygulanmasının temel nedeni, sıkıştırılmış verinin daha az yer kaplamasıdır.

Bir bilginin sayısal ortamda daha verimli bir şekilde depolanabilmesi ve iletilebilmesi için veri sıkıştırma teknikleri kullanılır. Bazen, uygulama programları verinin iletildiği ağın bant genişliğinin taşıyabileceğinden daha fazla veri üretir. Örneğin bir video uygulaması 10 Mbps veri akımı üretirken, iletişim ağı yalnızca 1 Mbps veriyi taşımaya elverişli olabilir. Veri sıkıştırma tekniklerinin amacı, verinin içindeki fazlalıkları atarak hacminin azaltılmasıdır. Verinin türü ne olursa olsun (metin, resim, video vb) içinde fazla öğeler barındırır. Bu fazlalıklar, hem verinin depolanması sırasında daha fazla disk alanı gerektirir, hem de aktarılması için gereken süreyi ve maliyeti arttırır.

Örneğin telefon hatları üzerinden çevirmeli ağ ile Internet'e bağlanan bir kullanıcıyı düşünelim. Herhangi bir dosyayı bilgisayarına indirmek için gereken süreyi azaltabilirse, arama maliyetini de aynı oranda azaltacaktır. PSTN üzerinden veri 4800 bps'de aktarıldığında dosyayı indirmek için gereken zamanın 20 dakika olduğunu varsayalım. Veri sıkıştırma yöntemlerini kullanarak, dosyanın büyüklüğünü yarıya indirebilirsek, iletişim maliyetinde % 50 tasarruf sağlanabileceği açıktır. Bu durum, veri sıkıştırması uygulanmış bir dosyayı 4800 bps'lik bir modem ile indirmek, veri sıkıştırması uygulanmamış dosyayı 9600 bps'lik bir modem kullanarak indirmeye eşdeğerdir.

Veri sıkıştırma tekniklerinde, kodlama (encoding) ve kod çözme (decoding) algoritmaları kullanılır. Veri sıkıştırılırken, atılan fazlalıkları temsil edecek bir kodlama kullanılmalıdır. Alıcı bu kodlamayı bilmeli, verinin iletilmesinden sonra kodu çözerek özgün veriyi tekrar oluşturabilmelidir.

Farklı veri sıkıştırma teknikleri vardır. Bunlar iki ana grupta toplanabilir:

1. Entropi (entropy) Kodlaması. (Kayıpsız Sıkıştırma)
2. Kaynak (source) Kodlaması. (Kayıplı Sıkıştırma)

Bu iki ana grup ve bu grupta yer alan temel sıkıştırma teknikleri aşağıda incelenmiştir.

6.2 Entropi Kodlaması (Kayıpsız Sıkıştırma)

Entropi kodlaması, verinin içeriği ile ilgilenmez. Yalnızca, veriyi oluşturan bitleri bir kodlama algoritması aracılığı ile yeniden kodlar. Bu nedenle, entropi kodlamasında bilgi kaybı olmaz, sıkıştırılmış veriyi alan bilgisayar kod çözme algoritmaları ile verinin özgün halini tıpa tıpa elde edebilir. Bu nedenle bu tip sıkıştırmaya “**kayıpsız sıkıştırma**” adı verilir.

Bir verinin entropisi (H), verinin içerdiği her sembol için kullanılması gereken en az bit sayısıdır. Entropi değeri, bilgi kaynağının ürettiği verinin çeşidine ve tekrarlanma olasılığının dağılımına göre değişir. Örneğin bir bilgi kaynağı yalnızca Y harfi üretip gönderiyorsa bunun entropi değeri sıfırdır. Çünkü o veri kaynağının başka bir veri gönderme olasılığı yoktur. Ancak birden çok çeşitte veri gönderen bir veri kaynağının entropisi de artar. Bir sembolün tekrarlanma oranı arttıkça entropi değeri de düşer. Entropi, aşağıdaki formül kullanılarak hesaplanır:

$$H = \sum_{m=1}^M P(m) \log_2 \frac{1}{P(m)} \text{ (bit / karakter)} \quad (6.1)$$

En fazla entropi ise (H), verideki tüm sembollerin tekrarlanma olasılığının eşit olduğu durumdur ve aşağıdaki formül kullanılarak hesaplanır:

$$H_{\max} = \log_2 M \text{ (bit / karakter)} \quad (6.2)$$

Fazlalık, kaynağın (verinin) olası olan en fazla entropisi (H_{\max}) ile gerçek entropisi (H) arasındaki farktır.

Veri, sıkıştırma teknikleri kullanılarak başka bir sembol seti ile tanımlanmış ise yeni kod sisteminin, yani sıkıştırmanın verimliliği aşağıdaki formülle hesaplanır:

$$\prod_{kod} = \frac{H}{H_{\max}} \times \% 100 \quad (6.3)$$

Entropi kodlamasına iki ayrı kodlama tekniği, “**Tekrarlama Uzunluğu Kodlaması (Run-length Encoding)**” ve “**İstatistiksel Kodlama (Statistical Encoding)**” örnek olarak verilebilir.

6.2.1 Tekrarlama Uzunluğu Kodlaması (Run-Lenght Encoding – RLE)

En basit veri sıkıştırma yöntemi olarak RLE (Run Lenght) sıkıştırma örnek verilebilir. Bu yöntemde verilerin ardarda adetleri tutularak sıkıştırma işlemi yapılır. Esas olarak bir çok sıkıştırma yöntemi bu mantık üzerine kurularak geliştirilmiştir. Örneğin pcx resim formatı bu formata çok benzer bir yöntemle sıkıştırma yapmaktadır.

Tekrarlama uzunluğu kodlaması yöntemi, bir sembolün verinin içinde ardarda tekrarlandığı durumlarda kullanılır. Örneğin ardarda gönderilen 25 adet X’i teker teker göndermek yerine, sembolün X olduğu ve 25 defa tekrarlandığı bilgisini göndermek yeterlidir.

Tekrarlama uzunluğu kodlaması için üç ayrı bileşene gerek vardır:

1. Sıkıştırma yapıldığına ilişkin özel bir işaret.
2. Sıkıştırılan sembolün ne olduğu bilgisi.
3. Sembolün ardarda kaç kez tekrarlandığı bilgisi.

Dolayısıyla, ardarda 3 kereden çok tekrarlanan karakterleri sıkıştırmak mantıklıdır. Bu sıkıştırma tekniği bir örnekle açıklanacak olursa; Aşağıdaki sayı dizisi, 60 karakterden oluşmaktadır. Her karakterin 8 bit ile gönderildiğini varsayılırsa, bu sayı dizisinin gönderilmesi için $8*60 = 480$ bite gerek vardır.

2326666666668889999939911121111000102333322200066666688889992

Şimdi bu dizi, tekrarlama uzunluğu kodlaması ile sıkıştırılsın. Sıkıştırmanın yapıldığı yerlerde özel bir işaret kullanılması gerekiyor. Bu işaret @ olsun. Sıkıştırılan kodlarda **özel sıkıştırma karakteri olan “@” + “sıkıştırılan rakam” + “kaç kez tekrarlandığı** bilgisi yazılır. Aşağıda, dizinin sıkıştırılmış halini görülmektedir.

232@69888@953991112@14000102@34222000@66@849992

Kodlama sonucunda dizi 47 karaktere düşmüştür. Yani $8*47 = 376$ bite gerek vardır. Görüldüğü gibi tekrarlama uzunluğu kodlaması, büyük oranda bir sıkıştırma sağlamamaktadır. Karakterlerin tekrarlanma oranları düştükçe sıkıştırmanın oranı da düşer. Bu sıkıştırma tekniğinin yaygın olarak kullanıldığı alanlardan birisi faks makineleridir. Yalnızca siyah ve beyazdan oluşan pikseller, ardarda gelme sayılarına göre kodlanırlar.

6.2.2 İstatistiksel Kodlama

İstatistiksel kodlama, “**Değişken Uzunlukta Kodlama (Variable Length Coding – VLE)**” olarak da bilinir. Bu kodlamadaki amaç, verinin içerdiği sembolleri istatistiksel olarak rastlanma sıklıklarına göre verimli biçimde kodlamaktır.

Bir karakterin 8 bit kullanılarak kodlandığı bir yöntemde, tekrarlama uzunluğuna göre kodlama tekniği yine her bir biti 8 karakter olarak kodlar, yalnızca tekrarlardan kaçınarak karakter sayısını azaltır, istatistiksel kodlamada ise amaç, sık kullanılan karakterleri daha az bit kullanarak ifade etmektir. Örneğin bir metinde ‘a’ harfi diğer harflere göre çok geçiyorsa belki de yalnızca iki bit (‘01’ gibi) kullanılarak kodlanabilir. Bu nedenle, farklı karakterler için farklı uzunlukta kodlar kullanır.

Bu kodlama yönteminde verinin içindeki sembollerin istatistiksel dağılım oranları bilinmelidir. Örneğin İngilizce bir metin ile Türkçe bir metinde geçen karakterlerin istatistiksel dağılımları farklı olacaktır. Sıkıştırma, belli bir istatistiksel dağılım varsayımına göre yapılır. Örnek olarak aşağıda, 8 ayrı karakterin istatistiksel dağılımları içeren bir tablo yer almaktadır.

M	A	B	C	D	E	F	G	H
P(m)	0.1	0.18	0.4	0.05	0.06	0.1	0.07	0.04

Tablo 6.1 : 8 ayrı karakterin istatistiksel dağılımları

Şimdi bu istatistiksel dağılıma göre verinin entropisi hesaplanırsa:

$$H = \sum_{m=1}^M P(m) \log_2 \frac{1}{P(m)} = 2.55 \text{ (bit / karakter)} \quad (6.4)$$

Normalde her karakter 8 bit ile gösterilmektedir. Bu durumda en fazla entropi

$$H_{\max} = \log_2 8 = 3 \text{ (bit / karakter)} \quad (6.5)$$

olur. Buradan da sıkıştırma verimliliği:

$$\Pi = \frac{2.55}{3} \% 100 = \% 85 \quad (6.6)$$

olarak hesaplanır. Görüldüğü gibi istatistiksel kodlama yöntemlerinin kullanılması ile önemli oranda bir kazanım elde edilir. En bilenen istatistiksel kodlama yöntemlerinin başında Huffman Kodlaması gelir.

6.2.2.1 Huffman Kodlaması

Huffman kodlaması olarak anılan sıkıştırma yöntemi veri haberleşmesinde, özellikle fax iletiminde sıklıkla kullanılır. Açık olarak tipik bir iş mektubunu sergileyen bir sayısal görüntüde, çoğunluk mektubun beyaz alanını, yüzde beşlik küçük bir bölüm siyah mürekkebi oluşturacaktır. Örneğin tekrar eden ardışık 1000 beyaz nokta yerine onları temsil eden daha kısa bir kod gönderilebilir. Sonuç olarak, fakslanan bir iş mektubu için veri sıkıştırma, toplam mesaj uzunluğunu önemli ölçüde azaltacaktır. Ancak siyah mürekkebin rasgele bir dağılımla mektubun %50'sini kapladığı durumlarda veri sıkıştırma bir fayda sağlamayacaktır.

Huffman kodlaması istatistiksel bir kodlama yöntemi olduğundan, iletilen bir verideki sembollerin farklı sıklıkta kullanılması özelliğine dayanır. Bu yöntem, diğer istatistiksel sıkıştırma yöntemlerinde olduğu gibi karakter başına sabit sayıda bit kullanmak yerine, sık tekrarlanan karakterlerin daha az, seyrek tekrarlanan karakterlerin daha fazla sayıda bit kullanılarak kodlandığı bir yöntemdir. Karakterler sabit sayıda bitten oluşmadığı için, sıkıştırılmış verinin aktarımı sırasında bit yönelimli aktarım teknikleri kullanılır.

İstatistiksel kodlamalarda öncelikle verinin analizi ve sembollerin istatistiksel dağılımının bilinmesi gerekir.

Huffman kodlama tekniği bir örnekle açıklanacak olursa; sıkıştırılacak veri yukarıda Tablo 6.1'de istatistiksel dağılımları verilmiş 8 karakter olsun.

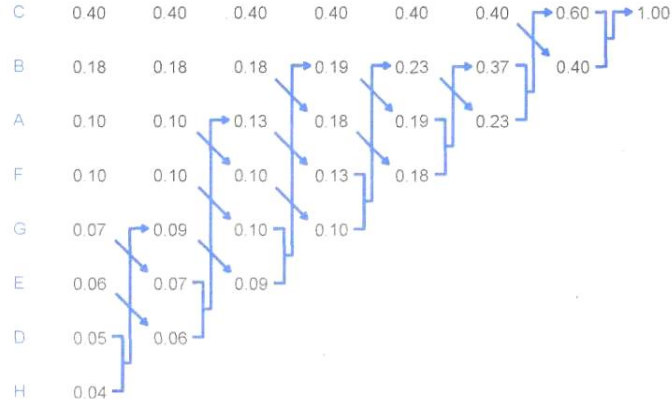
Bu noktada Huffman Kodlama algoritması iki aşamadan oluşur:

1. Azaltma
2. Bölme

Azaltma aşamasında şu adımlar izlenir:

1. Semboller sahip oldukları olasılık yüzdelerine göre en yüksek olasılıklı sembolden en düşüğe doğru sıralanır.
2. En alt sıradaki iki sembolün olasılık yüzdeleri toplanır. Listedeki sembol sayısı bir azaltılmış olur.
3. Semboller en yüksek olasılıklıdan en düşüğe doğru tekrar sıralanır.

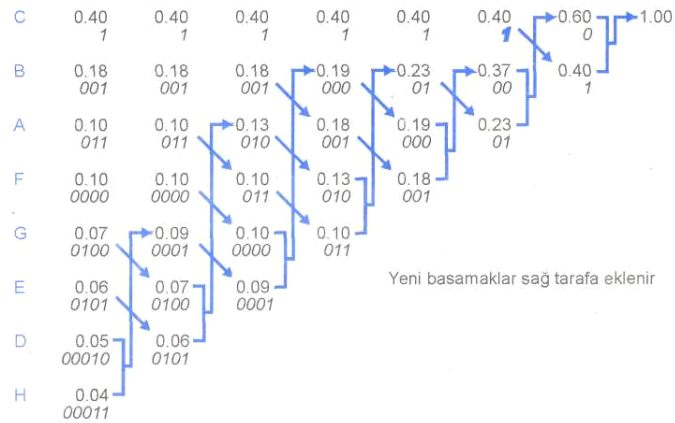
4. Bu adımlar, olasılıklarının toplam değeri %100 olan son iki sembol kalıncaya kadar sürdürülür. Şekil 6.1’de, azaltma algoritması gösterilmektedir.



Şekil 6.1 : Huffman kodlaması azaltma algoritması.

Bölme algoritmasında ise şu adımlar izlenir:

1. En son kalan iki sembole sırayla 0 ve 1 değerleri verilir.
2. Bir adım geriye gidilir, sembollerden birisi iki sembolün birleşmesinden oluşuyor ise, bu iki sembol için de 0 ve 1 değerleri eklenir.
3. Bu algoritma, daha önce toplanarak gelen tüm sembollere 0 ve 1 eklenerek kodlar oluşturulana kadar bölünerek gider.



Şekil 6.2 : Bölünme algoritması

Şekil 6.2’de bölünme algoritmasının nasıl işlediği gösterilmektedir. Uygulanan Huffman kodlamasının sonucunda oluşan kod değerleri Tablo 6.2’de verilmiştir.

M	C	B	A	F	G	E	D	H
P(m)	0.4	0.18	0.1	0.1	0.07	0.06	0.05	0.04
Kod	1	001	011	0000	0100	0101	00010	00011

Tablo 6.2 : Huffman kodlamasının sonucunda oluşan kod değerleri

Bu kodlama sisteminin kullanılması ile ortaya çıkan toplam kod uzunluğu şöyle hesaplanır:

$$L = \sum_{m=1}^M P(m)l_m \quad (6.7)$$

Formülü örneğe uygulanırsa oluşacak toplam kod uzunluğu,

$$L = 1(0.4) + 3(0.18+0.10) + 4(0.10+0.07+0.06) + 5(0.05+0.04) = 2.61 \quad (6.8)$$

Olarak elde edilir. Kod verimliliği ise entropi ve uzunluk kullanılarak şu formülle hesaplanabilir:

$$\Pi_{kod} = \frac{H}{L} \times \%100 \quad (6.9)$$

Buna göre verinin Huffman kodlaması kullanılarak sıkıştırılmasından elde edilen verimlik:

$$\Pi_{kod} = \frac{2.55}{2.61} \times \%100 = \%97.7 \quad (6.10)$$

olarak hesaplanır. Huffman kodlaması ile, sembollerin kodlanması için kullanılan bit sayısı azaltılarak çok yüksek oranlarda sıkıştırma sağlanabilir.

Huffman kodlamasında kullanılan başka bir yöntem de **“Huffman Kod Ağacı”** oluşturmaktır. Bu yöntemde de önce azaltma algoritması uygulanır, daha sonra kod ağacı çizilerek kodlar buradan oluşturulur. Ağacın yapısının dengeli olup olmadığı, kodlanan karakterlerin bulunma olasılıklarının bağlı sıklıklarına göre değişir. Bazı karakterlerin bulunma olasılığı diğerlerine göre daha fazla ise dengesiz bir ağaç, birbirine yakın olasılıklar var ise dengeli bir ağaç oluşur.

Huffman kod ağacı, ikili 0 ve 1'leri temsil eden dallanmalardan oluşan bir ağaçtır. Ağacın en tepesine **“kök düğümü”** denir. Ağaç aşağıya doğru dal düğümleri aracılığı ile yayılır. Bir dal düğümünden 2 tane dal çıkar. Soldaki dal ikili 0'ı, sağdaki dal ise ikili 1'i temsil

eder. Dalların açık uçlarına “**yaprak**” adı verilir ve her yaprakta bir karakter bulunur. Kök düğümünden yapraklara ulaşmak için izlenen yol, yaprakta bulunan karakterin kodunu verir.

Bir örnekle Huffman ağaç kodlaması yöntemi gösterilirse; Aşağıda veriyi oluşturan sekiz karakterin birbirlerine bağlı sıklıkları gösterilmektedir.

M	R0	R1	R2	R3	R4	R5	R6	R7
P(m)	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02

Tablo 6.3 : Örnek sekiz karakterin birbirlerine bağlı sıklıkları

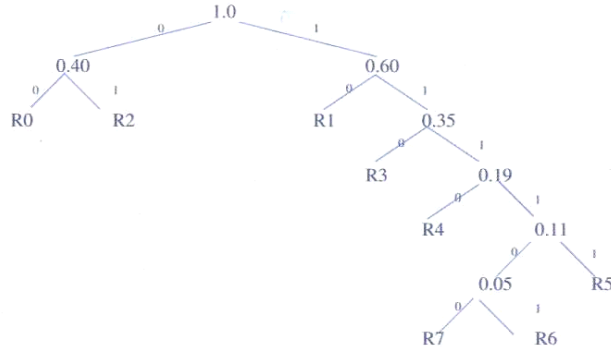
İlk olarak, birinci örnekte olduğu gibi azaltma algoritması uygulanır. Karakterler bulunma olasılıklarına göre en yüksek olasılıklı sembolden en düşük olasılıklıya doğru dizilir. En alt sırada bulunan iki karakter (R6-R7) yapraklardır ve bu yapraklara sırası ile 0 ve 1 dalları verilir. Bu iki dal birleşerek bir dal düğümü oluşturur (R6+R7). Karakterler tekrar sıralanır. Yine en alt sırada bulunan karakter yada dal düğümlerine sırası ile 1 ve 0 dalları atanır. Bu işlem kök düğüme ulaşana kadar sürer.

Birinci Sıralama	İkinci Sıralama	Üçüncü Sıralama	Dördüncü Sıralama	Beşinci Sıralama	Altıncı Sıralama	Yedinci Sıralama
R1-0.25	R1-0.25	R1-0.25	R1-0.25	R3+R4+R5+R6+R7-0.35	R0+R2-0.40	R1+R3+R4+R5+R6+R7-0.60 (1)
R2-0.21	R2-0.21	R2-0.21	R2-0.21	R1-0.25	R3+R4+R5+R6+R7-0.35 (1)	R0+R2-0.40 (0)
R0-0.19	R0-0.19	R0-0.19	R0-0.19	R2-0.21 (1)	R1-0.25 (0)	
R3-0.16	R3-0.16	R3-0.16	R4+R5+R6+R7-0.19 (1)	R0-0.19 (0)		
R4-0.08	R4-0.08	R5+R6+R7-0.11 (1)	R3-0.16 (0)			
R5-0.06	R5-0.06 (1)	R4-0.08 (0)				
R6-0.03 (1)	R6+R7-0.05 (0)					
R7-0.02 (0)						

Tablo 6.4 : Huffman ağaç kodlaması yöntemi

Daha sonra, her karakter için ilk sütundan başlayarak hangi dalların verildiği listelenir. Bu yaprak düğümünden köke doğru o karaktere gelene kadar geçilen dallardır. Karakterin kodu ise kök düğümünden yaprak, düğüme doğru oluşturulur. Dolayısıyla kodları bulmak için liste tersine çevrilir.

Aşağıdaki Şekil 6.3'te, bu örnek için oluşturulmuş Huffman kod ağacı gösterilmektedir.



Şekil 6.3 : Örnek Huffman kod ağacı

6.3 Kaynak Kodlaması (Kayıplı Sıkıştırma)

Kaynak kodlaması, genellikle görüntü ve video sıkıştırmalarında kullanılır, iki boyutlu resimler, piksel olarak adlandırılan küçük parçalardan oluşur. Bir görüntü iletilirken her pikselin karakteristiğini oluşturan yoğunluk, renk ve konum değerlerin korunması gerekir. Renk kırmızı, yeşil ve mavi işaretler ile RGB olarak yada işaret parlaklığı için beyaz (kırmızı+yeşil + mavi) işaretler ve bunun yanı sıra zıtlık (kontrast) için beyaz-kırmızı ve beyaz-mavi işaretlerle gönderilir. Video aktarımı sırasında ise bunların yanı sıra her pikselin zamandaki konumu da iletilmelidir.

Kaynak kodlaması, entropi kodlamasının tersine verinin içeriği ile ilgilenir. Kaynak kodlamasında uygulanan sıkıştırma yöntemi, verinin bir kısmını atarak yalnızca genel bir kurala uyan verinin aktarılmasıdır. Aktarılan veri, hiçbir zaman tam olarak gönderilen veri değildir. Bu nedenle bu tür kodlamalara **“kayıplı kodlamalar”** denir. Ancak bu kodlama yöntemi görüntü ve video gibi görsel verinin sıkıştırılmasında kullanıldığı için, bu kayıplar çok önemli değildir. Çünkü insan gözü bu atlamaları algılayamaz. Zaten verinin içinde de gözün algılayamadığı bozukluklar vardır.

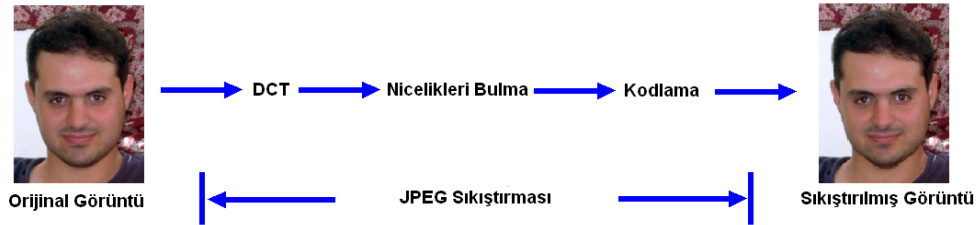
Kaynak kodlamasının en yaygın olarak kullanılan iki örneği, görüntü sıkıştırmasında kullanılan JPEG ve video sıkıştırmasında kullanılan MPEG kodlamalarıdır. Bu kodlamalarda çok yüksek oranlarda sıkıştırma sağlanabilir. Aşağıda bu yöntemler anlatılmaktadır.

6.3.1 JPEG - Resim Sıkıştırması

JPEG kodlama sistemi, sayısal görüntülerin (image) sıkıştırılmasında kullanılan bir standarttır. Fotoğraf Uzmanları Birliği Grubu (Joint Photographic Experts Group) tarafından geliştirilen kodlama sistemi, bu grubun baş harfleri ile anılır. JPEG, tek çerçevede oluşan renkli yada tek renkli görüntülerde büyük oranlarda sıkıştırma sağlar.

JPEG sıkıştırma algoritması üç aşamadan oluşur:

1. DCT,
2. Nicelikleri bulma (quantization)
3. Kodlama.

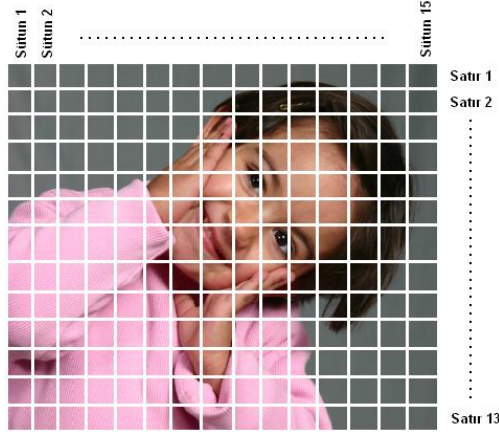


Şekil 6.4 : JPEG sıkıştırmasının aşamaları.

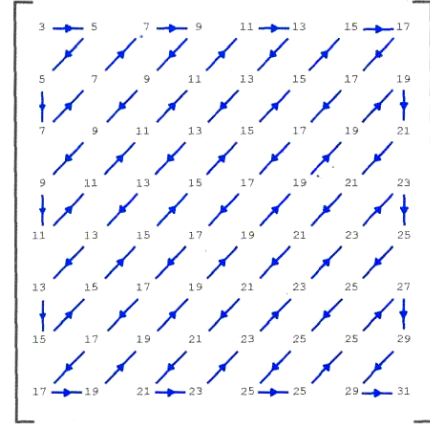
DCT (Ayrık Cosinüs Dönüşümü - Discrete Cosine Transform), görüntüye uygulanan ilk işlemdir. Bu aşamada herhangi bir sıkıştırma yapılmaz, dolayısıyla bilgi kaybı olmaz. Görüntü, yalnızca verinin fazlalıklarının daha rahat atılabileceği bir biçime dönüştürülür.

DCT algoritmasında resim 8*8 bloktan oluşan pikseller halinde alınır ve 8*8 frekans katsayısından oluşan bir matrise dönüştürülür. 64 katsayıdan oluşan bu çıktı, iki boyutta (x ve y eksenlerinde) bir işaret olarak tanımlanır.

“Nicelikleri bulma” algoritmasında, DCT aşamasında frekans katsayıları ile ifade edilmiş verilerin gereksiz bitleri atılır. Bu amaçla, her katsayı için 1-255 aralığında nicelik tanımlayan bir tablo oluşturulur. Bu tabloda her katsayı, bir önceki bloktaki katsayıdan farkı ile gösterilir. Piksel katsayılarının sıralaması, tablonun sağ üst köşesinden sol alt köşesine doğru çapraz biçimde taranır. JPEG nicelik bulma tablosunu her katsayının niceliğini belirlemek için kullanır. Bu tablo ne kadar bilginin atılacağını, dolayısıyla ne kadar sıkıştırma yapılacağını belirler. Genellikle niceliği 1'e yakın düşük katsayılar ve çok yüksek katsayılar atılır.



Şekil 6.5 : DTC algoritması ile matrise dönüştürme.

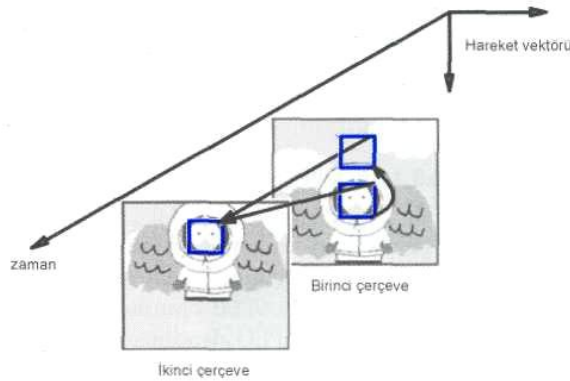


Şekil 6.6 : Nicelikleri bulma tablosu.

En son aşama, nicelikleri bulma tablosunda işlenen katsayıların istatistiksel olasılıklarına yada entropilerine göre kodlanmasıdır. Bu kodlama Huffman kodlama sistemine göre yapılır. JPEG kodlamada, % 200'den % 2000'e varan oranlarda bir sıkıştırma sağlanabilir. Düşük sıkıştırma oranlarında aktarılan görüntü, özgün görüntüyle olan farkı anlaşılacak şekilde yeniden oluşturulabilir.

6.3.2 MPEG- Video Sıkıştırması

MPEG kodlama sistemi, Hareketli Resim Uzmanları Birliği Grubu (Motion Picture Experts Group) tarafından geliştirilen ve video sıkıştırılmasında kullanılan bir standarttır. MGEP1 1.5 mbps, MPEG2 ise 10 mbps hızında hareketli resimleri kodlamada kullanılır. CD'lerde kullanılmak üzere daha düşük hızlar için de standartlar vardır.



Şekil 6.7 : MPEG başvuru sistemi.

Hareketli resimler, birbiri ardına gelen resim çerçevelerinin video hızına uygun olarak ardarda gösterilmesidir. Bu resim çerçevelerinden her biri JPEG'de olduğu gibi DCT temelli teknikler kullanılarak sıkıştırılabilir. Ancak bu yeterli değildir. Hareketli resimlerde fazlalığın

büyük bir kısmını birbirine çok benzeyen çerçeveler oluşturur. ardarda gelen iki çerçevede herhangi bir hareket yoksa, iki çerçeve birbirinin aynı olabilir yada çok az bir fark olabilir. Bir çerçevenin içerdiği bilginin büyük bir kısmı diğer çerçeveden elde edilebilir. Bir çerçevedeki cisim diğer çerçevede yalnızca konum değiştirmiş olabilir. Hareketli resim sıkıştırmanın temeli, hareketin vektörünü yakalamaktır. MPEG, çerçeveleri her defasında tekrarlamak yerine birbirlerine başvuru ile yeniden elde etmenin algoritmasını kullanır.

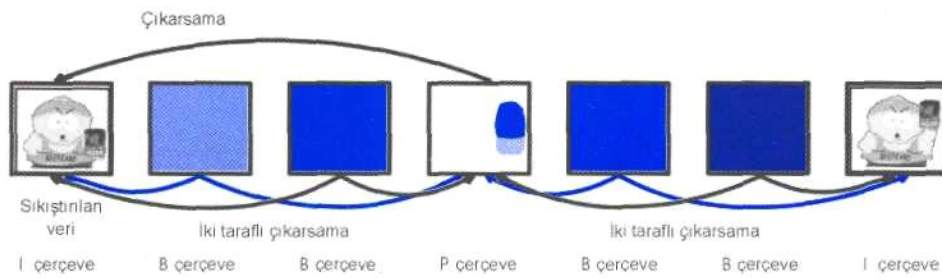
MPEG, hareketli resmi oluşturan tüm çerçeveleri üç ayrı biçimde ele alır:

1. I çerçeveler: Ara çerçeveler.
2. P çerçeveler: Tahmin edilen çerçeveler.
3. B çerçeveler: iki taraflı tahmin edilen çerçeveler.

I çerçeveler, herhangi bir çerçeveye bağlı olmadan kodlanmış çerçevelerdir. Bu çerçevelere yalnızca JPEG sıkıştırması uygulanır, bu da oransal olarak küçük bir sıkıştırma. I çerçeveler, diğer çerçeveleri oluşturmak için başvuru noktalarıdır. Bu çerçeveler aynı zamanda geri sarma sırasında videoyu tekrar oynatmak için kullanılan noktalardır.

P çerçeveler, kendilerinden önce gelen I çerçeveye başvuru ile kodlanmış çerçevelerdir. Bir P çerçeve, yalnızca kendinden önce gelen I çerçeveden farkları tahmin edilerek (çıkarılarak) kodlanır.

B çerçevelerin oluşturulmasında ise iki başvuru çerçevesi gerekmektedir. Bu başvurular B çerçeveden hemen önce ve sonra gelen I yada P çerçeveleridir. Aşağıda I, P ve B çerçevelerin sıkıştırma başvuru sistemi gösterilmektedir.



Şekil 6.8 : MPEG sıkıştırma I, P ve B çerçeveleri.

I, P ve B çerçevelerinin kullanılma oranı sıkıştırma oranını belirler. I çerçevelerinin oranı arttıkça görüntü kalitesi artar, ancak sıkıştırma oranı düşer. MPEG genellikle % 9000 oranında sıkıştırma sağlar. Bu oran % 15000'e kadar çıkabilir. MPEG algoritması oldukça karmaşık olduğu için, yüksek işlem gücü gerektirir ve genellikle çevrimdışı olarak sabit disk tarafından yapılır.

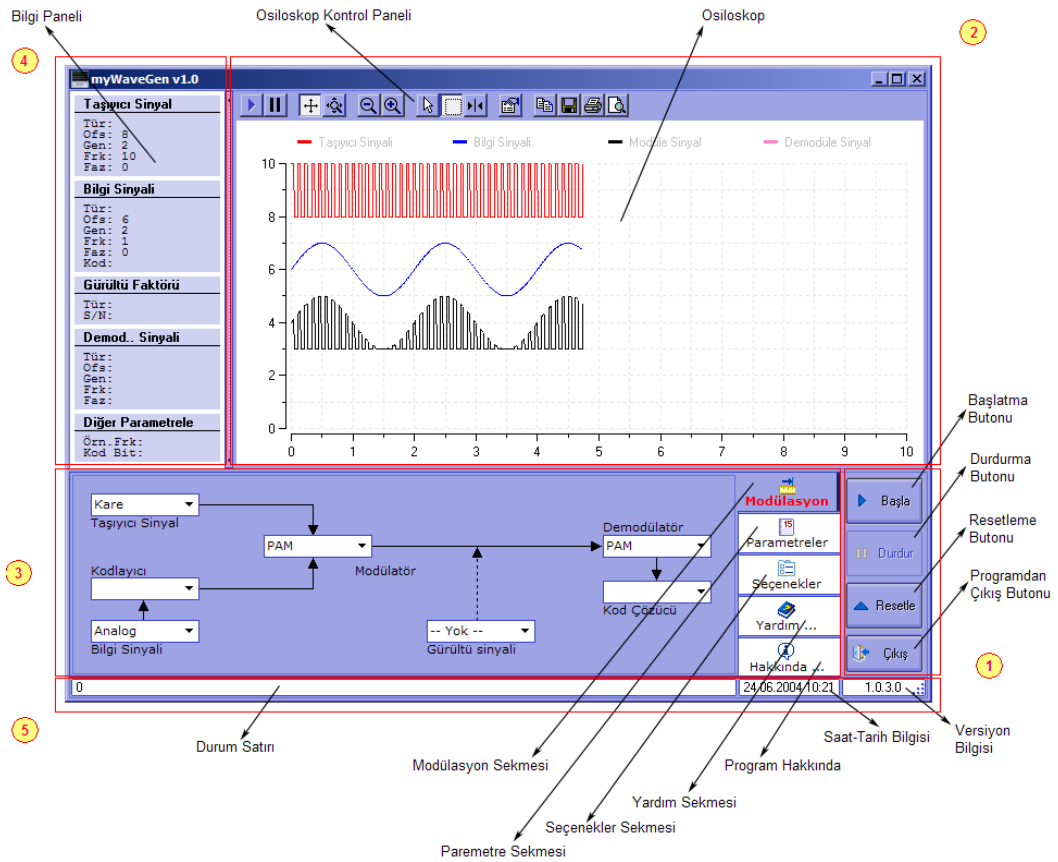
7 UYGULAMA VE MYWAVEGEN PROGRAMI

7.1 Giriş

Bu program önceki bölümlerde değinilen modülasyon türleri, kodlama teknikleri ve bunların demodülasyon işlemleri gerçekleştirmek amacıyla hazırlanmıştır. Program Delphi 7.0 görsel uygulama geliştirme aracı ile hazırlanmış ve Windows (98, 2000, XP) platformlarında sorunsuzca çalıştırılmıştır.

Program temelde bir simülasyon programı olup, dalga işaretlerinin (non-realtime) gösterimini yapmaktadır. Doğal olarak tüm simülasyon programlarında olduğu gibi matematiksel işlemlerin yoğunluğundan dolayı özellikle programın koşturulacağı donanımın dikkate alınması gerekir. Program, Pentium4 işlemci tabanlı ve 256 MB RAM bulunan bir PC de sorunsuzca çalışabilmektedir.

Program oldukça esnek ve kullanışlıdır. Kullanım felsefesi simülasyon öncelikli olarak, özellikle osiloskop ve modülasyon panellerinin rahat ve anlaşılır olması üzerine kurulmuştur.

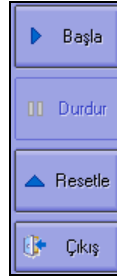


Şekil 7.1 : myWaveGen Programı ana ekranı

Program ilk çalıştırıldığında Şekil 7.1'deki ana ekran gelir. Şekilde belirtildiği üzere program beş ana kısımdan oluşmaktadır. Bunlar;

1. Ana kontrol paneli (Başlat, Durdur, Resetle, Çıkış)
2. Osiloskop Paneli
3. Modülasyon ve Parametre Paneli (Modülasyon, Parametreler, Seçenekler, Yardım ve Program Hakkında)
4. Bilgi Paneli
5. Durum Satırı

7.2 Program Ana Kontrol Paneli



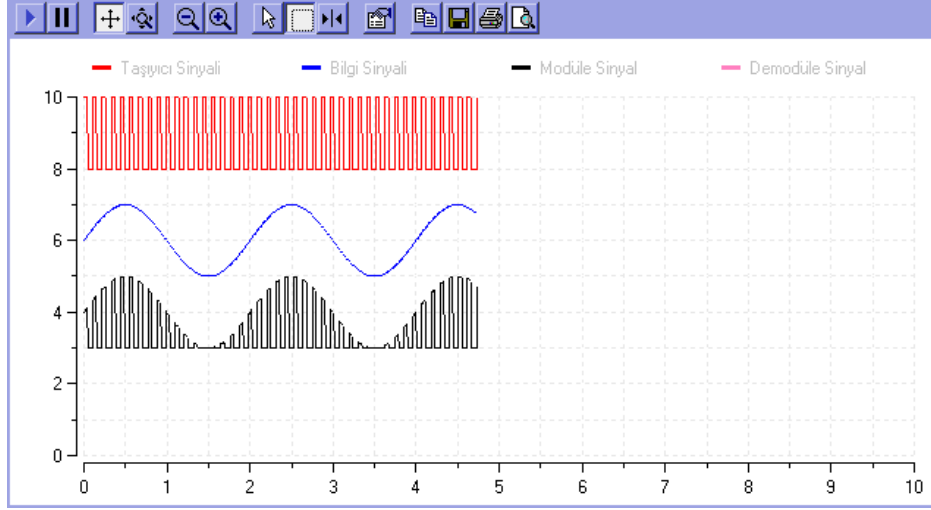
Şekil 7.2 : Program ana kontrol paneli

Programın ana kontrol panelinin işlevi çok basit ve kullanışlıdır. Bu panelde 4 düğme bulunmaktadır;

1. **Başla Butonu:** Programda ayarlanan parametrelere göre simülasyon sürecini başlatır.
2. **Durdur Butonu:** Normalde pasiftir ve Başla butonunun tersi işlev görür. Başla butonu ile aktif hale getirilen simülasyon sürecini durdurur.
3. **Resetle Butonu:** Bu buton osiloskop ve zaman ile ilgili tüm değerlerin sıfırlanmasını sağlar. Simülasyon sürecinin istenilen herhangi bir anında bu butona basılarak tüm değerler sıfırlanabilir.
4. **Çıkış Butonu:** Programdan çıkışı sağlar. (Programı kapatır)

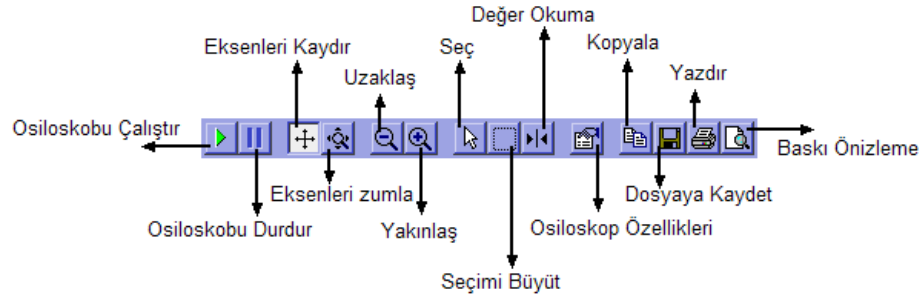
7.3 Osiloskop Paneli

Osiloskop paneli, çok gelişmiş ve kullanışlı bir paneldir. Programın ara yüzünün en temel parçası olarak görülebilir. Zira tüm simülasyon sonuçları bu ekranda gösterilir.



Şekil 7.3 : Osiloskop Paneli

Bu panel temel olarak iki kısımdan oluşur. Bunlar osiloskop ekranı ve osiloskop kontrol panelidir.



Şekil 7.4 : Osiloskop Kontrol Paneli

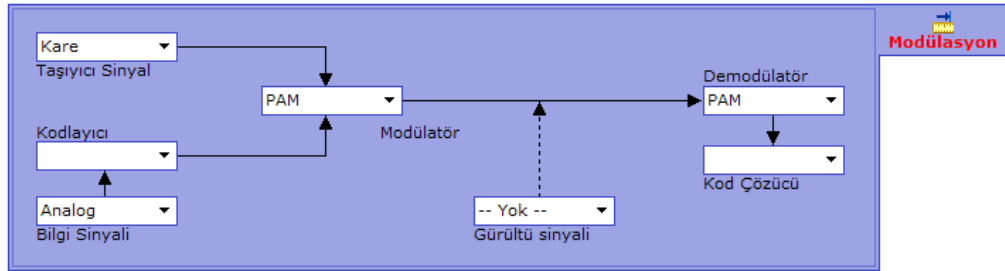
Osiloskop ekranı, arka planda hesaplanan simülasyon sonuçlarını son kullanıcıya gösterir. Kullanıcı simülasyon sonuçlarının tümünü yani tüm kanalları (taşıyıcı, bilgi, modüle sinyal ve istendiğinde demodüle sinyali) görür.

Osiloskop kontrol panelinin üzerinde bulunan butonlar aracılığı ile kullanıcı istediği anda ekran görüntüsünü yakınlaştırıp daha net inceleyebilir. Ayrıca sonuçları yazıcıya çıktı olarak gönderebilir veya bir dosyaya saklayabilir. Yine bu kontrol paneli aracılığı ile ekrandaki sinyaller ile ilgili seviye değerlerini öğrenebilir.

7.4 Modülasyon ve Parametre Paneli

Modülasyon ve Parametre Paneli, simülasyon ile ilgili tüm parametrik değerlerin değiştirilebildiği paneldir ve programın en önemli bölümlerinden biridir. Şekil 7.1’de görüleceği üzere bu panel beş adet sekmeden oluşur. Bu sekmeler amacına uygun olarak sırasıyla Modülasyon, Parametre, Seçenekler, Yardım ve Hakkında olarak adlandırılmıştır.

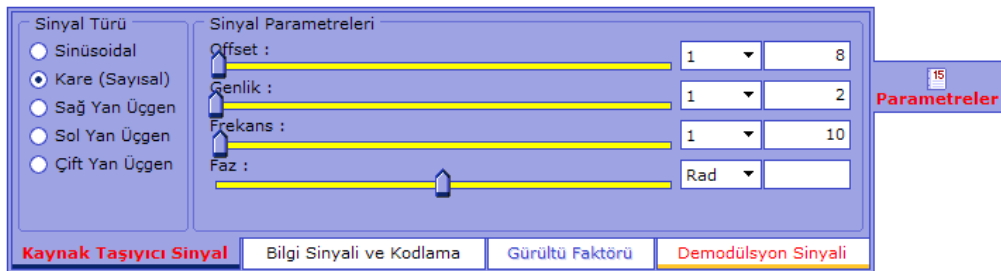
7.4.1 Modülasyon Sekmesi



Şekil 7.5 : Modülasyon Sekmesi

Bu sekmede kullanıcı istediği taşıyıcı ve bilgi sinyallerini kullanarak modülasyon ve demodülasyon işlemlerini gerçekleştirebilir. Parametreler aşağı açılır liste kutuları şeklindedir. Kullanıcı burada gerekli seçimleri yaptıktan sonra simülasyon sonuçlarını osiloskop ekranından anlık olarak takip edebilir. Simülasyonun herhangi bir anında kullanıcı bu seçimlerden istediğini değiştirerek sonuçlarını gözlemleyebilir.

7.4.2 Parametre Sekmesi

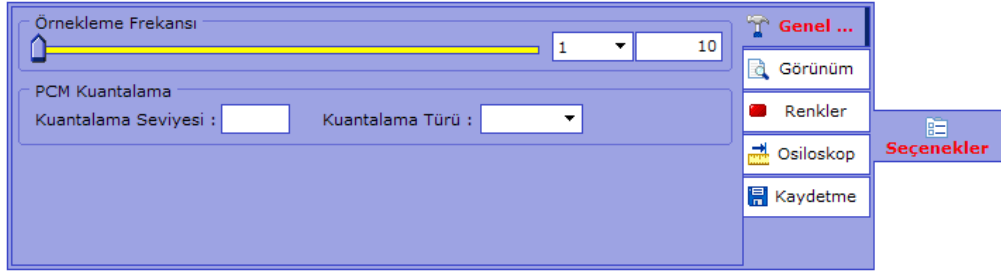


Şekil 7.6 : Parametre Sekmesi

Bu sekmede kullanıcı Taşıyıcı, Bilgi ve Demodülasyon sinyallerinin ofset, genlik, frekans ve faz gibi değişken parametrelerini istediği gibi değiştirme imkânına sahiptir. Buradaki parametreler değiştirilerek oluşan tepkiler osiloskop ekranından anlık olarak gözlemlenebilir. Ayrıca sinyallerin dalga şekilleri (sinyal türü) de bu sekmeden değiştirilebilir. Bu işlem modülasyon sekmesinden de gerçekleştirilebilir.

Normal şartlarda program simülasyon işlemlerini ideal (gürültüsüz ve bozunmasız) ortamda hesaplar. Eğer istenirse yine bu sekmeden gürültü faktörü belirlenebilir ve gerçek dünya koşulları oluşturulabilir. (Not: Bu kısım henüz programa entegre edilmemiştir).

7.4.3 Seçenekler Sekmesi



Şekil 7.7 : Seçenekler Sekmesi

Bu sekmede program ile ilgili bir takım özelleştirilebilir seçenekleri değiştirilmesine olanak verilir. Sekme, beş alt sekmeden oluşmuştur. Bunlar;

1. **Genel:** Örneklem frekansı, PCM kuantalama seviyesi gibi program genelinde etkili olan özellikler değiştirilir.
2. **Görünüm:** Programın görünümü ile ilgili özelliklerin değiştirilmesine olanak verir.
3. **Renkler:** Bu alt sekmede, programın görsel renk düzenleri değiştirilebilir.
4. **Osiloskop:** Bu alt sekmede, osiloskopla ilgili her türlü görünüm ve biçim özellikleri ayarlanabilir.
5. **Kaydetme:** Bu kısımda programın kaydetme seçenekleri özelleştirilebilir.

7.4.4 Hakkında Sekmesi



Şekil 7.8 : Hakkında Sekmesi

Program hakkında yapımçı, yapım yılı ve ulaşım bilgilerinin verildiği bu sekme aynı zamanda programın açılış ekranının bir parçasıdır.

7.4.5 Yardım Sekmesi

Bu sekmede programın kullanımı ve programın kullandığı teknikler hakkında teknik ve detaylı yardım bilgileri verilmektedir.

7.5 Bilgi Paneli

Taşıyıcı Sinyal
Tür: Ofs: 8 Gen: 2 Frk: 10 Faz: 0
Bilgi Sinyali
Tür: Ofs: 6 Gen: 2 Frk: 1 Faz: 0 Kod:
Gürültü Faktörü
Tür: S/N:
Demod. Sinyali
Tür: Ofs: Gen: Frk: Faz:
Diğer Parametrele
Örn. Frk: Kod Bit:

Bu panelin temel amacı kullanıcıya simülasyon esnasında, parametre değerleri hakkında bilgi vermektir. Panel istendiği anda gösterilip gizlenebilme özelliğine sahiptir.

Panel üzerinde, o anki Taşıyıcı Sinyal, Bilgi Sinyali, Gürültü Sinyali, Demodülasyon Sinyali ve Örneklem frekansı gibi parametre değerleri okunabilir.

Şekil 7.9 : Bilgi Paneli

7.6 Durum Satırı

PAM modülasyonu	24.06.2004 10:50	1.0.3.0 ..
-----------------	------------------	------------

Şekil 7.10 : Durum Satırı

Durum Satırı, programın çalışması sırasında o anki durum hakkında bilgi verir. Ayrıca bu satır üzerinde o anki tarih bilgileri ve programın çalışan aktif versiyonu bilgisi de gösterilir.

8 SONUÇ

Özellikle son yıllarda haberleşme konusu günlük hayatımızın her alanına girmiştir. Haberleşmenin temel felsefesi bilgiyi bir yerden başka bir yere aktarmak (iletmek ve/veya paylaşmak) tır. Yaşamakta olduğumuz bilgi çağında, hızlı ve güvenli bilgi alışverişinin sağlanması, bilginin çok kısa sürede yenilenmesi ve gerekli kaynaklara iletilmesi gereksinimi, iletişim ve iletişim amaçlı sistemlerin önemini her geçen gün daha da arttırmış ve gelişim sürecini hızlandırmıştır.

Öncelikle bu seminer çalışmasının amacı; haberleşme, modülasyon teknikleri, hata denetimi ve veri sıkıştırma konularının incelenmesi olmuştur. Bu bağlamda;

İlk bölümde, haberleşmenin önemi üzerine bir giriş yapılmış ve tarihsel süreç içerisinde bu konuyla ilgili genel bir anlatım verilmiştir.

İkinci bölümde, haberleşmenin nasıl yapıldığı sorusuna cevap aranmış ve bu kapsamda günümüzde kullanılan haberleşme türleri incelenmiş ve haberleşmede karşılaşılan sorunlar ve kısıtlamalara değinilmiştir.

Üçüncü bölümde, haberleşmenin en temel konularını teşkil eden modülasyon konusu ve modülasyon teknikleri ele alınmıştır. Ayrıca modülasyon teknikleri hakkında detaylı bir inceleme yapılmıştır. Bu bağlamda özellikle sayısal modülasyonlar üzerinde durulmuştur.

Dördüncü bölümde, sayısal modülasyon konusuna paralel olarak sayısal kodlama teknikleri ele alınmış ve bunlardan en çok kullanılanları izah edilmiştir.

Beşinci bölümde, haberleşmede istenmeyen, ancak kendi doğasının bir gereği olarak bir o kadar önüne geçilemeyen haberleşme hataları, bunların kaynakları ve hataları düzeltme teknikleri incelenmiştir.

Altıncı bölümde ise haberleşmenin daha etkili ve verimli olmasını sağlayan veri sıkıştırma teknikleri anlatılmıştır. Özellikle günümüzde en çok kullanılan RLE, Huffman, MPEG ve JPEG gibi teknikler üzerinde durulmuştur.

Son olarak yedinci bölümde, bu seminer çalışması kapsamında incelenen modülasyon ve kodlama tekniklerinin bir uygulaması olarak geliştirilen myWaveGen programı anlatılmış ve örnek çıktılarıyla izah edilmiştir.

İnsanoğlunun bilgi birikimi her geçen gün artmakta ve bu sahip olunan bilgi yığının saklanması veya başka hedeflere iletilmesi, bu artışa orantılı olarak gittikçe zorlaşmaktadır. Bu da mevcut kaynakların daha verimli ve etkili kullanılması konusunu gündeme getirmektedir. Bir taraftan daha yüksek kapasiteli depolama sistemleri ve daha hızlı haberleşme sistemleri bu soruna bir çözüm arayışı getirmekte öte yandan mevcut imkanların kullanılması ile bilgilerin sıkıştırılarak saklanması veya iletilmesi, bu noktada soruna etkili bir çözüm yaklaşımı sunmaktadır. Bu temellere dayanarak, ileriki aşamalarda özellikle veri sıkıştırma ve veri eşitleme (equalization) konuları üzerine daha yoğun çalışmalar yapılması hedeflenmektedir.

KAYNAKLAR

- [1] Proakis, J. G., 1995, "Digital Communications", McGraw-Hill
- [2] Glover, I., Grant P., 1998, "Digital Communications", Prentice Hall
- [3] Killen, H.B., 1994, "Modern Elektronik İletişim Teknikleri", Çev: Mustafa AKAY, MEB Yayınları
- [4] Ronayne, J., 1994, "Sayısal Haberleşmeye Giriş", Çev: Hasan Hüseyin ERKAYA, MEB Yayınları
- [5] Derin, H., Aşkar, M., 1979, "İletişim Kuramı", Ortadoğu Teknik Üniversitesi Mühendislik Fakültesi
- [6] Smale, P.H., 1994, "Haberleşme Sistemlerine Giriş", Osman PARLAKTUNA, MEB Yayınları
- [7] Kayran, A.H., 1999, "Analog Haberleşme", Birsen Yayınevi
- [8] Kayran, A.H., Panayırıcı, E., Aygözü, Ö., 1999, "Sayısal Haberleşme", Birsen Yayınevi
- [9] Pastacı, H., 1996, "Modern Elektronik sistemler", Yıldız Teknik Üniversitesi Yayınları
- [10] Çölkesen, R., Örencik, B., 1999, "Bilgisayar Haberleşmesi ve Ağ Teknolojileri", Papatya Yayıncılık
- [11] Kaplan, Y., 2000, "Veri Haberleşmesi Kavramları", Papatya Yayıncılık
- [12] Johnson, J.J., Fletcher, B.D., 1997, "Introductory Radio and Television Electronics", Macmillan education Ltd.
- [13] Blleloch, G.E., 2001, "Introduction to data Compression", Carnegie Mellon University
- [14] Smith, S.W., 1999, "The Scientist and Engineer's Guide to Digital Signal Processing", California Technical Publishing
- [15] Vaseghi, S.V. 2000, "Advanced sinal Processing and Noise Reduction", John Wiley & Sons Ltd.
- [16] Nelson, M., 2000, "The Data Compression Book", IDG Books Worldwide Inc.
- [17] Stergiopoulos, S., 2001, "Advanced Signal Processing Handbook", CRC Press LLC
- [18] Morreale P., Terplan K., 2001, "The CRC Handbook of Modern Telecommunications", CRC Press LLC
- [19] Baykal, N., 2001, "Bilgisayar Ağları", SAS Bilişim Yayınları
- [20] Türker, M.A., 2004 "Beyaz Noel ve Yaz Yağmuru: Veri Sıkıştırmanın Doğası Üzerine", www.teknoturk.org
- [21] Algan, S., 2004, "Huffman Veri Sıkıştırma Algoritması ve Uygulaması (C#)", www.ceturk.com

ÖZGEÇMİŞ

Musa ÇIBUK

Fırat Üniversitesi
Mühendislik Fakültesi
Elektrik – Elektronik Mühendisliği
23119, ELAZIĞ

Tel : 0-424-237 0000 / 5020
E-posta : mcibuk@firat.edu.tr
Web : <http://mcibuk.firat.edu.tr>

- 25.09.1976 Dicle/Diyarbakır’da doğdu.
- 1990 – 1993 Ergani Lisesi – Ergani/Diyarbakır (Ortaöğretim)
- 1993 – 1997 F.Ü., Müh. Fak. Elektronik – Elektronik Bölümü’nde Lisans Eğitimi
- 1997 – 1998 Özel bir müteahhitlik şirketinde Şantiye Şefliği
- 1998 – 2000 M.E.B. Araban Çok Programlı Lisesi ve Elazığ Merkez Endüstri Meslek Lisesi’nde Elektrik Öğretmenliği
- 1999 – 2002 F.Ü., Fen Bilimleri Enstitüsü, Elektronik – Elektronik Mühendisliği Anabilim Dalında Yüksek Lisans Eğitimi
- 2000 - Fırat Üniversitesi, Enformatik Bölümü’nde Uzman
- 2002 - F.Ü., Fen Bilimleri Enstitüsü, Elektrik – Elektronik Mühendisliği Anabilim Dalında Doktora Eğitimi